Extended abstracts of the third Workshop on Mathematical Cryptology (WMC 2012) and the third international conference on Symbolic Computation and Cryptography (SCC 2012)

Castro Urdiales, Spain, 9–13 July 2012

WMC 2012 and SCC 2012 are supported by:

- CIEM (International Centre for Mathematical meetings)
- Philips Research Europe
- Université Pierre et Marie Curie
- INRIA
- Universidad de Cantabria

Local Organization
Algorithmic Mathematics And Cryptography Research Group (AMAC)
Universidad de Cantabria, Santander, Spain.
http:// grupos.unican.es/amac

- Adrià Alcalá
- Domingo Gómez
- Jaime Gutierrez
- Álvar Ibeas
- Santos Merino
- Alina Ostafe

Third Workshop on Mathematical Cryptology

http://wmc2012.unican.es

PLENARY INVITED SPEAKERS:

- Antoine Joux (Université de Versailles, Paris, France)
- Susan Landau (Harvard University, Boston, US)
- Andrzej Schinzel (Polish Academy of Science, Warszawa, Poland)
- Igor Shparlinski (Macquarie University, Sydney, Australia)

INVITED SPEAKERS:

- Oscar Garcia-Morchon (Philips Research Europe, Eindhoven, Netherlands)
- Jorge Jiménez Urroz (Polytechnic University of Catalonia, Barcelona, Spain)
- Ludo Tolhuizen (Philips Research Europe, Eindhoven, Netherlands)
- Alev Topuzoglu (Sabanci University, Istanbul, Turkey)
- Jorge Villar (Polytechnic University of Catalonia, Barcelona, Spain)
- Arne Winterhof (Johann Radon Institute for Computational and Applied Mathematics, Linz, Austria)

Third International Conference on Symbolic Computation and Cryptography

http://scc2012.unican.es

INVITED SPEAKERS:

- Nadia Heninger (University of California, San Diego, US)
- Éric Schost (University of Western Ontario, London, Canada)
- Damien Stehlé (LIP laboratory, CNRS, ENSL, INRIA, UCBL, U. Lyon, École N. Supérieure de Lyon, France)

PROGRAMME CHAIRS

- Jean-Charles Faugère, UPMC-INRIA, France
- Jaime Gutierrez, Universidad de Cantabria, Santander, Spain

PROGRAMME COMMITTEE

- Martin Albrecht (LIP6-UPMC Univ Paris 6 & INRIA, France)
- Stanislav Bulygin (Center for Advanced Security Research Darmstadt, Germany)
- Carlos Cid (Royal Holloway, University of London, UK)
- Danilo Gligoroski (Norwegian University of Science and Technology, Norway)
- Martin Kreuzer (Universität Passau, Germany)
- Dongdai Lin (Institute of Software of Chinese Academy of Sciences, China)
- Vadim Lyubashevsky (INRIA, France)
- Gregor Leander (Technical University of Denmark, Denmark)
- Ayoub Otmani (GREYC-Ensicaen & University of Caen & INRIA, France)
- Ludovic Perret (LIP6-UPMC Univ Paris 6 & INRIA, France)
- Vladimir Shpilrain (The City College New York, US)
- Boaz Tsaban (Bar-Ilan University, Israel)

WCM

Invited Lectures

Contributed Talks

SCC

Invited Lectures

Contributed Talks

Invited Lectures

WMC

Invited Lectures

Small primitive roots and malleability of RSA moduli

Jorge Jiménez Urroz

This talk is based on a joint work with Luis Dieulefait.

Abstract

In their paper [9], P. Paillier and J. Villar make a conjecture about the malleability of an RSA modulus. In this paper we present an explicit algorithm refuting the conjecture. Concretely we can factorize an RSA modulus n using very little information on the factorization of a concrete n' coprime to n. However, we believe the conjecture might be true, when imposing some extra conditions on the auxiliary n' allowed to be used. In particular, the paper shows how subtle the notion of malleability is.

Introduction

The existence of a tradeoff between one-wayness and chosen ciphertext security dates back to the eighties when, for example, it was observed in [10, 11, 4]. In some sense, one cannot achieve one-way encryption with a level of security equivalent to solve certain difficult problem, at the same time as the cryptosystem being IND-CCA secure respect to it. This so called paradox has been attempted to be formally proved many times, by a number of authors, since first observed. However no one succeeded until very recently, when Pailler and Villar (cf. [9]) clarified the question for the case of factoring-based cryptosystems. In particular, they give precise conditions for certain security incompatibilities to exist. More precisely, they reformulate the paradox in terms of key preserving black-box reductions and prove that if factoring can be reduced in the standard model to breaking one-wayness of the cryptosystem then it is impossible to achieve chosen-cyphertext security. As the authors mention in their paper (cf. [9]), combining this result with the security proofs contained in [2, 3] gives a very interesting separation result between the Random Oracle model and the standard model.

Moreover, assuming an extra hypothesis, which they call "non-malleability" of the key generator, they are able to extend the result from key preserving black box reductions to the case of arbitrary black box reductions.
Hence, as the authors themselves stress in [9], it is very important to study non-malleability of key generators. In fact, they conjecture that most instance generators are non-malleable, but no arguments are given to support this belief. The goal of this note is to shed some light on this open question.

Actually, the notion of non-malleability captures a very basic fact in arithmetic: intuitively, one tends to believe that the problem of factoring a given number n (an RSA modulus) is not made easier if we know how to factor other numbers n' relatively prime to n. The random behavior of prime numbers, observed many times in the literature, suggests that if the numbers n' are randomly selected their factorization is useless for the problem of factoring n. However this might not be so relevant to malleability because we have the freedom to select cleverly the additional numbers n'.

Indeed, the result contained in this note goes against the non-malleability intuition, thus showing how subtle this notion is. Concretely, for any number n we are able to prove the existence of a polynomial time reduction algorithm from factoring n to factoring certain explicit numbers n', all relatively prime to n. In other words, we show that factoring is, in this generality, a malleable problem.

Let us stress that this might be compatible with the conjecture of [9] mentioned above because imposing extra conditions on the numbers n' may result in transforming the problem in a non-malleable one. In fact, it is our belief that malleability is a notion that depends strongly of these kind of extra conditions, and hence requires further research.

The algorithm

Given an RSA modulus $n = pq$, we want to find n' such that factoring n', with the help of an oracle, will allow us in finding the factorization of n. In fact we will only need very partial information about the factorization of n' in order to get the complete factorization of n. From now on, and without loose of generality, we will make the assumption that $p < q$.

A particular case

By construction, (which will be clear in a moment), it turns out that the particular case in which n is such that $2^{p-1} \not\equiv 1 \pmod{q}$ or $2^{q-1} \not\equiv 1 \pmod{p}$ is somehow simpler and we will dedicate this section to it. However, the whole idea of the method will arise in this case and so the general one, considered in the next section, will be very similar. We consider $n' = 2^n + 1$. Observe that an efficient encoding of n' of size comparable to n is available since all these numbers in binary form have a 1 at the beginning and end, and the rest are precisely $n-1$ zeros. Let us assume the existence of an oracle $\mathcal{O}$ which, on input n', returns the residue class modulo n of three prime factors $r|n'$. In fact, the only thing we need is the residue class of just one factor of n' modulo n different from 1 and 3 so, if convenient, one can admit an oracle answering any set $S \subset \{r \pmod{n} : r \text{ prime }, r|n'\}$, $S \not\subseteq \{1,3\}$ and polynomial size. We now present an algorithm which on the input and RSA modulus n in the conditions of this section, outputs a nontrivial factor of n.

Algorithm 1.

- *Send* $n' = 2^n + 1$ *in binary form to* $\mathcal{O}$.
- *Take* $r \in S$, $r \neq 1,3$, *and compute* $d = (r-1,n)$.

Theorem 2. *Let* $n = pq$ *be and RSA modulus such that either* $2^{p-1} \not\equiv 1 \,(mod\ q)$ *or* $2^{q-1} \not\equiv 1 \,(mod\ p)$. *Then the number* d *given by the previous algorithm, in polynomial time in* $\log n$, *is a prime divisor of* n.

Proof: The first thing we have to prove is that there exists a set S satisfying the conditions of the algorithm. In order to do so we have to prove that at least one prime factor of n' is not 1 or 3 modulo n. Suppose r is a prime factor of n'. Then, $2^{2n} \equiv 1 \pmod{r}$ and so, either $r = 3$ which always divides n', or the order of 2 in $\mathbb{F}_r^*$ is $\text{ord}_r(2) = p, q, 2p, 2q, pq$ or $2pq$. In this case we just have to recall that the order of any element must divide the order of the group to conclude that either $p|(r-1)$, $q|(r-1)$ or $n|(r-1)$. Note, on the other hand that 9 never divides n' since $n \equiv \pm 1 \pmod{6}$ and so $2^n \equiv 2$ or 5 modulo 9. Hence, If $n|(r-1)$ for any $r|n'/3$, then each factor of $n'/3$ is 1 modulo n and so $n'/3 \equiv 1 \pmod{n}$ which is the same as saying $2^{n-1} \equiv 1 \pmod{n}$. This is impossible since in particular $2^{n-1} \equiv 2^{p-1} \pmod{q}$ and $2^{n-1} \equiv 2^{q-1} \pmod{p}$. Hence there exists $r_0|n'$ such that $r_0 \not\equiv 1 \pmod{n}$. Observe also that any such factor verifies $r_0 \equiv 1 \pmod{p}$ or $r_0 \equiv 1 \pmod{q}$ and, in particular, $r_0 \not\equiv 3 \pmod{n}$.

□

The previous algorithm would work, in particular, for any modulus $n = pq$ such that $(p-1, q-1) = D$ is small, for example $D < \log_2(n)$. Indeed, if $2^{p-1} \equiv 1 \pmod{q}$ and $2^{q-1} \equiv 1 \pmod{p}$, then

$2^D \equiv 1 \pmod{n}$ which is impossible for $D < \log_2(n)$. This fact leads to the interesting observation that even the probability that $D > \log_2(n)$ tends to zero with n. This is the content of the following proposition

Proposition 3. *For any positive z we have*

$$\sum_{\substack{z \le p,q < 2z \\ (p-1,q-1) > \log z}} 1 \le \left(\frac{z}{\log z}\right)^2 \frac{(\log\log z)^2}{\log z},$$

where the sum runs over the prime numbers in the interval.

Remark: Before proving the proposition, let us observe that we just have to use the Prime Number Theorem to obtain $\sum_{z \le p,q < 2z} 1 \sim (\mathrm{z}/\log z)^2$ and hence, the probability of finding a pair of primes in the interval $[z, 2z]$ which do not satisfy the conditions in Theorem 2 tends to zero faster than $(\log\log z)^2/\log z$. Also note that even if $(p-1, q-1)$ would be big, we still would need 2 to have order D modulo p and modulo q which one expects to be false for many pairs of primes by Artin's conjecture, (cf. [8]).

Proof of Proposition 3. Given a positive z big enough, let

$$\pi(d;z) = \sum_{\substack{p \equiv 1 \pmod{d} \\ z \le p < 2z}} 1.$$

Then, the number of pairs of primes $z \le p, q < 2z$ such that $(p-1, q-1) = d > \log z$ is bounded above by

$$\sum_{\log z < d < z} \sum_{\substack{p,q \equiv 1 \pmod{d} \\ z \le q < p < 2z}} 1 < \sum_{\log z < d < z^\alpha} \pi(d;z)^2 + \sum_{z^\alpha < d < z} \pi(d;z)^2 = S_1 + S_2,$$

for any $0 < \alpha < 1$. For the second term we get trivially the bound $S_2 < 4z^{3-2\alpha}$. To estimate S_1 let us first introduce the following useful notation. We will write $E(d;z) = \pi(d;z) - z/(\varphi(d)\log z)$, as the error in the approximation of the number of primes in the congruence 1 modulo n by the total number of primes divided by the number of congruences. Then,

$$S_1 = \sum_{\log z < d < z^\alpha} \left(\frac{z}{\varphi(d)\log z} + E(d,z)\right)^2 =$$

$$\left(\frac{z}{\log z}\right)^2 \sum_{\log z < d < 2z} \frac{1}{\varphi(d)^2} + \sum_{\log z < d < z^\alpha} (E(d,z))^2 + 2 \sum_{\log z < d < z^\alpha} \frac{z}{\varphi(d)\log z} E(d,z).$$

We can use now Cauchy-Schwartz inequality to get, for the last sum above

$$\sum_{\log z < d < z^\alpha} \frac{z}{\varphi(d)\log z} E(d,z) \le \left(\sum_{\log z < d < z^\alpha} \left(\frac{z}{\varphi(d)\log z}\right)^2\right)^{1/2} \left(\sum_{\log z < d < z^\alpha} (E(d,z))^2\right)^{1/2}. \qquad (1)$$

We are in the correct position to use the Barban-Davenport-Halberstam Theorem for primes in artihmetic progressions, (cf. page 421, [7]), which we now include for convenience.

Theorem 4. *(Barban-Davenport-Halberstam) We have*

$$\sum_{d \le z^{1-\varepsilon}} (E(d;z))^2 \ll z^2/(\log z)^A,$$

for any $A > 0$, and $\varepsilon > 0$, where the implied constant only depends on A and ε.

Substituting the above inequality in S_1, putting $A = 4$ and $\varepsilon = 1/4$, and using (1) we get for some constant C

$$S_1 \leq \left(\frac{z}{\log z}\right)^2 \sum_{\log z<d<2z} \frac{1}{\varphi(d)^2} + \frac{Cz^2}{(\log z)^4} + \frac{Cz}{(\log z)^3} \left(\sum_{\log z<d<2z} \frac{1}{\varphi(d)^2}\right)^{1/2}.$$

To finish the proof of the Proposition we just have to note that

$$\varphi(d) = d\prod_{p|d}(1-1/p) > d\prod_{p<d}(1-1/p) > Cd/\log d,$$

by Mertens Theorem (cf. p.34, [7]) and so

$$\sum_{\log z<d<2z} \frac{1}{\varphi(d)^2} \leq C \sum_{\log z<d} \left(\frac{\log d}{d}\right)^2 \leq C_1 \frac{(\log\log z)^2}{\log z},$$

for some constants C, C_1. The result follows.

The general case

For a few pairs of primes, it could happen that the order of 2 in $\mathbb{F}_q^*$ and $\mathbb{F}_p^*$ was a divisor of D and, in that case, 2^n is indeed 2 modulo n which could make Algorithm 1 fail. To avoid this problem, instead of 2, we will choose a primitive root of $\mathbb{F}_q^*$, g, to build our test number $n' = g^n + 1$. It is very easy to see that the number of primitive roots of $\mathbb{F}_q^*$ is $\phi(q-1)$, hence, the probability for an integer m to be a primitive root verifies

$$\frac{\varphi(q-1)}{q-1} = \prod_{p|(q-1)}\left(1-\frac{1}{p}\right) > \prod_{p<q}\left(1-\frac{1}{p}\right) \sim \frac{e^{-\gamma}}{\log q},$$

again by Mertens theorem. In other words, a set of size $C\log q$ of integers contains a primitive root modulo q with probability as close to one as we want, making the constant C big enough. To see this, note that the probability for a random set of this size to contain no primitive roots would be $(1-1/(e^\gamma \log q))^{C\log q} \sim e^{-C/e^\gamma}$. In this sense Bach, in [1], made a much more accurate heuristic argument to claim that the least primitive root modulo p, which we will call $g(p)$ should verify $g(p) \leq e^\gamma \log p(\log\log p)^2(1+\varepsilon)$ for almost all p. Although this fact is not yet proved, there are conditional results which certify the truth of the statement. In particular we will mention the following result of V. Shoup in [12] proved under the Grand Riemann Hypothesis, GRH from now on.

Theorem 5. *(Shoup) Let p be a prime and denote $g(p)$ as the least positive integer which is a generator of $\mathbb{F}_p^*$. Then, if GRH is true, $g(p) = O((\log p)^6)$.*

Observe that, although far from the expected result, $g(p) = O((\log p)^6)$ is still of polynomial size and, hence, good enough for our purposes. It is worth mentioning that Heath-Brown was able to prove in [5] that among $2,3,5$ there is a primitive root for infinitely many primes p. Let us now describe the algorithm.

For convenience we will call $c \in \{0,1\}^{n+2}$ the binary encoding of $2^n + 1$. We will take advantage of the fact that the m-ary representation of the numbers $m^n + 1$ is always c, independent of m. Let $n'_m = m^n + 1$ and consider the function $\omega(n)$ counting the number of distinct prime factors of n. Assume the existence of an oracle $\mathcal{O}$ which, on input (c,m), returns a set of residue classes S of size $|S| = \omega(m) + 2$ when such a set S exists, and otherwise returns $\perp$. Again, the only thing we need is the residue class of just one factor of n'_m modulo n different from 1 and the classes of the prime divisors of $m+1$. Hence, if convenient, we can consider the set S to be of polynomial size such that

$S \subset \{r \,(\mathrm{mod}\, n) \,:\, r \text{ prime }, r|n'_m\}$, $S \not\subseteq S_m \cup \{1\}$ where $S_m = \{r \,(\mathrm{mod}\, n) \,:\, r \text{ prime }, r|(m+1)\}$. The following algorithm on the input of an RSA modulus n outputs a nontrivial factor of n.

Algorithm 6.

1. *$m = 2$*

2. *Send (c, m) to $\mathcal{O}$.*

3. *If $S = \perp \Rightarrow m = m+1$ and go to (2). Else,*

4. *Take $r \in S$, $r \neq S_m \cup \{1\}$, and compute $d = (r-1, n)$.*

Theorem 7. *Let $n = pq$ be an RSA modulus. If GRH is true then the Algorithm* 6 *runs in polynomial time and the number d given by it is a prime divisor of n.*

Proof: By Theorem 5 we can assume that m is a primitive root modulo q, at a polynomial time cost. Then $m^{p-1} \not\equiv 1 \,(\mathrm{mod}\, q)$, since $p < q$. Hence, in a similar way as in the proof of Theorem 2 we have to prove that a certain prime factor r of n'_m belongs to a residue class modulo n not in $S_m \cup \{1\}$. We will use the following straightforward lemma.

Lemma 8. *Let n be an RSA modulus. For any integer m, such that $(m+1, n) = 1$ we have $((m^n + 1)/(m+1), m+1) = 1$.*

Proof: Observe that if $r|(m+1)$, then

$$(m^n+1)/(m+1) = \sum_{j=0}^{n-1} (-m)^j \equiv \sum_{j=0}^{n-1} 1 \,(\mathrm{mod}\, r) \equiv n \,(\mathrm{mod}\, r)\,.$$

□

Now, analogously to what we did in the proof of Theorem 2, if $r|n'_m$ then $m^{2n} \equiv 1 \,(\mathrm{mod}\, r)$, and so $\mathrm{ord}_r(m) = 2, p, q, 2p, 2q, pq$ or $2pq$ and clearly $\mathrm{ord}_r(m) \neq 2$ for any r a prime factor of $(m^n + 1)/(m+1)$. To see this use Lemma 8 and observe that if $r|(m-1)$ then $m^n + 1 \equiv 2 \,(\mathrm{mod}\, r)$. Hence, as in the previous section, for any $r|(m^n+1)/(m+1)$ then either $p|(r-1)$, $q|(r-1)$ or $n|(r-1)$. If $r \equiv 1 \,(\mathrm{mod}\, n)$ for any $r|(m^n+1)/(m+1)$ then $m^{n-1} \equiv 1 \,(\mathrm{mod}\, n)$, which is impossible for m a primitive root modulo q since $m^{n-1} \equiv m^{p-1} \,(\mathrm{mod}\, q)$. The proof of the theorem concludes as in Theorem 2.

References

[1] E. Bach, Comments on search procedures for primitive roots, Math. of Computation, vol 66-220, pg 1719–1727, 1997.

[2] M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols. In ACM CCS'93, pp. 62–73, 1993.

[3] D. Boneh, Simplified OAEP for the RSA and Rabin functions. In CRYPTO'01, LNCS 2139, pp. 275–291. Springer Verlag, 2001.

[4] S. Goldwasser, S. Micali and R. L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. on Comp., 17(2):281–308, April 1988.

[5] R. Heath-Brown, A remark on Artin's conjecture. Quart. J. Math. Ser. 37(2), 27–38, Oxford 1986.

[6] P. Pailler, IPAM Workshop: Number Theory and Cryptography–Open Problems, October 9-13, Los Angeles, CA, USA.

[7] H. Iwaniec and E. Kowalski, Anlytic number theory, Colloquim publications Vol 53 of the AMS, 2004

[8] M. R. Murty, Artin's conjecture for primitive roots. Math. Intelligencer, 10 (1988), no. 4, 59–67.

[9] P. Paillier and J. L. Villar, Trading One-Wayness Against Chosen-Ciphertext Security in Factoring-Based Encryption, Asyacrypt 2006.

[10] M. O. Rabin, Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Jan. 1979.

[11] H. C. Williams, A modification of the RSA public-key encryption procedure. IEEE Transactions on Information Theory, IT-26(6):726–729, 1980.

[12] V. Shoup, Searching for primitive roots in finite fields. Math. Comp. 58.197, 369–380, (1992).

J. Jiménez Urroz Universitat Politècnica de Catalunya
jjimenez@ma4.upc.edu

Linear algebra for index calculus based discrete logarithm computations

Antoine Joux

Abstract

Most recent algorithms for computing discrete logarithms in various groups are based on index calculus. These algorithms first contruct many sparse linear equations using a fast and efficiently parallelizable technique such as sieving. In a second phase, one need to find a non trivial solution to the large resulting system of equations modulo the order of the multiplicative group.

Currently, this second phase, performing linear algebra, is the most difficult in practice. This is due to the fact that this phase is much harder to parallelize. However, the overall strategy to perform it has remained the same for decades. Start by reducing the size of the linear system using a technique called structured Gaussian elimination. Then use an iterative algorithm to solve the reduced system using a reasonable amount of memory.

The goal of this paper is to describe the current state of the art when programming this linear algebra phase on a large parallel computer.

Introduction

To compute discrete logarithms in finite groups, there exists several type of algorithms. The type contains generic algorithms which work for arbitrary groups and do not rely on any specific property of the group encoding. These generic algorithms include the Pohlig-Hellman algorithm which shows that computing the discrete logarithm in a group can be performed by computing a few discrete logarithms in its prime order subgroup. Moreover, in prime order generic groups, discrete logarithms can be computing by algorithms whose running time is of the order of the square-root of the group order. The simplest of these algorithms is the baby-step giant-step method, however, memoryless algorithms are usually prefered.

The second type which we mostly interest us in this paper contains index-calculus based algorithms. These algorithms only apply to specific group encodings and they use the details of the encoding to produce a decomposition of the identity element as a product (or sum if the group is presented additively) of elements taken in a subset $\mathcal{B}$ of relatively small size. This subset is often called the smoothness-basis or the decomposition basis. Taking logarithm, each decomposition yields a linear equation between the logarithms of the elements of $\mathcal{B}$ modulo the group order. If enough equations are collected, one expects to obtained a system of linear equations with a kernel of dimension 1. As a consequence, any non-zero solution of the system yields discrete logarithm values for all elements of $\mathcal{B}$. To fix the basis of the logarithms to be some fixed element $g_0 \in \mathcal{B}$, it suffices to divide the obtained solution vector by its value at g_0.

An important property of index calculus-based algorithm is that the produced equations have low weight, i.e., each equation only contains a small number of elements for $\mathcal{B}$. Moreover, the non-zero coefficients that appear in the equations are generally small[1].

The usual strategy to solve such systems of equations is to proceed in two steps. The first step reduces the size of the system while somewhat degrading its sparsity using structured Gaussian elimination. The second step uses an iterative algorithm, such as Lanczos' or Wiedemann's algorithm to solve the resulting system. The advantage of these algorithms is that they compute a solution vector without operating on the matrix itself, just by performing matrix-vector products and vector operations. As a consequence, the amount of memory is much smaller than what would be required

[1] With some index calculus based algorithms, a few special elements of $\mathcal{B}$ may appear wih large coefficients. In this case, these special elements are usually dense, i.e. they appear in a large number of equations.

by full Gaussian elimination, where dense matrices of the same size as our reduced matrix would appear.

This paper presents the algorithmic details behind the linear algebra necessary for a large index-calculus based discrete logarithm computation on an elliptic curve defined over a sextic extension [4]

Structured Gaussian Elimination

The idea of structured gaussian elimination for index calculus algorithm was first proposed by Odlyzko in [7] and further developed in [5]. It consists in performing a certain number of well-chosen pivoting step. The only difference with the pivoting steps occuring in regular gaussian elimination is that the pivots are chosen to minimize the growth of the matrix size during elimination. This is done by chosing as pivot a variable x_i in an equation E_j, such that:

1. The coefficient before x_i in E_j is either 1 or -1.
2. The product $(t_i - 2) \cdot (\ell_j - 2)$ is minimal, where t_i is the number of occurences of x_i and ℓ_j the number of sparse variables that appear in E_j.

The partitioning of variables in sparse and dense variables is one of the numerous heuristic choices which are required when implementing the algorithm. The available options are wide and, in some cases, it is even possible to work with an empty set of dense vaiables.

An important fact to remember is that if the line E_j is deleted after pivoting and assuming that no variable except x_i is canceled during a pivot step, we can see [3, Section 3.4.2] that the size of the sparse part matrix increases by $(t_i - 2) \cdot (\ell_j - 2) - 2$.

Large primes variation. A special case of interest is to consider only pivots with $t_i \leq 2$ or $\ell_j \leq 2$. In that special case, gaussian elimination can be performed in a very efficient way using graph based techniques. First, we preprocess the linear system by removing all equations with $\ell_j > 2$ and all the variables with $t_i = 1$ together with the equation they appear in. We also remove all variables with $t_i = 2$ by removing the two equations they appear in and replacing them by an adequate linear combination that cancels x_i. Note that the resulting equation contains at most 2 sparse variables. This is repeated until no more variables or equations can be removed. Once this is done, we can build a graph whose nodes are labelled by the sparse variables, together with an extra “empty” node. We draw a vertice between two sparse variables if they appear in a common equation and we draw a vertice between a variable and the empty node if the variable appears alone in some equation.

It is clear that any linear combination of equations that cancels all sparse variables corresponds to a cycle in the above graph. The converse is not true, however, if all coefficients are 1 or -1 half the cycles yield a linear combination.

This special case of gaussian elimination is usually known as the large prime variation. The sparse variables are called “small primes” and the dense variables are called “large primes”. These names are inherited from the number field sieve algorithm (in particular see [6]). When there is no natural choices of small primes versus large primes, the method can still be applied by partitioning the variables in a random fashion. It has been introduced in this form in [2]

A very nice property of large prime variation, shown in [2], is that it is possible to analyze its asymptotic behavior nicely under some reasonable condition about the distribution of the variables in the equations.

General structured gaussian elimination. In the general case, structured gaussian elimination starts by selecting a partition of the variables into sparse and dense variables. A simple and efficient approach already hinted at in [5] is to count the number of occurences of each variables and to declare a variable sparse when its number of occurence is smaller than some threshold.

Once this is done, it is possible to devise an algorithm that efficiently keeps track of the products $(t_i - 2) \cdot (\ell_j - 2)$, selects the best possible current pivot and updates the matrix in memory. However, this has several drawback: first, the data structures involved are complex and costly; second, all indermediate matrices must fit into the main memory; finally, this process is hard to debug and not efficiently parallelizable. For all these reasons, it is important to propose a different approach to structure gaussian elimination.

Let us introduce two new ingredients: simultaneous independent pivots and lazy pivoting.

With these two ingredients, structured gaussian elimination can be done more efficiently for large matrices. Indeed, due to the independance of simultaneous pivots, it is possible to process several subsets of equations on different processors in parallel without using too much communications. Moreover, thanks to the lazy evaluation, the original equations are never modified and we only need to keep track of the position of the successive pivots. As a consequence, the original matrix does not need to be fetched into memory and can remain oin disk. This greatly increases the size of the manageable systems and also permit to deal with matrices with a very large number of extraneous equations quite efficiently.

Block Wiedemann algorithm

The computations presented in [4] reached the limits of our implementation of Lanczos's algorithm. The difficulty with this algorithm is that consecutive matrix-vector products are inherently sequential and that using block Lanczos modulo large prime does not solve the problem because it requires more scalar products between large vectors.

The block Wiedemann algorithm, introduced in [1], offers a nice solution to the parallelization issue. This is an algorithm that comprises three consecutives phases. The first phase computes k independent matrix-vector product sequences. Each of the sequence is initialized with a independent random vector and the matrix-vector product is applied about $2N/k$ times, where N is the dimension of the matrix. The k first coordinates of each vector (or, more generally, the scalar products of these vectors with k fixed vectors randomly chosen at the beginning of the algorithm) are assembled into $k \times k$ matrices at each step.

The second phase search for a linear relationship that holds between the columns of the $k \times k$ matrices appearing in any shifted windows of n/k such matrices. It is highly probable that such a linear relation also holds on the full output vectors.

The third phase computes the vectorial value of the linear relation starting from the random starting points (and not from their images by the matrix). Due to phase two, applying the matrix once to this combination should output the null vector. Moreover, the combination itself has no special reason to be null. As a consequence, we obtain a kernel element of the matrix.

The first and third phase of block Wiedemann can easily parallelized on k independent computers. The interesting part is the second phase of block Wiedemann which can be performed by a variation of Berlekamp-Massey algorithm. However, there is a much more efficient option, described by Thomé in [8]. This efficient method can be expressed in a simple way, using matrix-univariate polynomials, i.e. polynomials in X whose coefficients are matrices or, equivalently, matrices whose entries are polynomials in X. The data generated by the first phase can be compacted into such a matrix-polynomial F of degree $< D$ (as a matrix, F is square of dimension k) and we seek matrix-polynomials f and g of small degree (around $D/2$) such that:

$$f \cdot F + g = 0 \pmod{X^D}.$$

To explain Thomé's algorithm it is useful to generalize it slightly. Given two matrix-polynomials of F and G degree $< D$, find a linear basis of the ideal containing all the polynomials (f,g) such that:

$$f \cdot F + g \cdot G = 0 \pmod{X^D}.$$

The original problem is the simple case where G is the constant polynomial equal to the identity matrix.

Let H be a square matrix-polynomial of dimension $2k$ that spans this ideal. Then H has full rank and:

$$H \cdot \begin{pmatrix} A \\ B \end{pmatrix} = 0 \pmod{X^D}.$$

To construct H efficiently, we proceed recursively. First, we compute H_1 as a solution to the same problem at degree $D_1 = \lceil D/2 \rceil$. This can be done by working on truncated versions of A and B. Then, we define A_1 and B_1 by:

$$\begin{pmatrix} X^{D_1} \cdot A_1 \\ X^{D_1} \cdot B_1 \end{pmatrix} = H_1 \cdot \begin{pmatrix} A \\ B \end{pmatrix} \pmod{X^D}.$$

Then, we compute H_2 to be the result of the algorithm when applied to A_1 and A_2, which have degree $D - D_1$. Finally, we obtain H as the product[2] $H_2 \cdot H_1$.

The complexity of the recursive algorithm we obtain is dominated by the complexity of multiplying matrix-polynomials of dimension $2k$. Using fast Fourier transforms techniques and textbook matrix multiplication, this yields a total complexity of $O(k^2 \cdot D \log(D)(k + \log D))$ arithmetic operations.

To terminate the recursion, we need to solve the problem when F and G have degree 0. This can simply be done using gaussian elimination. For example, if F and G have degree 0 and are both invertible matrices, then we find:

$$H = \begin{pmatrix} F^{-1} & -G^{-1} \\ X & 0 \end{pmatrix}$$

It is interesting to note that the product of two matrix-polynomial of this form is a matrix-polynomial of degree 1. Thus, in the generic case, the matrix H obtained when F and G have degree 2. Similarly, for generic matrix-polynomials of degree $2D$, the resulting H has degree D. This remark can be used to reduce the degree of intermediate matrix-polynomial in an implentation of Thomé's algorithm.

References

[1] D. Coppersmith. Solving homogeneous linear equations over $gf(2)$ via block wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.

[2] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comput.*, 76(257):475–492, 2007.

[3] A. Joux. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC, 2009.

[4] A. Joux and V. Vitse. Cover and decomposition index calculus on elliptic curves made practical - application to a previously unreachable curve over $\mathbb{F}_{p^6}$. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 9–26, 2012.

[5] B. A. LaMacchia and A. M. Odlyzko. Solving large sparse linear systems over finite fields. In *CRYPTO'90*, volume 537 of *Lecture Notes in Computer Science*, pages 109–133, 1991.

[6] A. K. Lenstra and M. S. Manasse. Factoring with two large primes. In *EUROCRYPT'90*, volume 473 of *Lecture Notes in Computer Science*, pages 72–82, 1991.

[7] A. M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In *EUROCRYPT'84*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314, 1985.

[2] The full truth is more complex. Indeed, $H_2 \cdot H_1$ may generate a proper sub-ideal. To obtain the full H, we must track the degrees of the polynomials in H_1 and H_2 and use linear changes in order to minimize the degree of the resulting H.

[8] E. Thomé. Subquadratic computation of vector generating polynomials and improvement of the block wiedemann algorithm. *J. Symb. Comput.*, 33(5):757–775, 2002.

A. Joux DGA and Université de Versailles
antoine.joux@m4x.org

Untangling attribution: understanding the requirements for network attribution
Susan Landau

This talk is based on a joint work with David Clark.

As a result of increasing spam, DDoS attacks, cybercrime, and data exfiltration from corporate and government sites, there have been multiple calls for an Internet architecture that enables better network attribution at the packet layer. The intent is for a mechanism that links a packet to some packet level personally identifiable information. But cyberattacks and cyberexploitations are more different than they are the same. One result of these distinctions is that packet-level attribution is neither as useful nor as necessary as it would appear. In this talk, I analyze the different types of Internet-based attacks, and observe the role that currently available alternatives to attribution already play in deterrence and prosecution. I focus on the particular character of multi-stage network attacks, in which machine A penetrates and “takes over” machine B, which then does the same to machine C, etc. and consider how these types of attacks might be traced, and observe that any technical contribution can only be contemplated in the larger regulatory context of various legal jurisdictions.

S. Landau Harvard University
susan.landau@privacyink.org

An application of symmetric functions to cryptology

Andrzej Schinzel

We consider Shamir's secret sharing schemes over finite fields, with the secret placed as any coefficient a_i of the scheme polynomial of degree $k-1$, determined by a sequence of pairwise different public identities, called a track. If the sequence defines a k-out-of-n Shamir's secret sharing scheme then the track is called (k,i)-admissible. If it is (k,i)-admissible for all i we call it k-admissible. Using some estimates for the elementary symmetric polynomials, we shall show that the track $(1,\dots,n)$ is practically always k-admissible, i.e., the scheme allows to place the secret as an arbitrary coefficient of its generic polynomial even for relatively small p. Here k is the threshold and n the number of shareholders in the scheme.

References

[1] A. Schinzel, S. Spiez and J. Urbanowicz, Elementary symmetric polynomials in Shamir's scheme, J. Number Theory, 130(2010), 1572–1580.

A. Schinzel Polish Academy of Science
schinzel@impan.pl

On a hidden shift from powers
Igor Shparlinski

This talk is based on a joint work with Jean Bourgain, Moubariz Z. Garaev, and Sergei V. Konyagin.

Introduction

Set-up and Motivation

Let $\mathbb{F}_q$ be a finite field of q elements.

For $e \mid q-1$ and $s \in \mathbb{F}_q$ we denote by $\mathcal{O}_{e,s}$ an oracle that on every input $x \in \mathbb{F}_q$ outputs $\mathcal{O}_{e,s}(x) = (x+s)^e$ for some "hidden" element $s \in \mathbb{F}_q$.

We consider the *Hidden Shifted Power Problem*:

> given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_q$, find s.

Furthermore, we also consider the following two versions of the *Shifted Power Identity Testing*:

> given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_q$ and known $t \in \mathbb{F}_q$, decide whether $s = t$ provided that the call $x = -t$ is forbidden;

and

> given two oracles $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$ for some unknown $s,t \in \mathbb{F}_q$ decide whether $s = t$.

These problems are special cases of the more general problems of oracle (also sometimes called "black-box") polynomial interpolation and identity testing for arbitrary polynomials, see [2] and references therein.

Clearly, the knowledge of $(x+s)^e$ is equivalent (modulo solving a discrete logarithm problem in the subgroup of $\mathbb{F}_q$ of order $(q-1)/e$) to the knowledge of $\chi(x+s)$ for some fixed multiplicative character χ of $\mathbb{F}_q^*$, see [9, 10, 17], where several classical and quantum algorithms for this and some other similar problems are given. The Hidden Shifted Power Problem, under the name of *Hidden Root Problem*, has also been re-introduced by Vercauteren [20] in relation to the so-called fault attack on pairing based cryptosystems on elliptic curves.

Although for application to pairing based cryptography the Hidden Shifted Power Problem usually appears in extension fields $q = p^k$ with $k > 1$, it has been shown by Koblitz and Menezes [14] that there are elliptic curves that lead to the case of prime fields, that is, $q = p$, on which we concentrate in this work.

For a prime $q = p \geq 3$ and $e = (p-1)/2$ the Hidden Shifted Power Problem has several other links to cryptography, and been considered in a number of works, see [1, 3, 11, 13] and references therein.

Naive Approaches

Certainly the most straightforward approach is to query $\mathcal{O}_{e,s}$ on $e+1$ arbitrary elements $x \in \mathbb{F}_q$ and then interpolate the results. Using a fast interpolation algorithm, see [12] leads to a deterministic algorithm of complexity $e(\log q)^{O(1)}$. For the Shifted Power Identity Testing, there is also a trivial probabilistic algorithm that is based on querying $\mathcal{O}_{e,s}$ (and $\mathcal{O}_{e,t}$) at randomly chosen elements $x \in \mathbb{F}_q$.

Our Approach

Let $\mathcal{G}_e \subseteq \mathbb{F}_q^*$ be the multiplicative group of order $e \mid q-1$, that is, $\mathcal{G}_e = \{\mu \in \mathbb{F}_q \ : \ \mu^e = 1\}$. We now define the polynomials

$$F_{s,t}(X) = \prod_{\mu \in \mathcal{G}_e} (X + s - \mu(X+t)) .$$

Our approach is based on the idea of choosing a small "test" set $\mathcal{X}$, which nevertheless is guaranteed to contain at least one non-zero of the polynomial $F_{s,t}$ for any $s \neq t$. This is based on a careful examination of the roots of $F_{s,t}$ and relating it to some classical number theoretic problems about the distribution of elements of small subgroups of finite fields.

Clearly, if $F_{s,t}(x) = 0$ for some $x \in \mathbb{F}_q^*$ then

$$\frac{x+s}{x+t} \in \mathcal{G}_e \tag{1}$$

(provided $x + t \neq 0$). If t is known, then we can choose the "test" set $\mathcal{X}$ in the form

$$\mathcal{X} = \{y^{-1} - t \ : \ y \in \mathcal{Y}\}$$

for some set $\mathcal{Y} \subseteq \mathbb{F}_q^*$. Then the condition (1) means that a shift of $\mathcal{Y}$ is contained inside of a coset of $\mathcal{G}_e$, that is $\mathcal{Y} + r \subseteq r\mathcal{G}_e$, where $r = (s-t)^{-1}$.

So our goal is to find a "small" set $\mathcal{Y} \subseteq \mathbb{F}_q^*$ such that its shifts cannot be inside of any coset of $\mathcal{G}_e$ (we note that the value of r is unknown). Questions about the distribution of cosets of multiplicative groups have been considered in a number of works and have numerous applications, see [15] and also [4, 5, 8, 6, 7, 16, 18, 19] for several more recent results and applications to cryptographic and computational number theory problems.

Our Results

Hidden Shifted Power Problem

Here we present some deterministic and probabilistic algorithms for the Hidden Shifted Power Problem that runs in about the same time as the interpolation algorithm, but use significantly less oracle calls.

Theorem 1. *For a prime p and a positive integer $e \mid p-1$ with $e \leq p^{1-\delta}$ for some fixed $\delta > 0$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$ and ℓ-th power nonresidues for all prime divisors $\ell \mid e$, there is a deterministic algorithm that for any fixed $\varepsilon > 0$ makes $O(1)$ calls to the oracle $\mathcal{O}_{e,s}$ and finds s in time $e^{1+\varepsilon}(\log p)^{O(1)}$.*

Theorem 2. *For a prime p and a positive integer $e \mid p-1$ with $e \leq p^{1-\delta}$ for some fixed $\delta > 0$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$, there is a deterministic algorithm that for any fixed $\varepsilon > 0$ makes $O(1)$ calls to the oracle $\mathcal{O}_{e,s}$ and finds s in time $O(ep^{\varepsilon})$.*

Moreover, under the Extended Riemann Hypothesis one can finds s in time $e^{1+\varepsilon}(\log p)^{O(1)}$.

The following result is applicable to the case when e does not satisfy the restriction in Theorems 1 and 2 (namely, to $e = p^{1+o(1)}$ as $p \to \infty$).

Theorem 3. *For a prime p and a positive integer $e \mid p-1$ with $e \leq (p-1)/2$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$, there is a deterministic algorithm that makes $O(\log p/(\log(p/e)))$ calls to the oracle $\mathcal{O}_{e,s}$ and finds s in time $p(\log p)^{O(1)}$.*

We now present a probabilistic algorithm which is slightly more efficient in some cases.

Theorem 4. *For a prime p and a positive integer $e \mid p-1$ with $e \leq p^{1-\delta}$ for some fixed $\delta > 0$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$, there is a probabilistic algorithm that makes in average $O(1)$ calls to the oracle $\mathcal{O}_{e,s}$ and finds s in the expected time $e(\log p)^{O(1)}$*

Shifted Power Identity Testing with Known t

We recall that for the shifted power identity testing with known t the call $x = -t$ is forbidden.

Theorem 5. *For a prime p and a positive integer $e \mid p-1$ with $e \le p^{1-\delta}$ for some fixed $\delta > 0$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$ and known $t \in \mathbb{F}_p$, there is a deterministic algorithm to decide whether $s = t$ in time $e^{1/4+o(1)}(\log p)^{O(1)}$ as $e \to \infty$.*

For large values of e we can use bounds of character sums.

Theorem 6. *For a prime p and a positive integer $e \mid p-1$ with $e \le (p-1)/2$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$ and known $t \in \mathbb{F}_p$, there is a deterministic algorithm to decide whether $s = t$ in time $p^{1/4+o(1)}$ as $p \to \infty$.*

Collecting the results of Theorems 5 and 6, we obtain an algorithm of complexity $e^{1/4}p^{o(1)}$ for any $e \le (p-1)/2$.

For small values of e, we have

Theorem 7. *For a prime p and a positive integer $e \mid p-1$ with $e \le p^{\delta}$ for some fixed $\delta > 0$, given an oracle $\mathcal{O}_{e,s}$ for some unknown $s \in \mathbb{F}_p$ and known $t \in \mathbb{F}_p$, there is a deterministic algorithm to decide whether $s = t$ in time $e^{c_0\delta}(\log p)^{O(1)}$, where c_0 is some absolute constant.*

Shifted Power Identity Testing with Unknown t

For large values of e we have the following simple result.

Theorem 8. *For a prime p and a positive integer $e \mid p-1$ with $e \le (p-1)/2$, given two oracles $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$ for some unknown $s,t \in \mathbb{F}_p$, there is a deterministic algorithm to decide whether $s = t$ in time $p^{1/2+o(1)}$.*

For $e \le p^{3/4}$ we have a stronger result.

Theorem 9. *For a prime p and a positive integer $e \mid p-1$ with $e \le (p-1)/2$, given two oracles $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$ for some unknown $s,t \in \mathbb{F}_p$, there is a deterministic algorithm to decide whether $s = t$ in time $\max\{e^{1/2}p^{o(1)}, e^2p^{-1+o(1)}\}$.*

Finally, for very small values of e we obtain:

Theorem 10. *For a prime p and a positive integer $e \mid p-1$ with $e \le p^{\delta}$ for some fixed $\delta > 0$, given two oracles $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$ for some unknown $s,t \in \mathbb{F}_p$, there is a deterministic algorithm to decide whether $s = t$ in time $e^{c_0\delta^{1/3}}(\log p)^{O(1)}$, where c_0 is some absolute constant.*

References

[1] M. Anshel and D. Goldfeld, ‘Zeta functions, one-way functions, and pseudorandom number generators’, *Duke Math. J.*, **88** (1997), 371–390.

[2] M. Beecken, J. Mittmann and N. Saxena, ‘Algebraic independence and blackbox identity testing’, *Electronic Coll. Comput. Compl.*, Report No. 22, (2011), 1–32.

[3] D. Boneh and R. Lipton, ‘Algorithms for black-box fields and their applications to cryptography’, *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **1109** (1996), 283–297.

[4] J. Bourgain, ‘On the distribution of the residues of small multiplicative subgroups of $\mathbb{F}_p$’, *Israel J. Math.* **172** (2009), 61–74.

[5] J. Bourgain, K. Ford, S. V. Konyagin and I. E. Shparlinski, ‘On the divisibility of Fermat quotients’, *Michigan Math. J.*, **59** (2010), 313–328.

[6] J. Bourgain, S. V. Konyagin and I. E. Shparlinski, ‘Product sets of rationals, multiplicative translates of subgroups in residue rings and fixed points of the discrete logarithm’, *Intern. Math. Research Notices*, **2008** (2008), Article ID rnn090, 1–29.

[7] J. Bourgain, S. V. Konyagin and I. E. Shparlinski, ‘Distribution of elements of cosets of small subgroups and applications’, *Intern. Math. Research Notices*, (to appear).

[8] J. Bourgain, S. Konyagin, C. Pomerance and I. E. Shparlinski, ‘On the smallest pseudopower’, *Acta Arith.*, **140** (2009), 43–55.

[9] W. van Dam, ‘Quantum algorithms for weighing matrices and quadratic residues’, *Algorithmica*, **34** (2002), 413–428.

[10] W. van Dam, S. Hallgren and L. Ip, ‘Quantum algorithms for some hidden shift problems’, *SIAM J. Comp.*, **6** (2006), 763–778.

[11] I. B. Damgrard, ‘On the randomness of Legendre and Jacobi sequences’, *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **403** (1990), 163–172.

[12] J. von zur Gathen and J. Gerhard, *Modern computer algebra*, Cambridge University Press, Cambridge, 2003.

[13] J. Hoffstein and D. Lieman, ‘The distribution of the quadratic symbol in function fields and a faster mathematical stream cipher’, *Proc. Workshop on Cryptography and Computational Number Theory, Singapore 1999*, Birkhäuser, 2001, 59–68.

[14] N. Koblitz and A. Menezes, ‘Pairing-based cryptography at high security levels’, *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **3796** (2005), 13–36.

[15] S. V. Konyagin and I. E. Shparlinski, *Character sums with exponential functions and their applications*, Cambridge Univ. Press, Cambridge, 1999.

[16] A. Ostafe and I. E. Shparlinski, ‘Pseudorandomness and dynamics of Fermat quotients’, *SIAM J. Discr. Math.*, **25** (2011), 50–71.

[17] A. C. Russell and I. E. Shparlinski, ‘Classical and quantum algorithms for function reconstruction via character evaluation’, *J. Compl.*, **20** (2004), 404–422.

[18] I. E. Shparlinski, ‘On the value set of Fermat quotients’, *Proc. Amer. Math. Soc.*, **140** (2012), 1199–1206.

[19] I. E. Shparlinski, ‘On vanishing Fermat quotients and a bound of the Ihara sum’, *Preprint*, 2011.

[20] F. Vercauteren, ‘Hidden root problem’, *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **5209** (2008), 89–99.

J. Bourgain Institute for Advanced Study, Princeton
bourgain@ias.edu

M. Z. Garaev Universidad Nacional Autónoma de México
garaev@matmor.unam.mx

S. V. Konyagin Steklov Mathematical Institute
konyagin@mi.ras.ru

I. Shparlinski Macquarie University
igor.shparlinski@mq.edu.au

Towards full collusion resistant. ID-based establishment of pairwise keys

Óscar García-Morchon and Ludo Tolhuizen

This talk is based on a joint work with Domingo Gómez and Jaime Gutierrez.

Abstract

Usually a communication link is secured by means of a symmetric-key algorithm. For that, a method is required to securely establish a symmetric-key for that algorithm. This old problem is still relevant and of paramount importance both in existing computer networks and new large-scale ubiquitous systems comprising resource-constrained devices. Identity-based pairwise key agreement allows for the generation of a common key between two parties given secret keying material owned by the first party and the identity of the second one. However, existing methods are prone to collusion attacks.

In this paper we discuss a new class of key establishment scheme aiming at full collusion resistant identity-based symmetric-key agreement and propose a specific scheme, the HIMMO algorithm, relying on two design concepts: Hiding Information and Mixing Modular Operations. Collusion attacks on schemes from literature cannot readily be applied to HIMMO. Also, the simple logic of the HIMMO algorithm allows for very efficient implementations in terms of both speed and memory. Finally, being an identity-based symmetric-key establishment scheme, HIMMO allows for efficient real-world key exchange protocols.

Introduction

This paper deals with the classical problem of key establishment. As in previous works [4],[2],[7], we focus on an *identity-based* (ID-based) scheme for symmetric-key agreement between pairs of devices in a network. That is, each node in the network has an identifier, and a trusted third party (TTP) provides it with secret keying material - linked to the device identifier - in a secure way. A node that wishes to communicate with another node uses its own secret keying material and the identity of the other node to generate a common pairwise key.

Existing ID-based symmetric-key agreement schemes are prone to *collusion attacks*: secret keying material of various nodes can be combined in order to obtain information on the secret key generated by a pair of (other) nodes. This combining can be performed by colluding legitimate owner(s) of the nodes, or by an attacker who has compromised some nodes and obtained their secret keying material. Existing schemes [4],[2],[7] allow for efficient collusion attacks (see Section). These efficient collusion attacks imply that it is infeasible to prevent successful attacks by relatively few colluding devices unless much secret keying material is stored in each node, which may be problematic in real-world applications since it increases CPU and storage needs.

This paper discusses a new class of ID-based key establishment schemes allowing for efficient operation – with respect to the amount of stored keying material and key computation time, which is especially relevant for resource-constrained devices – while it is based on mathematical problems for which the collusion attacks on the schemes from literature cannot readily be applied. We hope that our scheme, the HIMMO algorithm, and its underlying design principles can be a step towards full collusion resistant identity-based establishment of symmetric-keys.

Definition 1 (Full collusion resistant)**.** *An identity-based symmetric-key establishment scheme is* full collusion resistant *if for any set of colluding nodes no bit of a key shared by non-colluding nodes can be guessed with a probability higher than* $1/2$ *in polynomial time.*

The rest of this paper is organized as follows. In Section we give an overview of related work. In Section describes our HIMMO algorithm. In Section we discuss the design principles and underlying mathematical problems. Finally, we present our conclusions in Section .

Previous identity-based symmetric-key distribution schemes

Matsumoto and Imai [4] give a nice description of the key distribution problem, and provide a solution that serves as a base for many other schemes from literature. They propose that a trusted third party (TTP) chooses a secret function $f(x,y)$ that is *symmetric*, that is, $f(x,y) = f(y,x)$. The variables x and y are taken from a set of node identifiers $\mathcal{I}$, and the output from f is the key. The secret key material for the node with identifier η is a function $KM_\eta(y)$ which is such that $KM_\eta(\eta') = f(\eta,\eta')$ for all η'. As f is symmetric, it is guaranteed that the keys generated by two nodes for communicating with each other are equal.[1]

In [2], Blundo *et al.* choose the secret function $f(x,y)$ to be a symmetric polynomial over a finite field of degree α in each variable; the identifiers are considered as field elements as well. Blundo *et al.* show that their scheme offers information-theoretic security as long as an attacker knows the secret keying material of α or less nodes. However, $\alpha+1$ colluding nodes can obtain the root keying material by simple Lagrange interpolation.

In order to avoid the simple interpolation attack, Zhang et al. [7] proposed a ”noisy” version of the scheme of Blundo et al. [2]. Their basic idea is to provide node η with a polynomial $KM_\eta(x)$ that is ”close” to, but not exactly the same as $f(x,\eta)$. Nodes η and η' can compute $KM_\eta(\eta')$ and $KM_{\eta'}(\eta)$ as before; these values are no longer equal, but because they are close they can be used to generate a shared key. We now describe the main steps:

- *The TTP chooses a random symmetric, bivariate polynomial $f(x,y) \in \mathbb{Z}_p[x,y]$ of degree α in each variable and a noise bound r with $r < p$. It also chooses at random univariate ”noise” polynomials $g(y)$ and $h(y)$ of degree α over $\mathbb{Z}_p$. Next, it determines*

$$\mathcal{N} := \{\eta \in \mathbb{Z}_p : g(\eta), h(\eta) \in [0,r]\}$$

  Each node each given an identifier from $\mathcal{N}$. For each node $\eta \in \mathcal{N}$, the TTP chooses a random bit b_η and provides node η the univariate polynomial:

$$KM_\eta(x) = f(x,\eta) + b_\eta g(x) + (1-b_\eta)h(x).$$

- *A node η wishing to communicate with node η' computes $KM_\eta(\eta')$ and takes its $\ell - r$ most significant bits as key (where ℓ is such that $2^{\ell-1} < p \leq 2^\ell$). It sends $h(KM_\eta(\eta'))$ to node η', where h is an hash-function. Node η' computes three numbers, namely $KM_{\eta'}(\eta), KM_{\eta'}(\eta) + 2^r$ and $KM_{\eta'}(\eta) - 2^r$, and takes as key the $\ell - r$ most significant bits of the number for which the hash-value agrees with the received hash-value $h(KM_\eta(\eta'))$.*

Albrecht et al. [1] designed an efficient collusion attack on the scheme of Zhang *et al.* based on error-correcting techniques, that works if the $4\alpha+1$ nodes collude. They also provide an attack that works with 3α colluding nodes, but has time complexity $\mathcal{O}(r)$. Then, they suggested a generalized scheme based on adding more noise:

- *The TTP also chooses a natural number u such that $4ur < p$ and, for each node $\eta \in \mathcal{N}$, integers a_η, b_η and c_η such that $a_\eta, b_\eta \in [-u,u]$ and $c_\eta \in [-ur,ur]$, and gives node η the univariate polynomial:*

$$KM_\eta(x) = f(x,\eta) + a_\eta g(x) + b_\eta h(x) + c_\eta.$$

[1] Matsumoto and Imai in fact consider the more general situation that any group of t nodes must generate a common key; we restrict ourselves to the case $t = 2$.

They also provided an attack on this new cryptography protocol of time complexity $O(\alpha^3 + 8\alpha u^3)$, and requiring only $\alpha + 3$ compromised nodes. Their attack consists of two steps. In the first step, by means of linear algebra methods, they recover the linear vector space generated by the univariate polynomials $g(x)$ and $h(x)$. In the second step, they use lattice reduction techniques to recover f, knowing the polynomials g and h.

The HIMMO Algorithm

In this section, we describe our HIMMO algorithm for ID-based symmetric-key establishment. It relies on two new design principles:

1. **Hiding of information** by adding noise that is completely independent and random, for each node. This is similar to what is done by Zhang *et al.* [7], but they have only two possible noise contributions (the noise polynomials g and h, see previous section).

2. **Mixing of modular operations** by using m symmetric bivariate polynomials with coefficients in the integers modulo p_i for generating the secret keying material.

A key difference with all previous schemes [2], [7], [1] is that the modules $p_1, \ldots, p_m$ are kept secret and are only known to the TTP, *not* to the nodes. The nodes do know, however, that each module differs a multiple of 2^b from a known constant N.

In our description, we use the following notation. For each real x, we denote by $\lfloor x \rfloor$ the value of x rounded downwards to the closest integer, that is,

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}.$$

For integer a and integer $p \geq 2$, we denote by $\langle a \rangle_p$ the remainder of dividing a by p. Stated differently,

$$0 \leq \langle a \rangle_p \leq p - 1 \text{ and } a \equiv \langle a \rangle_p \bmod p.$$

Description

The operation of our ID-based symmetric-key establishment scheme comprises three phases:

1. System initialization

The TTP selects a private positive integer m, and three public positive integers b,N and α satisfying:

$$2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}.$$

The TTP also generates the following private material:

- m distinct positive integers $p_1, \ldots, p_m$ of the form $p_i = N - 2^b \beta_i$ where $1 \leq \beta_i \leq 2^b - 1$, for $i = 1, \ldots, m$;
- m symmetric bi-variate polynomials $f_1(x,y), \ldots, f_m(x,y)$, all of degree at most α in each variable, such that for $i = 1, \ldots, m$, the polynomial $f_i(x,y)$ has its coefficients in the set $\{0, 1, \ldots, p_i - 1\}$.

For $1 \leq i \leq m$, we write

$$f_i(x,y) = \sum_{j=0}^{\alpha} f_{i,j}(y) x^j \text{ with } f_{i,j}(y) \in \mathbb{Z}_{p_i}[y].$$

2. Node registration: distribution of secret keying material

For each node $\eta \in \{1,\ldots,2^b-1\}$, that wants to register, the TTP selects $\alpha+1$ integers $\varepsilon_{\eta,j}$ (the noise) satisfying the following equation:

$$|\varepsilon_{\eta,j}| < 2^{(\alpha+1-j)b-2}, j = 0,\ldots,\alpha. \tag{1}$$

The TTP provides node η with the secret keying material coefficients $KM_{\eta,0}, KM_{\eta,1},\ldots,KM_{\eta,\alpha}$, defined as

$$KM_{\eta,j} = \langle \sum_{i=1}^{m} \langle f_{i,j}(\eta)\rangle_{p_i} + 2^b\varepsilon_{\eta,j}\rangle_N. \tag{2}$$

3. Operational phase: key agreement

Node η generates its key with η' as:

$$K_{\eta,\eta'} = \langle\langle \sum_{j=0}^{\alpha} KM_{\eta,j}\eta'^j\rangle_N\rangle_{2^b}. \tag{3}$$

With explicit examples, it can be shown that $K_{\eta,\eta'}$ and $K_{\eta',\eta}$ are not necessarily equal. It can be shown, however, that the keys are approximately equal, as described in the following theorem.

Theorem 2. *Let* $0 \le \eta,\eta' \le 2^b-1$. *Then we have that*

$$K_{\eta,\eta'} \in \{\langle K_{\eta',\eta} + jN\rangle_{2^b} \mid -\Delta \le j \le \Delta\}, \text{ where } \Delta = 3m+\alpha+1.$$

In order that devices η and η' agree on a common key, an additional step is performed. In this step, device η to device η' the value $h(K_{\eta,\eta'})$, where the function h is such that $h(i) \neq h(K_{\eta,\eta'})$ for each potential key i (as indicated in Theorem 2) different from $K_{\eta,\eta'}$. In this way, η' finds the key $K_{\eta,\eta'}$ that is subsequently used to secure communications. An example of such a function h is a hash function like in [7].

Design principles of the HIMMO algorithm and discussion

As stated before, our HIMMO algorithm relies on two principles, namely (i) hiding of information and (ii) mixing of modular operations. Both principles further exhibit the feature that only partial knowledge on the used modules is available. This is described below.

Hiding of information ($m \ge 1$)

In Equation 2, we see that for each key material coefficient $KM_{\eta,j}$, parts of the sum of the polynomial evaluations are hidden by the noisy term $2^b\varepsilon_{\eta,j}$. This design concept is related to the so called Extended Hidden Number Problem (EHNP) [5], which can be stated as follows:

Problem 3 (EHNP). *Let p be a prime and b a positive integer* $2^b < p$. *Suppose for many random values* $\eta \in \{0,1,\ldots,p-1\}$, *the value* $\langle\langle f(\eta)\rangle_p\rangle_{2^b}$ *is given, where and* $f(x) \in F_p[x]$ *is an unknown polynomial of known degree* α. *Recover* $f(x)$ *in polynomial time*

Among other applications, Boneh and Venkatesan in [3] found nice links between the EHNP for $\alpha = 1$ and the security of the Diffie-Hellman Key Exchange protocol. Others interesting generalizations can be consulted in [5]. When $p_1 = p$ is a prime number, attacks are known, e.g. [6] that work if the number of colluding nodes is sufficiently large.

The main security issue with this design principle is that the usage of a single polynomial does not remove the underlying ring structure because the generated key is approximately equal[2] to the one generated from the original polynomial:

[2] Equation 3 uses module N, where $\beta_1 << N$ is missing, while here all reductions are module p_1.

$$K_{\eta,\eta'} \approx \langle\langle f_1(\eta,\eta')\rangle_{p_1}\rangle_{2^b} = \langle\langle f_1(\eta',\eta)\rangle_{p_1}\rangle_{2^b} \approx K_{\eta',\eta}$$

However, existing attacks cannot directly be applied to our scheme with $m = 1$ if p_1 is secret, as we assumed above. Also, if p_1 would be known, possibly an attack could be derived that requires less colluding nodes than current attacks, using that the identifiers for our scheme are in the relatively small set $\{1,\dots,2^b-1\}$, while current attacks assume that the identifiers are uniformly distributed on $\{0,1,\dots,p-1\}$.

Mixing of modular operations ($m \geq 2$)

In Equation 2, we see (for $m \geq 2$) a mixing of modular operations in the sum $\sum_{i=1}^{m}\langle f_{i,j}(\eta)\rangle_{p_i}$.

Problem 4 (Mixing of modular operations)**.** *Let $p_1,\dots,p_m$ be m distinct positive integer numbers such that $p_i = N - \beta_i 2^b$, where $2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}$ and $0 \leq \beta_i < 2^b$. Moreover, for $i = 1,\dots,m$, be let $f_i(x) \in \mathbb{Z}_{p_i}[x]$ have degree at most α. For η in $S = \{1,..,2^b-1\}$, we define $H(\eta) := \langle\sum_{i=1}^{m}\langle f_i(\eta)\rangle_{p_i}\rangle_N$. Given a number N_S of pairs $(\nu, H(\nu))$, the problem consists in guessing any bit of $H(\eta)$ associated to a known input value η with a probability higher than 1/2.*

Remark Problem 4 is further enhanced by the fact that the attacker does not know the modules $p_1,\dots,p_m$; all he knows is that each p_i differs the b bit unknown integer β_i multiple of 2^b from N.

In order to explain the idea behind this second design principle, we consider a simple special case, *viz.* that for $1 \leq i \leq m$, we have that $f_i(x,y) = A_i x^\alpha y^\alpha$ for some $A_i \in \{1,\dots,p_i-1\}$. Moreover, we take $N = 2^{b(\alpha+2)}-1$ and $\varepsilon_{\eta,\alpha} = 0$. We write:

$$A_i\eta^i = R^{(2)}_{i,\eta}2^{b(\alpha+2)} + R^{(1)}_{i,\eta}2^b + R^{(0)}_{i,\eta},$$

$$\text{with } 0 \leq R^{(0)}_{i,\eta} \leq 2^b-1 \text{ and } 0 \leq R^{(1)}_{i,\eta} \leq 2^{b(\alpha+1)}-1.$$

As $p_i = 2^{b(\alpha+2)} - \beta_i 2^b - 1$, the single non-zero coefficient $KM_{\eta,\alpha}$ of node η is given by

$$\left\langle \sum_{i=1}^{m} \left\langle f_{i,j}(\eta)\right\rangle_{p_i} \right\rangle_N = \left\langle \sum_{i=1}^{m} \left\langle A_i\eta^i\right\rangle_{p_i} \right\rangle_N = \left\langle \sum_{i=1}^{m} \left\langle \left(R^{(1)}_{i,\eta} + \beta_i R^{(2)}_{i,\eta}\right)2^b \; + \; \left(R^{(0)}_{i,\eta} + R^{(2)}_{i,\eta}\right) \right\rangle_{p_i} \right\rangle_N =$$

$$\left\langle \sum_{i=1}^{m} \left\langle \left(R^{(1)}_{i,\eta} + \beta_i R^{(2)}_{i,\eta} + \left\lfloor \frac{R^{(0)}_{i,\eta} + R^{(2)}_{i,\eta}}{2^b} \right\rfloor \right) 2^b + \langle R^{(0)}_{i,\eta} + R^{(2)}_{i,\eta}\rangle_{2^b} \right\rangle_{p_i} \right\rangle_N \approx$$ [3]

$$\left\langle \sum_{i=1}^{m} \left(R^{(1)}_{i,\eta} + \beta_i R^{(2)}_{i,\eta} + \left\lfloor \frac{R^{(0)}_{i,\eta} + R^{(2)}_{i,\eta}}{2^b} \right\rfloor \right) 2^b + \langle R^{(0)}_{i,\eta} + R^{(2)}_{i,\eta}\rangle_{2^b} \right\rangle_N \tag{4}$$

In this example, we observe that the modulo computations affect the $b(\alpha+1)$ most significant bits of the keying material in a way that is dependent on β_i. By adding over i, these β_i-dependencies are mixed. We also see mixing in the b least significant bits of the keying material, as they depend on the sum of the most and least significant bits of $A_i\eta^i$ The nice aspect of the design is that the components originating from different polynomials $f_i(x,y)$ hide each other so that an attacker can only observe the sum modulo N, learning nothing about the individual components.

Thus, our HIMMO algorithm applies the second design concept by using p_i with such a form that they introduce non-linear operations when the TTP generates the secret keying material for node η from the secret bivariate polynomials. However, the public modulus N and the p_i share a given structure that still allows for the generation of a b bit key by means of Equation 3. Thus, the

[3] The effect of the reduction module p_i due to carry propagation is limited due to the form of p_i.

smart part of the cryptoblock happens in the step in which the TTP generates the keying material shares from the secret root keying material creating a non-linear keying material structure in the most significant bits of the secret keying material coefficients as shown in the specific example in Equation 4. Later, during key establishment only the common terms of p_i and N are used so that a common key can be generated mod N, i.e., without requiring knowledge of the secret terms β_i. Thus, the resulting b-bit key combines the contributions from all polynomials over different rings:

$$K_{\eta,\eta'} \approx \langle \sum_{i=1}^{m} \langle f_i(\eta,\eta') \rangle_{p_i} \rangle_{2^b} = \langle \sum_{i=1}^{m} \langle f_i(\eta',\eta) \rangle_{p_i} \rangle_{2^b} \approx K_{\eta',\eta}$$

Conclusions

Our HIMMO algorithm addresses the old key establishment problem in a different way bringing many advantages. Operationally, it allows for direct ID-based pairwise key establishment simplifying protocol operation. Computationally, the design concepts relying on polynomials allow for very fast operation with minimal memory needs. From a security point of view, although the design concepts seem to be sound, further analysis is required because they are also fairly new. In particular, the first design concept presents some links to the EHNP, and thus, it might make possible partial security analysis of our scheme. To the best of our knowledge, our second design concept, mixing of the evaluation of polynomials using different modules, has not been explored in literature so far. The task of an attacker with regard to both design concepts is further complicated by the fact that he only has partial knowledge on which modules have been used.

References

[1] M. Albrecht, C. Gentry, S. Halevi, and J. Katz. “Attacking cryptographic schemes based on perturbation polynomials”. In E. Al-Shaer, S. Jha, and A.D. Keromytis, editors, ACM Conference on Computer and Communications Security, pages 1-10. ACM, 2009.

[2] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. “Perfectly-secure key distribution for dynamic conferences” . In E.F. Brickell, editor, CRYPTO ’92, volume 740 of Lecture Notes in Computer Science, pp. 471-486. Springer, 1992.

[3] D. Boneh and R. Venkatesan. “Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes”. In N. Koblitz, editor, CRYPTO, volume 1109 of Lecture Notes in Computer Science, pp. 129-142. Springer, 1996.

[4] T. Matsumoto and H. Imai, “On the key predistribution system: a practical solution to the key distribution system”, in C. Pomerance (Ed): Advances in Cryptology, CRYPTO’87, volume 293 of Lecture Notes in Computer Science, pp. 185-193, Springer, 1988.

[5] I. E. Shparlinski, “Sparse polynomial approximation in finite fields” , Proc. 33rd ACM Symp. on Theory of Comput., Crete, Greece, July 6-8, 2001, 209-215.

[6] I.E. Shparlinski and A. Winterhof. “Noisy interpolation of sparse polynomials in finite fields”. Appl. Algebra Eng. Commun. Comput., 16(5):307 317, 2005.

[7] W. Zhang, M. Tran, S. Zhu and G. Cao, “A random perturbation-based scheme for pairwise key establishment in sensor networks”, 8th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), 2007.

Ó. García-Morchon Philips Group Innovation
oscar.garci@philips.com
L. Tolhuizen Philips Group Innovation
ludo.tolhuizen@philips.com

Carlitz rank of permutations of finite fields

Alev Topuzoğlu

Permutation polynomials over finite fields have attracted a lot of attention in the last decades, due to their vast applications, especially in pseudorandom number generation, combinatorics, coding and symmetric crytography. In order to meet the specific requirements of individual applications, methods of construction of various types of permutations and/or new ways of classifying them are needed.

The aim of this talk is to present a new classification of permutation polynomials (see [1, 2]), report on recent developments and describe some of its interesting applications.

By a classical result of Carlitz, the group of permutation polynomials of the finite field Fq under the operation of composition and reduction modulo $x^q - x$, is generated by the monomial x^{q-2} , and the linear polynomials. Consequently, as pointed out in [2], with $\mathcal{P}_0(x) = a_0 x + a_1$, any permutation $\rho(x)$ of a finite field $\mathbb{F}_q$ can be represented by a polynomial

$$\mathcal{P}_n(x) = (\ldots((a_0 x + a_1)^{q-2} + a_2)^{q-2} \cdots + a_n)^{q-2} + a_{n+1}, n \geq 0,$$

where $a_1, a_{n+1} \in \mathbb{F}_q, a_1 \in \mathbb{F}_q^* = \mathbb{F}_q \backslash \{0\}$ for $i = 0, 2, \ldots, n$.

The Carlitz rank of a permutation polynomial can naturally be consid- ered as a complexity measure. Relations between this concept and properties like the degree, weight and cycle structure of permutation polynomials will be discussed. The question of evaluating the Carlitz rank of a given permutation $\rho(x)$ will be addressed, and results on the enumeration of permutations of a fixed Carlitz rank will be presented. Applications, for instance, construc- tion of ”random” permutations with a particular cycle structure, or APN permutations, which are almost Costas will be described.

Finally the notion of Carlitz rank will be extended to the case of multi- variate polynomial systems of $\mathbb{F}_{q^m},\ m > 1$.

References

[1] E. Aksoy, A. Çeşmelioğlu, W. Meidl, and A. Topuzoğlu. On the Carlitz rank of permutation polynomials. *Finite Fields Appl.*, 15(4):428–440, 2009.

[2] A. Çeşmelioğlu, W. Meidl, and A. Topuzoğlu. On the cycle structure of permutation polynomials. *Finite Fields Appl.*, 14(3):593–614, 2008.

A. Topuzoğlu Sabanci University
alev@sabanciuniv.edu

Optimal reductions of some decisional problems to the rank problem

Jorge L. Villar

Abstract

In the last years the use of large matrices and their algebraic properties proved to be useful to instantiate new cryptographic primitives like Lossy Trapdoor Functions and encryption schemes with improved security, like Key Dependent Message resilience. In these constructions the rank of a matrix is assumed to be hard to guess when the matrix is hidden by elementwise exponentiation. This problem, that we call here the Rank Problem, is known to be related to the Decisional Diffie-Hellman problem, but in the known reductions between both problems there appears a loss-factor in the advantage which is linear in the rank of the matrix.

In this work, we give a new and better reduction between the Rank problem and the Decisional Diffie-Hellman problem, such that the reduction loss-factor depends logarithmically in the rank. This new reduction can be applied to a number of cryptographic constructions improving their efficiency. The main idea in the reduction is to build a matrix from a DDH tuple which rank shifts from r to $2r$ and then apply a hybrid argument to find a reduction in the general case.

On the other hand, the new reduction is optimal as we show the nonexistence of more efficient ones in a wide class of reductions containing all the “natural” ones (i.e., black-box and algebraic). The result is twofold: there is no (natural) way to build a matrix which rank shifts from r to $2r+\alpha$ for $\alpha > 0$, and no hybrid argument can improve the logarithmic loss-factor obtained in the above reduction.

The techniques used in this work extend naturally to other “algebraic” problems like DLinear or Decisional 3-Party Diffie-Hellman problems, also obtaining reductions of logarithmic complexity.

Motivation

In the last years the use of large matrices and their algebraic properties proved to be useful to instantiate new cryptographic primitives like Lossy Trapdoor Functions [5, 3] and encryption schemes with improved security, like Key Dependent Message [1]. In these constructions the rank of a matrix is assumed to be hard to guess when the matrix is hidden by elementwise exponentiation. This problem, that we call here the Rank Problem, is known to be related to the Decisional Diffie-Hellman (DDH) problem, but in the known reductions between both problems there appears a loss-factor in the adversaries’ advantage which is linear in the rank of the matrix. The Rank Problem first appeared in some papers under the names Matrix-DDH [1] and Matrix d-Linear [4] problems.

In the cryptographic constructions mentioned above, some secret values (messages of keys) are encoded as group element vectors and then hidden by multiplying them by an invertible matrix. The secret value is recovered by inverting the operations: first multiplying by the inverse matrix and then inverting the encoding as group elements. This last step requires to encode a few bits (typically, a single bit) in each group element, forcing the length of the vector and the rank of the matrix to be comparable to the binary length of the secret value. Security of these schemes is related to the indistinguishability of full-rank matrices and low-rank (e.g., rank 1) matrices: If the invertible matrix is replaced by a low rank one, the secret value is information-theoretically hidden. Therefore, the security of these schemes is related to the hardness of the Rank problem for matrices of large rank (e.g., 320 or 1024).

Reductions of the DDH problem to the Rank problem are based in the obvious relationship between them in the case of 2×2 matrices. Namely, from a DDH problem tuple (g, g^x, g^y, g^z)

one can build a matrix $g^M = \begin{pmatrix} g & g^x \\ g^y & g^z \end{pmatrix}$, which is the elementwise exponentiation of the $\mathbb{Z}_q$ matrix $M = \begin{pmatrix} 1 & x \\ y & z \end{pmatrix}$. Therefore, for a 0-instance (i.e., $z = xy$), $\det M = 0$, while for a 1-instance (i.e., $z \neq xy$), $\det M \neq 0$, that is the rank of M shifts from 1 to 2 depending on the DDH instance. This technique can be applied to larger (even non-square) matrices by just padding the previous 2×2 block with some ones in the diagonal and zeroes elsewhere, just increasing the rank from 1 or 2 to $r+1$ or $r+2$, where r is the number of ones added to the diagonal.

Now a general reduction to any instance of the rank problem (i.e., telling apart hidden matrices of ranks r_1 and r_2) to DDH is obtained by applying a hybrid argument, incurring into a loss-factor in the adversaries' advantage which grows linearly in the rank difference $r_2 - r_1$.

This loss-factor has an extra impact on the efficiency of the cryptographic schemes based on matrices, as for the same security level the size of the group has to be increased, and therefore the size of public keys, ciphertexts, etc. is increased accordingly.

Until now it was an open problem to find a tighter reduction of DDH to the Rank problem. To face this kind of problems one can choose between building new tighter reductions or showing impossibility results. However, most of the known impossibility results are quite limited because they only state the nonexistence of reductions of certain type (e.g., black-box, algebraic, etc.). But still this negative results have some value since they capture all possible 'natural' reductions between computational problems at least in the generic case (e.g., without using specific properties of certain groups).

Main Results

In this work, we give a new and better reduction between the Rank and the DDH problems, such that the reduction loss-factor depends logarithmically in the rank of the matrices. This new reduction can be applied to a number of cryptographic constructions improving their efficiency. The main idea in the reduction is to build a matrix from a DDH tuple which rank shifts from r to $2r$ and then apply a hybrid argument to find a reduction in the general case.

On the other hand, the new reduction is optimal as we show the nonexistence of more efficient ones in a wide class of reductions containing all the "natural" ones (i.e., black-box and algebraic). The result is twofold: there is no (natural) way to build a matrix which rank shifts from r to $2r + \alpha$ for $\alpha > 0$, and no hybrid argument can improve the logarithmic loss-factor obtained in the above reduction.

Basically, the new reduction achieves the following result.

(Informal) Theorem 1. *For any ℓ_1, ℓ_2, r_1, r_2 such that $1 \leq r_1 < r_2 \leq \min(\ell_1, \ell_2)$ there is a reduction of the DDH problem to the Rank problem for $\ell_1 \times \ell_2$ matrices of rank either r_1 or r_2, where the advantage of the problem solvers fulfil*

$$\mathbf{AdvRank}(\mathcal{G}, \ell_1, \ell_2, r_1, r_2; t) \leq \lceil \log_2 r_2 - \log_2 r_1 \rceil \, \mathbf{AdvDDH}(\mathcal{G}; t')$$

and their running times t and t' are essentially equal.

In particular, our reduction relates the hardness to tell apart $\ell \times \ell$ full rank matrices from rank 1 matrices with a loss-factor of only $\log_2(\ell)$, instead of the factor ℓ obtained in previous reductions.

At this point, it arises the natural question of whether a tight reduction exists. Unfortunately we also show optimality of the new reduction via the following negative result.

(Informal) Theorem 2. *For any ℓ_1, ℓ_2, r_1, r_2 such that $1 \leq r_1 < r_2 \leq \min(\ell_1, \ell_2)$ and any 'natural' reduction $\mathcal{R}$ of DDH to the Rank problem, the advantages of the Rank problem solver $\mathcal{A}$ and the DDH solver $\mathcal{R}([\mathcal{A}])$ fulfil*

$$\mathbf{AdvRank}_{\mathcal{R}[\mathcal{A}]}(\mathcal{G}, \ell_1, \ell_2, r_1, r_2; t) \geq \lceil \log_2 r_2 - \log_2 r_1 \rceil \, \mathbf{AdvDDH}_{\mathcal{A}}(\mathcal{G}; t') - \varepsilon$$

where the running times t, t' are similar and ε is a negligible quantity.

Here ‘natural’ reduction basically means a black-box reduction which transforms a DDH tuple into a hidden matrix by performing only (probabilistic) algebraic manipulations, which are essentially linear combinations of the exponents with known integer coefficients, depending on the random coins of the reduction.

All generic reductions from computational problems based on cyclic groups fall into this category. Therefore, this result has to be interpreted as one cannot expect finding a tighter reduction for a large class of groups unless a new (non-black-box or not algebraic) technique is used. Nevertheless, falsifying this negative result would imply both an improvement on both the efficiency of the cryptosystems based on matrices and the discovery of a new reduction approach.

The techniques used in this work extend naturally to other “algebraic” problems like DLinear or Decisional 3-Party Diffie-Hellman problems, also obtaining reductions with logarithmic complexity. Actually, these reductions recently appeared in [2].

Further Research

Some of the ideas and techniques used in this work suggest that the problem of the optimality of certain type of reductions for a class of decisional assumptions can be studied under the Algebraic Geometric point of view. In particular, this could help to close the gap in the loss-factor between the reduction and the lower bound when reducing DLinear or D3DH to Rank, and could made possible to obtain similar results for a broad class of computational problems. A second open problem is how the techniques and results adapt to the case of composite order groups, specially when the factorization of the order or the order itself is unknown.

References

[1] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008.

[2] David Galindo, Javier Herranz, and Jorge Luis Villar. Identity-based encryption with master key-dependent message security and applications. *IACR Cryptology ePrint Archive*, 2012:142, 2012.

[3] Dennis Hofheinz. All-but-many lossy trapdoor functions. Cryptology ePrint Archive, Report 2011/230, 2011. `http://eprint.iacr.org/`.

[4] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2009.

[5] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *IACR Cryptology ePrint Archive*, 2007:279, 2007.

J. L. Villar Universitat Politècnica de Catalunya
jvillar@ma4.upc.edu

Generalizations of complete mappings of finite fields and some applications

Arne Winterhof

This talk is based on a joint work with Alina Ostafe.

Let $\mathbb{F}_q = \{\xi_1, \ldots, \xi_q\}$ be the finite field of q elements and $f(X) \in \mathbb{F}_q[X]$ a permutation polynomial over $\mathbb{F}_q$. For $k = 0, 1, \ldots$ we define the k-th iteration $f^{(k)}(X)$ of $f(X)$ by the recurrence relation

$$f^{(0)}(X) = X, \qquad f^{(k)}(X) = f^{(k-1)}(X), \qquad k = 1, 2, \ldots$$

For a finite set of s positive integers $\mathcal{K} = \{k_1, \ldots, k_s\}$ we call $f(X)$ a $\mathcal{K}$*-complete mapping* if

$$F_{\mathcal{K}}(X) = X + \sum_{k \in \mathcal{K}} f^{(k)}(X)$$

is also a permutation polynomial. For $\mathcal{K} = \{1\}$, that is, $f(X)$ and $X + f(X)$ are both permutation polynomials we get *complete mappings* as a first special case. A permutation polynomial f is called an *orthomorphism* if $-X + f(X)$ is also a permutation polynomial. Note that $f(X)$ is an orthomorphism whenever $-f(X)$ is a complete mapping and both terms coincide in characteristic 2. In analogy to $\mathcal{K}$-complete mappings we define a $\mathcal{K}$-orthomorphism as a permutation polynomial such that $-X + \sum_{k \in \mathcal{K}} f^{(k)}(X)$ is also a permutation.

In the first part of this talk we recall some known applications of these permutations to combinatorics, cryptography, numerics, and coding theory.

Complete mappings are pertinent to the construction of *orthogonal Latin squares* [2]. A $q \times q$ array (a_{ij}) is called a *Latin square* over $\mathbb{F}_q$ if each row and each column contains every element of $\mathbb{F}_q$ exactly ones. Two Latin squares (a_{ij}) and (b_{ij}) are said to be *orthogonal* if the q^2 ordered pairs (a_{ij}, b_{ij}) are all different. If $f(X)$ is a complete mapping, (a_{ij}) with $a_{ij} = \xi_i + \xi_j$ and (b_{ij}) with $b_{ij} = f(\xi_j) - \xi_i$ are orthogonal Latin squares.

$\mathcal{K}$-orthomorphisms can be used to define uniformly distributed sequences. Uniform distribution is a desirable feature of a sequence for both Monte Carlo-methods and cryptography and is very often estimated in terms of character sums. For an integer $K \geq 2$ let $f(X)$ be a $\{k\}$-orthomorphism for all $k = 1, \ldots, K-1$ and define a sequence over $\mathbb{F}_q$ by

$$u_{n+1} = f(u_n), \quad n \geq 0,$$

of least period $t \leq q$ with some initial value $u_0 \in \mathbb{F}_q$. Then for any nontrivial additive character ψ of $\mathbb{F}_q$ we have [1, Theorem 2]

$$\left| \sum_{n=0}^{N-1} \psi(u_n) \right| \ll K^{-1/2} t^{1/2} q^{1/2} \log t \quad \text{for } 1 \leq N \leq t.$$

Hence, the Erdős-Turán inequality (in the case that q is prime) implies a small discrepancy and thus a nice uniform distribution of the points $\{u_0/q, \ldots, u_{N-1}/q\}$ in the unit interval (if we identify $\mathbb{F}_q$ with the integers $\{0, 1, \ldots, q-1\}$) provided that K (and also N) is large with respect to t and q (and t and q are sufficiently large).

We can also use $\mathcal{K}$-complete mappings and $\mathcal{K}$-orthomorphisms to design check digit systems which detect the most common errors. A *check digit system* (defined with one permutation polynomial over $\mathbb{F}_q$) consists of a permutation polynomial $f(X) \in \mathbb{F}_q[X]$ and a control symbol $c \in \mathbb{F}_q$ such that each word $a_1,\ldots,a_{s-1}\in\mathbb{F}_q^{s-1}$ of length $s-1$ is extended by a check digit $a_s\in\mathbb{F}_q$ such that

$$\sum_{i=0}^{s-1} f^{(i)}(a_{i+1}) = c.$$

Since $f(X)$ is a permutation polynomial such a system detects all single errors $a\mapsto b$. Moreover it detects all

- neighbor transpositions $ab\mapsto ba$ if $f(X)$ is an orthomorphism;
- twin errors $aa\mapsto bb$ if $f(X)$ is a complete mapping;
- jump errors $abc\mapsto cba$ if $f(X)$ is a $\{2\}$-orthomorphism;
- jump twin errors $aca\mapsto bcb$ if $f(X)$ is a $\{2\}$-complete mapping.

In the second part of the talk we study the problem if certain classes of polynomials contain $\mathcal{K}$-complete mappings or $\mathcal{K}$-orthomorphisms for certain types of $\mathcal{K}$. These classes are

- polynomials of small degree;
- cyclotomic mapping polynomials;
- monomials;
- linearized polynomials.

In particular, several classes of complete mappings are listed in [3], an asymptotic formula for the number of cyclotomic mapping polynomials (of a fixed index) which are $\{k\}$-orthomorphisms and $\{k\}$-complete mappings for $k=1$ and 2 is given in [5], and the existence of cyclotomic mapping polynomials which are $\{k\}$-orthomorphisms for $k=1,\ldots,K-1$ with a K of order of magnitude $\log q$ is proved in [4].

In this talk we also present new results on $\{1,\ldots,k-1\}$-complete mappings, $k\geq 2$, which we call also *k-complete mappings* for simplicity. In particular, we search for polynomials which are k-complete mappings for $k=2,\ldots,K$ and call these mappings *K-strong complete*. Analogously we define *k-orthomorphisms* and *K-strong orthomorphisms*. Note that a polynomial which is K_1-strong complete and a K_2-strong orthomorphism can be used to design check digit systems which detect all errors of the form

$$\underbrace{a\ldots a}_{k}\mapsto\underbrace{b\ldots b}_{k},\quad k=1,\ldots,K_1-1,$$

and

$$a\underbrace{b\ldots b}_{k-1}\mapsto b\underbrace{a\ldots a}_{k-1},\quad k=1,\ldots,K_2-1,$$

respectively, but may also have other applications.

For example, take $f(X)=aX$ with an element $a\in\mathbb{F}_q^*$ of order s and $ind_a(2a-1)=t$, that is, $a^t=2a-1$ with $0\leq t<s$ or $t=\infty$ if such a t doesn't exist. If $a\neq 1$, we have

$$\sum_{j=0}^{k-1} f^{(j)}(X)=\sum_{j=0}^{k-1} a^jX=\frac{a^k-1}{a-1}X$$

and $f(X)$ is obviously $(s-1)$-strong complete but not s-strong complete. Moreover, since

$$-X+\sum_{j=1}^{k-1} f^{(j)}(X)=\frac{a^k-2a+1}{a-1}X,$$

$f(X)$ is a $(t-1)$-strong orthomorphism but if $t<s$, it is not a t-strong orthomorphism.

References

[1] Cohen, S. D.; Niederreiter, H.; Shparlinski, I. E.; Zieve, M. Incomplete character sums and a special class of permutations. 21st Journées Arithmétiques (Rome, 2001). J. Théor. Nombres Bordeaux 13 (2001), no. 1, 53–63.

[2] Mann, Henry B. The construction of orthogonal Latin squares. Ann. Math. Statistics 13, (1942). 418–423.

[3] Niederreiter, Harald; Robinson, Karl H. Complete mappings of finite fields. J. Austral. Math. Soc. Ser. A 33 (1982), no. 2, 197–212.

[4] Niederreiter, Harald; Winterhof, Arne Cyclotomic R-orthomorphisms of finite fields. Discrete Math. 295 (2005), no. 1–3, 161–171.

[5] Shaheen, Rasha; Winterhof, Arne Permutations of finite fields for check digit systems. Des. Codes Cryptogr. 57 (2010), no. 3, 361–371.

A. Winterhof Austrian Academy of Sciences
arne.winterhof@oeaw.ac.at
A. Ostafe Macquarie University Sydney
alina.ostafe@mq.edu.au

Contributed Talks

WMC

Contributed Talks

Unified addition formulæ for hyperelliptic curve cryptosystems

Oumar Diao and Marc Joye

Introduction

Hyperelliptic curve cryptography was introduced by Koblitz in 1989 [8] (see also [9]) as an alternative to elliptic curve cryptography. It bases its security on the discrete logarithm problem in the Jacobian of an hyperelliptic curve of genus $g \geq 2$ (HCDLP). Recent cryptanalytic results [7] have shown that hyperelliptic curve cryptosystems of genus $g \geq 3$ are prone to attacks better than generic methods for solving the HCDLP. As a consequence, although our techniques readily apply to any genus, the focus will be put on genus-2 hyperelliptic curves.

In practice, the hardness of the HCDLP is not sufficient (but necessary) to guarantee the security of the underlying cryptosystems; it only provides black-box security. An attacker may have more information than a mere access to the input and output of the algorithms. Specifically, the attacker may monitor the execution of the algorithm and get additional information through certain side channels such as the running time [10] or the power consumption [11]. Of particular importance is the resistance against simple side-channel analysis. Resistance against the more sophisticated differential side-channel analysis can be achieved using various randomization techniques [1]. This paper presents unified addition formulæ for hyperelliptic curve cryptosystems as an efficient means to thwart simple side-channel attacks, extending the techniques of [2, 3] to genus $g > 1$.

Background on Hyperelliptic Curves

A *hyperelliptic curve of genus g over a field* $\mathbb{K}$ is a non-singular curve given by an equation

$$C : y^2 + h(x)y = f(x)$$

where $f \in \mathbb{K}[x]$ is a monic polynomial of degree $2g+1$ and $h \in \mathbb{K}[x]$ is a polynomial of degree $\leq g$. The set of $\mathbb{K}$-rational points on C, denoted $C(\mathbb{K})$, is the set of all points $(x,y) \in \mathbb{K} \times \mathbb{K}$ satisfying the above equation together with the so-called 'point at infinity' ∞. The opposite of a finite point $P = (a,b)$ is the point $-P = (a, -b-h(a))$ and $-\infty = \infty$.

A *divisor on C* is a finite formal sum $D = \sum_{P \in C(\overline{\mathbb{K}})} n_P(P)$ with $n_P \in \mathbb{Z}$; its degree is defined as $\sum n_P$. A divisor D is said defined over $\mathbb{K}$ if $D = \sum n_P(P^\sigma)$ for every automorphism σ of $\overline{\mathbb{K}}$ over $\mathbb{K}$. The *function field of C over* $\mathbb{K}$, denoted $\mathbb{K}(C)$, is the field of fractions of the polynomial ring $\mathbb{K}[C] = \mathbb{K}[x,y]/(y^2 + h(x)y - f(x))$. Similarly, the function field $\overline{\mathbb{K}}(C)$ is defined as the field of fractions of $\overline{\mathbb{K}}[C]$. To any nonzero rational function $\psi \in \overline{\mathbb{K}}(C)$, one can associate a divisor via the valuation at all points as $\operatorname{div}(\psi) = \sum_{P \in C(\overline{K})} \nu_P(\psi)(P)$. Such a divisor is called a *principal divisor* and is of degree 0. The set of divisors defined over $\mathbb{K}$ forms an additive group denoted Div_C. The subgroup of degree-0 divisors is denoted Div_C^0 and its subgroup of principal divisors is denoted Princ_C. The *Jacobian of the curve C* is the quotient group $J_C = \mathrm{Div}_C^0 / \mathrm{Princ}_C$. Riemann-Roch theorem tells us that each element of J_C can be uniquely represented by a *reduced divisor*, that is, a divisor of the form

$$D = \sum_{i=1}^{m} (P_i) - m(\infty)$$

with (i) $P_i \neq \infty$, (ii) $P_i \neq -P_j$ if $i \neq j$, and (iii) $m \leq g$. A divisor satisfying Conditions (i) and (ii) (but not necessarily Condition (iii)) is said *semi-reduced*.

To avoid working in an extension of $\mathbb{K}$, a semi-reduced divisor $D = \sum_{i=1}^{m}(P_i) - m(\infty)$ is preferably identified using *Mumford representation* as a pair of polynomials $u(x)$ and $v(x)$ in $\mathbb{K}[x]$ where, letting $P_i = (x_i, y_i)$,

- $u := u(x) = \prod_{i=1}^{m}(x - x_i)$, and
- $v := v(x)$ is the unique polynomial of degree $< m$ such that $v(x_i) = y_i$ with appropriate multiplicity when P_i appears more than once in D.

We write $D = [u, v]$. Mumford representation leads to efficient algorithms for adding or doubling group elements in J_C [4].

Explicit formulæ for genus-2 hyperelliptic curves are detailed in [12]. The formulæ were subsequently improved by Costello and Lauter through a more direct geometric interpretation of the group law. Letting M, S and I the respective costs of a multiplication, squaring and inversion in $\mathbb{K}$, the best operation counts are $\underline{1\mathsf{I} + 17\mathsf{M} + 4\mathsf{S}}$ for the addition in J_C and $\underline{1\mathsf{I} + 19\mathsf{M} + 6\mathsf{S}}$ for the doubling in J_C [5].

Unified Addition Formulæ

Classically, computing in Jacobians of hyperelliptic curves is carried out as an application of Cantor's algorithm [4]. It takes on input two reduced divisors in Mumford representation and outputs a reduced divisor in Mumford representation. In more detail, given two reduced divisors $D_1 = [u_1, v_1]$ and $D_2 = [u_2, v_2]$, the algorithm first produces a semi-reduced divisor $[u, v]$ equivalent to $D_1 + D_2$ modulo Princ_C, such that

$$u = \frac{u_1 u_2}{d^2} \quad \text{and} \quad v \equiv \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f)}{d} \pmod{u} \tag{1}$$

with $d = \gcd(u_1, u_2, v_1 + v_2 + h) = s_1 u_1 + s_2 u_2 + s_3(v_1 + v_2 + h)$ for polynomials $s_1, s_2, s_3 \in \mathbb{K}[x]$ given by the extended Euclidean algorithm. This divisor is then reduced in a second step by repeatedly applying

$$u \leftarrow \mathrm{Monic}\left(\frac{v^2 + vh - f}{u}\right) \quad \text{and} \quad v \leftarrow -v - h \pmod{u} \; .$$

until $\deg(u) \leq g$.

For computational purposes, there are two main cases to consider:

1. Cantor general doubling: $D_1 = D_2$ and $\gcd(u_1, 2v_1 + h) = 1$;
2. Cantor general addition: $D_1 \neq D_2$ and $\gcd(u_1, u_2) = 1$.

Distinguishing these two cases allows one to derive explicit formulæ for low-genus curves. As shown in [4], the expression for v then verifies the simpler equation

$$v \equiv v_1 + s_3(f - v_1 h - {v_1}^2) \pmod{u} \quad \text{and} \quad v \equiv v_1 + s_1 u_1 (v_2 - v_1) \pmod{u} \tag{2}$$

for a Cantor general doubling and a Cantor general addition, respectively. The next proposition is our main ingredient. It states a relation that is satisfied for both cases. This will be useful in designing unified addition formulæ.

Proposition 1. *Using the previous notation, let $D_1 = [u_1, v_1]$ and $D_2 = [u_2, v_2]$ be two reduced divisors in a general Cantor operation. Then $[u, v] \sim D_1 + D_2$ where*

$$u = u_1 u_2 \quad \textit{and} \quad v(h + v_1 + v_2) \equiv f + v_1 v_2 \pmod{u} \; . \tag{3}$$

Proof. See e.g. [6, Proof of Theorem 10.3.14]. □

We now develop explicit addition formulæ in the Jacobian of genus-2 curves. We are concerned with the frequent case involving divisors of full degree. So, for $i \in \{1,2\}$, we let $u_i := u_i(x) = x^2 + u_{i,1}x + u_{i,0}$ and $v_i := v_i(x) = x^2 + v_{i,1}x + v_{i,0}$. We also let $v := v(x) = \ell_3 x^3 + \ell_2 x^2 + \ell_1 x + \ell_0$ for unknown coefficients ℓ_j, $0 \le j \le 3$. As in [5], we build a system of linear equations that solves to give these coefficients ℓ_j.

From Eq. (2), it clearly appears that in both cases (i.e., doubling and addition) $v \equiv v_1 \pmod{u_1}$ — remember that $u_1 \mid (f - v_1 h - {v_1}^2)$. This can be rewritten as

$$\ell_3 x^3 + \ell_2 x^2 + \ell_1 x + \ell_0 - (v_{1,1}x + v_{1,0}) \equiv 0 \pmod{(x^2 + u_{1,1}x + u_{1,0})},$$

which gives rise to two linear equations:

$$\begin{cases} (u_{1,1}^2 - u_{1,0})\ell_3 - u_{1,1}\ell_2 + \ell_1 = v_{1,1} \\ \qquad u_{1,1}u_{1,0}\ell_3 - u_{1,0}\ell_2 + \ell_0 = v_{1,0} \end{cases},$$

or equivalently,

$$\begin{pmatrix} u_{1,1}^2 - u_{1,0} & -u_{1,1} & 1 & 0 \\ u_{1,1}u_{1,0} & -u_{1,0} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \ell_3 \\ \ell_2 \\ \ell_1 \\ \ell_0 \end{pmatrix} = \begin{pmatrix} v_{1,1} \\ v_{1,0} \end{pmatrix} . \tag{4}$$

Further linear equations are obtained from the second relation in Eq. (3). We have $u_1 u_2 := u_1(x)u_2(x) = x^4 + (u_{1,1} + u_{2,1})x^3 + (u_{1,0} + u_{2,0} + u_{1,1}u_{2,1})x^2 + (u_{1,1}u_{2,0} + u_{1,0}u_{2,1})x + u_{1,0}u_{2,0}$. Hence, letting $f := f(x) = x^5 + f_4 x^4 + f_3 x^3 + f_2 x^2 + f_1 x + f_0$ and $h := h(x) = h_2 x^2 + h_1 x + h_0$, we get after a little algebra $f + v_1 v_2 \bmod u_1 u_2 := F_3 x^3 + F_2 x^2 + F_1 x + F_0$ with

$$\begin{aligned}
F_3 &= u_{1,1}^2 + u_{2,1}^2 + u_{1,1}u_{2,1} - (u_{1,0} + u_{2,0}) - f_4(u_{1,1} + u_{2,1}) + f_3 \\
F_2 &= (u_{1,1} + u_{2,1} - f_4)(u_{1,0} + u_{,20} + u_{1,1}u_{2,1}) - (u_{1,1}u_{2,0} + u_{1,0}u_{2,1}) + f_2 + v_{1,1}v_{2,1} \\
F_1 &= (u_{1,1} + u_{2,1} - f_4)(u_{1,1}u_{2,0} + u_{1,0}u_{2,1}) - u_{1,0}u_{2,0} + f_1 + v_{1,1}v_{2,0} + v_{1,0}v_{2,1} \\
F_0 &= (u_{1,1} + u_{2,1} - f_4)u_{1,0}u_{2,0} + f_0 + v_{1,0}v_{2,0}
\end{aligned}$$

and $v(h + v_1 + v_2) \bmod u_1 u_2 := L_3 x^3 + L_2 x^2 + L_1 x + L_0$ with

$$\begin{aligned}
L_3 = {} & [h_2(u_{1,1}^2 + u_{2,1}^2 + u_{1,1}u_{2,1} - u_{1,0} - u_{2,0}) + H_0 - (u_{1,1} + u_{2,1})H_1]\ell_3 + {} \\
& [H_1 - h_2(u_{1,1} + u_{2,1})]\ell_2 + h_2\ell_1 \\
L_2 = {} & [h_2((u_{1,1} + u_{2,1})(u_{1,0} + u_{2,0} + u_{1,1}u_{2,1}) - (u_{1,1}u_{2,0} + u_{1,0}u_{2,1})) - {} \\
& H_1(u_{1,0} + u_{2,0} + u_{1,1}u_{2,1})]\ell_3 + [H_0 - h_2(u_{1,0} + u_{2,0} + u_{1,1}u_{2,1})]\ell_2 + H_1\ell_1 + h_2\ell_0 \\
L_1 = {} & [h_2((u_{1,1} + u_{2,1})(u_{1,1}u_{2,0} + u_{1,0}u_{2,1}) - u_{1,0}u_{2,0}) - H_1(u_{1,1}u_{2,0} + u_{1,0}u_{2,1})]\ell_3 + {} \\
& h_2(u_{1,1}u_{2,0} + u_{1,0}u_{2,1})\ell_2 + H_0\ell_1 + H_1\ell_0 \\
L_0 = {} & [h_2(u_{1,1} + u_{2,1})u_{1,0}u_{2,0} - H_1 u_{1,0}u_{2,0}]\ell_3 - h_2 u_{1,0}u_{2,0}\ell_2 + H_0\ell_0
\end{aligned}$$

where $H_1 = h_1 + v_{1,1} + v_{2,1}$ and $H_0 = h_0 + v_{1,0} + v_{2,0}$.

The previous relations hold over a field of any characteristic. In order to get a fair comparison with the best operation count in [5], we henceforth suppose that the underlying field $\mathbb{K}$ is such that $\operatorname{Char}\mathbb{K} \neq 2, 5$, in which case we can assume without loss of generality $h_2 = h_1 = h_0 = 0$ and $f_4 = 0$. The expressions for F_j and L_j then have a simpler form and, combining with Eq. (4), the previous relations become

$$\begin{pmatrix}
u_{1,1}^2 - u_{1,0} & -u_{1,1} & 1 & 0 \\
u_{1,1}u_{1,0} & -u_{1,0} & 0 & 1 \\
H_0 - (u_{1,1} + u_{2,1})H_1 & H_1 & 0 & 0 \\
-H_1(u_{1,0} + u_{2,0} + u_{1,1}u_{2,1}) & H_0 & H_1 & 0 \\
-H_1(u_{1,1}u_{2,0} + u_{1,0}u_{2,1}) & 0 & H_0 & H_1 \\
-H_1 u_{1,0}u_{2,0} & 0 & 0 & H_0
\end{pmatrix} \cdot \begin{pmatrix} \ell_3 \\ \ell_2 \\ \ell_1 \\ \ell_0 \end{pmatrix} = \begin{pmatrix} v_{1,1} \\ v_{1,0} \\ F_3 \\ F_2 \\ F_1 \\ F_0 \end{pmatrix} . \tag{5}$$

Multiplying row 1 by $-H_1$ and adding the resulting row to row 4 yields the smaller system

$$\begin{pmatrix} H_0-(u_{1,1}+u_{2,1})H_1 & H_1 \\ -H_1(u_{1,1}^2+u_{2,0}+u_{1,1}u_{2,1}) & H_0+H_1u_{1,1} \end{pmatrix} \cdot \begin{pmatrix} \ell_3 \\ \ell_2 \end{pmatrix} = \begin{pmatrix} F_3 \\ F_2-H_1v_{1,1} \end{pmatrix} \tag{6}$$

that can be solved for ℓ_3 and ℓ_2. The values of ℓ_1 and ℓ_0 can then be obtained from Eq. (4). The next step consists in reducing the so-obtained divisor $[u,v]$ to get $[\widetilde{u},\widetilde{v}]=[u_1,v_1]+[u_2,v_2]$. Letting $\widetilde{u}:=\widetilde{u}(x)=x^2+\widetilde{u}_{11}x+\widetilde{u}_{10}$ and $\widetilde{v}:=\widetilde{v}(x)=\widetilde{v}_{11}x+\widetilde{v}_{10}$, this can be achieved as presented in [5]; i.e.,

$$\begin{aligned} \widetilde{u}_{11} &= -(u_{1,1}+u_{2,1})-(1-2\ell_2\ell_3)/\ell_3^2\,, \\ \widetilde{u}_{10} &= -\big(u_{1,0}+u_{2,0}+u_{1,1}u_{2,1}+(u_{1,1}+u_{2,1})\widetilde{u}_{1,1}\big)+(2\ell_1\ell_3+\ell_2^2)/\ell_3^2\,, \\ \widetilde{v}_{11} &= -\big(\ell_3(\widetilde{u}_{1,1}^2-\widetilde{u}_{1,0})-\ell_2\widetilde{u}_{1,1}+\ell_1\big)\,, \\ \widetilde{v}_{10} &= -(\ell_3\widetilde{u}_{1,1}\widetilde{u}_{1,0}-\ell_2\widetilde{u}_{1,0}+\ell_0)\ \ . \end{aligned}$$

Altogether our unified addition algorithm can be evaluated using only $\underline{1\mathsf{I}+21\mathsf{M}+6\mathsf{S}}$. A detailed Magma implementation is provided in Appendix .

Conclusion

This paper presented efficient unified addition formulæ for hyperelliptic curve cryptography. Interestingly, the proposed formulæ only slightly increase the complexity and therefore provide a cost-efficient way to prevent simple side-channel attacks.

References

[1] R. M. Avanzi. Countermeasures against differential power analysis for hyperelliptic curve cryptosystems. In C. D. Walter et al., editors, *Cryptographic Hardware and Embedded Systems – CHES 2003*, volume 2779 of *Lect. Notes in Comp. Sci.*, pages 366–381. Springer, 2003.

[2] É. Brier and M. Joye. Weierstraß elliptic curves and side-channel attacks. In D. Naccache and P. Paillier, editors, *Public Key Cryptography*, volume 2274 of *Lect. Notes in Comp. Sci.*, pages 335–345. Springer-Verlag, 2002.

[3] É. Brier, I. Déchène, and M. Joye. Unified point addition formulæ for elliptic curve cryptosystems. In N. Nedjah and L. de Macedo, editors, *Embedded Cryptographic Hardware: Methodologies & Architectures*, pages 247–256. Nova Science Publishers, 2004.

[4] D. G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177): 95–101, 1987.

[5] C. Costello and K. Lauter. Group law computations on Jacobians of hyperelliptic curves. Cryptology ePrint Archive, Report 2011/306, 2011. `http://eprint.iacr.org/`.

[6] S. D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.

[7] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comp.*, 76(257):475–492, 2007.

[8] N. Koblitz. Hyperelliptic cryptosystems. *J. Cryptology*, 1(3):139–150, 1989.

[9] N. Koblitz. *Algebraic Aspects of Cryptography*. Springer, 1998.

[10] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lect. Notes in Comp. Sci.*, pages 104–113. Springer-Verlag, 1996.

[11] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO ’99*, volume 1666 of *Lect. Notes in Comp. Sci.*, pages 388–397. Springer-Verlag, 1999.

[12] T. Lange. Formulae for arithmetic on genus 2 hyperelliptic curves. *Appl. Algebra Eng. Commun. Comput.*, 15(5):295–328, 2005.

Magma Implementation

```
UniAddLaw := function(D1,D2)
   J := Parent(D1);
   C := Curve(J);
   Fq := BaseRing(C);
  _<x> := PolynomialRing(Fq);
   f,h := HyperellipticPolynomials(C);

   u1 := D1[1]; v1 := D1[2];
   u2 := D2[1]; v2 := D2[2];

   f2 := Coefficient(f,2); f3 := Coefficient(f,3);

   u11 := Coefficient(u1,1); u10 := Coefficient(u1,0);
   u21 := Coefficient(u2,1); u20 := Coefficient(u2,0);
   v11 := Coefficient(v1,1); v10 := Coefficient(v1,0);
   v21 := Coefficient(v2,1); v20 := Coefficient(v2,0);

   U11 := u11^2; U10 := u11*u10;
   U21 := u21^2; U20 := u21*u20;
   Su1 := u11 + u21; Su0 := u10 + u20;

   Pu1 := (Su1^2 - U11 - U21)/2; // instead of Pu1 := u11*u21;
   H1 := v11 + v21; H0 := v10 + v20;
   M1 := H0 - H1*Su1; M2 := H1;
   M3 := -H1*(U11 + Pu1 + u20); M4 := H0 + u11*H1;
   z1 := f2 + Su1*Pu1 + U10 + U20 - v11^2;
   z2 := f3 + U11 + U21 + Pu1 - Su0;

   t1 := (z1 + M3)*(z2 - M1); t2 := (z1 - M3)*(z2 + M1);
   t3 := (z1 + M4)*(z2 - M2); t4 := (z1 - M4)*(z2 + M2);
   d := t3 + t4 - t1 - t2 - 2*(M3 - M4)*(M1 + M2);

   l2 := t2 - t1; l3 := t3 - t4;

   A := 1/(d*l3); B := d*A; C := d*B; D := l2*B; E := l3^2*A; C2 := C^2;

   utilde11 := 2*D - C2 - Su1;
   utilde10 := D^2+C*(v11+v21)-((utilde11-C2)*Su1 + (U11 + U21))/2;
   Utilde11 := utilde11^2;
   Utilde10 := utilde11*utilde10;
   vtilde11 := D*(u11 - utilde11) + Utilde11 - utilde10 - U11 + u10;
   vtilde10 := D*(u10 - utilde10) + Utilde10 - U10;
   vtilde11 := -(E*vtilde11 + v11);
   vtilde10 := -(E*vtilde10 + v10);

   utilde := xˆ2 + utilde11*x + utilde10;
   vtilde := vtilde11*x + vtilde10;

   return J![utilde,vtilde];
end function;
```

O. Diao Université de Rennes I
oumar.diao@univ-rennes1.fr
M. Joye Technicolor
marc.joye@technicolor.com

On the probability of generating a lattice

Felix Fontein and Pawel Wocjan

Introduction

One of the mathematical primitives many public-key cryptosystems are based on is the *Discrete Logarithm Problem* (DLP). These are based on many different kind of groups; examples include the multiplicative group of $\mathbb{F}_q$ [9], the group of $\mathbb{F}_q$-rational points of an elliptic curve [3], more generally the divisor class group of an algebraic curve, or the ideal class group or infrastructure of an algebraic number field [1, 13]. For most of these groups, subexponential algorithms exist which can solve the DLP. It is only in the case of low genus curves that many instances were found for which only exponential algorithms are known on classical computers. On classical computers, for almost all instances, no polynomial time algorithms are known.

On the other hand, on quantum computers, polynomial time algorithms are known which solve these DLPs [12, 2, 7, 6, 15, 14]. Assuming large enough quantum computers can be built, cryptosystems based on the DLP are not secure anymore.

Even though all these quantum algorithms are polynomial time algorithms, some of them are much more efficient than others. In particular, the algorithms for solving the DLP in the infrastructure of a number field of unit rank ≥ 2 have the worst performance of all of them [5]. The main problem is that the involved lattice is not discrete anymore, as in the other cases where one essentially has finite abelian groups. In the infrastructure of a number field, one works in a torus $T = \mathbb{R}^n/\Lambda$, where Λ is a lattice of full rank in $\mathbb{R}^n$ [4]. The coefficients of any non-trivial vector of Λ are transcendental, whence one has to work with approximations. Solving the DLP can be reformulated as a lattice problem. The task is to find a basis of a lattice $\Lambda' \subseteq \mathbb{R}^{n+1}$, where vectors with a non-zero entry in the last component yield the desired solution of the DLP.

To find a basis of Λ', the quantum algorithm has a mechanism which, with a certain probability $p_1 > 0$, outputs an essentially uniformly distributed vector $\lambda^* \in (\Lambda')^* \cap [0,B)^{n+1}$, where $(\Lambda')^*$ is the dual lattice of Λ' and where $B > 0$ is suitably large. If one has $\lambda_1^*, \dots, \lambda_m^*$ with $(\Lambda')^* = \langle \lambda_1^*, \dots, \lambda_m^* \rangle_{\mathbb{Z}}$, one can compute a basis of $(\Lambda')^*$ out of these vectors and then use linear algebra to retrieve a basis of Λ' itself.

To compute the success probability of the algorithm, one has to consider the probability that the m sampled vectors are actually in $(\Lambda')^*$, and the probability that m random vectors from $(\Lambda')^* \cap [0,B)^{n+1}$ generate $(\Lambda')^*$. If the latter probability is p_2, then the overall success probability is $\approx p_1^m p_2$, and one expects that one has to run the algorithm $\approx (p_1^m p_2)^{-1}$ times before it outputs a basis of $(\Lambda')^*$ and thus of Λ' itself.

The main problem is that for $n > 1$, the lower bound one can prove for p_1 is quite small. In [5] we have explicitly specified the probabilities, and showed that already for $n = 2$, the success probability is so small that the algorithm, although being polynomial, will not have any practical relevance even if large enough quantum computers can be built.

Therefore, one wants to minimize the value of m. In this extended abstract, we want to present the to our knowledge first correct bound on p_2, using $m = 2(n+1)+1$. We also need to use two different window sizes B: the first $n+1$ vectors are sampled from a smaller window $[0,B)^{n+1}$, and the latter $(n+1)+1$ vectors from a larger window $[0,B_1)^{n+1}$ with $B_1 > B$.

It is our hope that our work will provide more attention to this problem, and hopefully also inspire others to search for bounds for smaller values of m.

P. W. gratefully acknowledges the support from the NSF grant CCF-0726771 and the NSF CAREER Award CCF-0746600. F. F. gratefully acknowledges the support form Armasuisse and the SNF grant No. 132256.

Solving the Problem

To simplify notation, we from now on use the lattice $\Lambda \subseteq \mathbb{R}^n$ of rank n, instead of using the lattice $(\Lambda')^* \subseteq \mathbb{R}^{n+1}$ of rank $n+1$. Thus we are working with $m = 2n+1$ vectors.

We solve our problem in two steps. First, we consider the probability that n vectors sampled uniformly at random from Λ generate a sublattice Λ_1 of full rank, i.e. do not lie in a hyperplane. Then, we compute the probability that the residue classes of the next $n+1$ vectors generate the finite abelian quotient group Λ/Λ_1. Finally, we combine these two results.

In the following, we assume that $n > 1$. In case $n = 1$, one can easily show that two random vectors from $[0,B) \cap \Lambda$ generate Λ with probability greater than $\frac{3^3}{\pi^2 2^3} > \frac{1}{3}$ provided that $B \geq 3 \det \Lambda + 1$.

Note that our approach is very similar to the one presented in [14]. The first part of the approach is identical, while the second is different. The differences will be discussed in more detail in Section .

Generating a Sublattice of Full Rank

Note that $\lambda_1, \dots, \lambda_n \in \Lambda \cap [0,B)^n$ generate a sublattice of full rank if and only if they are linearly independent over $\mathbb{R}$. This is the case if λ_i is not contained in the $(i-1)$-dimensional hyperplane generated by $\lambda_1, \dots, \lambda_{i-1}$. This allows us to find the following bound on the probability that n random vectors generate a sublattice of full rank:

Proposition 1. *Assume that* $B \geq \max\{8n-2, n^{(n-1)/2} 2^{n+1} - 2\} \cdot \nu(\Lambda)$. *Let*

$$X := (\Lambda \cap [0,B)^n)^n \textit{ and } Y := \{(y_1, \dots, y_n) \in X \mid \operatorname{span}_{\mathbb{R}}(y_1, \dots, y_n) = \mathbb{R}^n\}.$$

Then $|Y| \geq \frac{1}{4}|X|$.

In the proposition, $\nu(\Lambda)$ denotes the covering radius of Λ. Note that $\nu(\Lambda) \leq \frac{1}{2} n^{n/2+1} \frac{\det \Lambda}{\lambda_1(\Lambda)^{n-1}}$, where $\lambda_1(\Lambda)$ denotes the first successive minimum of Λ [10], i.e. the length of a shortest vector in Λ. The proof proceeds by using lower and upper bounds on the number of lattice points in certain convex sets, similar to the bounds of Proposition 8.7 in [10].

Generating a Finite Abelian Group

In case Λ_1 is a sublattice of full rank of Λ, the quotient group $G = \Lambda/\Lambda_1$ is a finite abelian group. Its order equals the index $[\Lambda : \Lambda_1]$, and by the Elementary Divisor Theorem, it can be generated by n elements.

Proposition 2. *Let G be a finite abelian group known to be generated by n elements. Then the probability that $n+1$ elements drawn uniformly at random from G generate G is at least* $\hat{\zeta} := \prod_{i=2}^{\infty} \zeta(i)^{-1} \geq 0.434$, *where ζ denotes the Riemann zeta function.*

Note that if one just requires n elements instead of $n+1$, one can find a sequence of finite abelian groups generated by n elements such that the probability that they are generated by n randomly selected elements goes down to 0. This shows that our approach will not work with less than $2n+1$ elements, if the desired bound on the probability should be independent of n and B.

This result can be shown by considering the Sylow decomposition of G and by using a result in [11] on the probability that the p-Sylow subgroup is generated by $n+1$ elements.

The Final Result

Assume that the first n sampled vectors from $\Lambda \cap [0,B)^n$ generate a sublattice Λ_1 of full rank. Then $G = \Lambda/\Lambda_1$ is a finite abelian group which can be generated by n elements. Thus if we sample $n+1$ elements $\lambda + \Lambda_1$ from G in a uniform random manner, we can bound the probability that they generate G. In case $G = \langle \lambda_{n+1} + \Lambda_1, \dots, \lambda_{2n+1} + \Lambda_1 \rangle$ and $\Lambda_1 = \langle \lambda_1, \dots, \lambda_n \rangle$, we have $\Lambda = \langle \lambda_1, \dots, \lambda_n, \lambda_{n+1}, \dots, \lambda_{2n+1} \rangle$.

The main problem is that we cannot directly sample uniformly at random from G: if we choose $\lambda \in \Lambda \cap [0,B)^n$ uniformly at random, then $\lambda + \Lambda_1$ will in general be not uniformly distributed in $G = \Lambda/\Lambda_1$. By enlarging the window $[0,B)^n$ to $[0,B_1)^n$ with $B_1 > B$ large enough, we can ensure that the residue classes of the samples $\lambda \in \Lambda \cap [0,B_1)^n$ are essentially distributed uniformly at random in G.

This can be made more concrete:

Theorem 3. *Let Λ be a lattice of full rank in $\mathbb{R}^n$, and assume that $B \geq \max\{8n-2, n^{(n-1)/2}2^{n+1} - 2\} \cdot \nu(\Lambda)$ and $B_1 \geq 8n^2(n+1)b$. If n vectors are selected uniformly at random from $\Lambda \cap [0,B)^n$ and $n+1$ vectors uniformly at random from $\Lambda \cap [0,B_1)^n$, then the probability that all these vectors generate Λ is at least $\frac{1}{4}\left(\hat{\zeta} - \frac{1}{4}\right) \geq 0.046$.*

This proposition is similar to Satz 2.4.23 in [14]. We emphasize that our bound on the success probability is constant, whereas the bound presented in Satz 2.4.23 decreases exponentially fast with the dimension n. The first part of proof of Satz 2.4.23 (concerning the generation of a full-rank sublattice) is unfortunately not correct, but can be corrected as we have shown in our proof of Proposition 1. The idea behind the second part is completely different from our proof and cannot be used to prove a constant success probability. Perhaps it could be used to prove that only $2n$ random elements (as opposed to $2n+1$ elements) are needed to guarantee a non-zero success probability.

Note that for a fixed dimension n, one can obtain better bounds. The proofs of the above results yield a lower bound on the success probability of $\left(\prod_{i=2}^{n+1} \zeta(i)^{-1} - \frac{1}{4}\right) \cdot \prod_{i=1}^{n-1}(1-2^{-i})$. For $n = 2$, 3, 4 and 5, this is larger than 0.127, 0.081, 0.065 and 0.059, respectively.

A Conjecture

We conjecture that for a sufficiently large B, already $n+1$ vectors from $\Lambda \cap [0,B)^n$ should suffice. To see this, fix a basis $b_1, \dots, b_n$ of Λ. If $\lambda_1, \dots, \lambda_m \in \Lambda$ are elements, they can be represented in terms of the b_i's via $\lambda_j = \sum_{i=1}^n a_{ij} b_i$. One then has that the matrix $A = (a_{ij})_{ij} \in \mathbb{Z}^{n \times m}$ is *unimodular* if and only if $\Lambda = \langle \lambda_1, \dots, \lambda_m \rangle$. Moreover, if $X := \{(a_1, \dots, a_n) \in \mathbb{R}^n \mid \sum_{i=1}^n a_i b_i \in [0,B)^n\}$, then selecting m vectors uniformly at random from $\Lambda \cap [0,B)^n$ is equivalent to choosing an integer matrix A with columns uniformly at random from $X \cap \mathbb{Z}^n$.

This shows that the probability we seek equals the probability that a random integer matrix with columns selected uniformly at random from a convex set X is unimodular. In case the chosen basis is strongly reduced, the set X will be rather “nice”. In support of our conjecture, we note that in [8] it has been shown that the probability that a random integer matrix in $\{0, \dots, B-1\}^{n \times m}$ is invertible goes to $\prod_{j=m-n+1}^m \zeta(j)^{-1}$ for $B \to \infty$. As soon as $m > n$, this can be bounded from below by $\hat{\zeta} \geq 0.434$.

References

[1] J. A. Buchmann. Number theoretic algorithms and cryptology. In *FCT '91: Proceedings of the 8th International Symposium on Fundamentals of Computation Theory*, pages 16–21, London, UK, 1991. Springer-Verlag.

[2] K. K. H. Cheung and M. Mosca. Decomposing finite abelian groups. *Quantum Information & Computation*, 1(3):26–32, 2001.

[3] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

[4] F. Fontein. The infrastructure of a global field of arbitrary unit rank. *Math. Comp.*, 80(276): 2325–2357, 2011.

[5] F. Fontein and P. Wocjan. Quantum algorithm for computing the period lattice of an infrastructure. `http://arxiv.org/abs/1111.1348`, 2012.

[6] S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 653–658 (electronic), New York, 2002. ACM.

[7] S. Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474. ACM, New York, 2005.

[8] G. Maze, J. Rosenthal, and U. Wagner. Natural density of rectangular unimodular integer matrices. `http://arxiv.org/abs/1005.3967`, 2010.

[9] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*. CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997. With a foreword by Ronald L. Rivest.

[10] D. Micciancio and S. Goldwasser. *Complexity of lattice problems*. The Kluwer International Series in Engineering and Computer Science, 671. Kluwer Academic Publishers, Boston, MA, 2002. A cryptographic perspective.

[11] C. Pomerance. The expected number of random elements to generate a finite abelian group. *Periodica Mathematica Hungarica*, 43(1–2):191–198, 2001.

[12] P. Sarvepalli and P. Wocjan. Quantum algorithms for one-dimensional infrastructures. `http://arxiv.org/abs/1106.6347`, 2011.

[13] R. Scheidler, J. A. Buchmann, and H. C. Williams. A key-exchange protocol using real quadratic fields. *J. Cryptology*, 7(3):171–199, 1994.

[14] A. Schmidt. *Zur Lösung von zahlentheoretischen Problemen mit klassischen und Quantencomputern*. Ph.D. thesis, Technische Universität Darmstadt, 2007.

[15] A. Schmidt and U. Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field (extended abstract). In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480. ACM, New York, 2005.

F. Fontein University of Zurich
felix.fontein@math.uzh.ch
P. Wocjan University of Central Florida
wocjan@eecs.ucf.edu

Collisions in compositions of triangular polynomial systems and hash functions

Domingo Gómez-Pérez, Jaime Gutierrez, Alina Ostafe, and Igor Shparlinski

Hash functions are deterministic procedures that take a block of data of arbitrary length and digest it into a string of fixed size. They are of special importance because they are commonly used in digital signatures and due to the NIST hash function competition, hash functions have attracted considerable attention.

In [3], the authors proposed a new construction of hash functions based on iterations of polynomial systems. This construction is motivated by that of D. X. Charles, E. Z. Goren and K. E. Lauter [1] and in some sense it may be considered as its extension.

We recall the construction of the hash function proposed in [3]. Let n, s and r be positive integers. Choose a random n-bit prime $p > 2$, $q = p^s$ and 2^r permutation polynomial systems

$$\mathcal{R}_\ell = \{R_{\ell,1}, \dots, R_{\ell,m}\}, \quad R_{\ell,i} \in \mathbb{F}_q[X_1, \dots, X_m],$$

$i = 1, \dots, m,\ \ell = 0, \dots, 2^r - 1$, not necessary distinct. We also consider a random initial vector $\vec{w}_0 \in \mathbb{F}_q^m$.

As in [1], the input of the hash function is used to decide what polynomial system $\mathcal{F}_\ell$ is used to iterate. More precisely, it works as follows given an input bit string Σ, we execute the following steps:

- Pad Σ with at most $r-1$ zeros on the left to make sure that its length L is a multiple of r.
- Split Σ into blocks σ_j, $j = 1, \dots, J$, where $J = L/r$, of length r and interpret each block as an integer $\ell_j \in [0, 2^r - 1]$.
- Starting at the vector $\vec{w}_0$, apply the polynomial systems $\mathcal{R}_{\ell_j}$ iteratively obtaining the sequence of vectors $\vec{w}_j \in \mathbb{F}_q^m$:
$$\vec{w}_j = \mathcal{R}_{\ell_j}(\vec{w}_{j-1}), \qquad j = 1, \dots, J.$$
- Output $\vec{w}_J$ as the value of the hash function (which can also be now interpreted as a binary mns-bit string).

The above construction is quite similar to that of [1] where $m = 2$, the vectors $\vec{w}_j$ represent the coefficients of an equation describing an elliptic curve for example, of the Weierstrass equation

$$Y^2 = X^3 + aX + b,$$

and polynomials maps are associated with isogenies of a fixed degree.

As remarked in [3], the initial vector $\vec{w}_0$ is fixed and in particular, does not depend on the input of the hash function. Furthermore, the collision resistance does not rely on the difficulty of inverting the maps generated by the polynomial systems $\mathcal{R}_\ell$. Rather, it is based on the difficulty of making the decision which system to apply at each step when one attempts to back trace from a given output to the initial vector $\vec{w}_0$ and thus produce two distinct strings Σ_1 and Σ_2 of the same length L, with the same output.

We remark that the condition $L \equiv 0 \pmod{r}$ is necessary to avoid collisions between messages of different lengths. It is enough to take $\Sigma_2 = (0, \Sigma_1)$ (that is, Σ_2 is obtained from Σ_1 by augmenting it by 0). If $L \not\equiv 0 \pmod{r}$ then they lead to the same output.

The goal of this talk is to study collisions in compositions of polynomial systems within certain classes of systems. We aim to construct concrete classes of polynomial systems that are J-collision free, that is, the composition of any J systems in these classes is unique. We study two classes of triangular polynomial systems, which come in two different flavors,

- *slow degree growth*, that is polynomial systems $\mathcal{F}_\ell = \{F_{\ell,1},\dots,F_{\ell,m}\}$ of the form

  $$\begin{aligned} F_{\ell,i} &= X_i G_{\ell,i}(X_{i+1},\dots,X_m) + H_{\ell,i}(X_{i+1},\dots,X_m), \quad i = 1,\dots,m-1, \\ F_{\ell,m} &= g_{\ell,m}X_m + h_{\ell,m}, \end{aligned} \tag{1}$$

  where $G_{\ell,i}, H_{\ell,i} \in \mathbb{F}_q[X_{i+1},\dots,X_m]$, $g_{\ell,m} \in \mathbb{F}_q^*$, $\ell = 0,\dots,2^r-1$;

- *exponential degree growth and sparse representation*, that is polynomial systems $\mathcal{F}_\ell = \{F_{\ell,1},\dots,F_{\ell,m}\}$ of the form

  $$\begin{aligned} F_{k,i} =& (X_i - h_i)^{e_{k,i}} G_{k,i} + h_i, \quad i = 1,\dots,m-1, \\ F_{k,m} =& g_{k,m}(X_m - h_m)^{e_{k,m}} + h_m, \end{aligned} \tag{2}$$

  where $G_{\ell,i} \in \mathbb{F}_q[X_{i+1},\dots,X_m]$, $h_i, g_{\ell,m} \in \mathbb{F}_q$, and $g_{\ell,m} \neq 0$ for all $i = 1,\dots,m-1$ and $\ell = 0,\dots,2^r-1$.

We remark that the problem of collisions of polynomials has been previously studied in [4] for a special class of linearized univariate polynomials of degree p^2.

In this paper we consider the hash function described above using triangular polynomial systems of the form (1) or (2). It is conceivable that the triangular shape and linearity of F_i in X_i or sparsity of F_i in the systems (1) or (2), respectively, can be a weakness from the cryptographic point of view. As suggested in [2], a way to overcome this potential weakness is based on using polynomial automorphisms.

Let $\mathcal{A} = \{A_1,\dots,A_m\}$ be an arbitrary polynomial automorphism in m variables in $\mathbb{F}_q[X_1,\dots,X_m]$, that is, there exists a system of polynomials $\mathcal{A}^{-1} = \{A_1^{-1},\dots,A_m^{-1}\}$ such that for their composition we have $\mathcal{A}^{-1} \circ \mathcal{A} = \{X_1,\dots,X_m\}$.

We consider systems of the form

$$\mathcal{R}_\ell = \{R_{\ell,1},\dots,R_{\ell,m}\} = \mathcal{A}^{-1} \circ \mathcal{F}_\ell \circ \mathcal{A}, \tag{3}$$

where $\mathcal{F}_\ell$ is of the form (1) or (2). We use our results on collisions of compositions of triangular polynomial systems to study the hash function defined using the systems (3), and in particular, we give estimates on the number of collissions.

References

[1] D. X. Charles, K. E. Lauter, and E. Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.

[2] D. Gómez-Pérez, A. Ostafe, and I. E. Shparlinski. Algebraic entropy, automorphisms and sparsity of algebraic dynamical systems and pseudorandom number generators, 2012.

[3] A. Ostafe and I. Shparlinski. Pseudorandom numbers and hash functions from iterations of multivariate polynomials. *Cryptography and Communications*, 2(1):49–67, 2010.

[4] J. von zur Gathen, M. Giesbrecht, and K. Ziegler. Composition collisions and projective polynomials: statement of results. In W. Koepf, editor, *ISSAC*, pages 123–130. ACM, 2010.

D. Gómez-Pérez University of Cantabria
domingo.gomez@unican.es

J. Gutierrez University of Cantabria
jaime.gutierrez@unican.es

A. Ostafe Macquarie University
alina.ostafe@mq.edu.au

I. Shparlinski Macquarie University
igor.shparlinski@mq.edu.au

Stable polynomials and irreducible divisors of iterated polynomials

Domingo Gómez-Pérez, Alina Ostafe, and Igor E. Shparlinski

Introduction

Let q be an odd prime power and $\mathbb{F}_q$ be the finite field with q elements. For a polynomial $f \in \mathbb{F}_q[X]$ we define the sequence of iterations:

$$f^{(0)} = X, \qquad f^{(n)} = f\left(f^{(n-1)}\right), \quad n = 1, 2, \ldots.$$

Following [2, 3, 8], we say that the polynomial $f \in \mathbb{F}_q[X]$ is *stable* if all polynomials $f^{(n)}$ are irreducible over $\mathbb{F}_q$, $n \geq 1$.

The goal of this talk is to present several recent results on stable polynomials over finite fields as well as new results regarding the growth of irreducible factors of polynomial iterates. We also outline some work in progress and formulate several open questions. This work is a step towards better understanding of the algebraic structure of iterated polynomials over finite fields that plays an important role in studying pseudorandom number generators, see [10].

Stability of polynomials

Studying the stability of polynomials has proved to be a very hard problem and only the quadratic polynomial case over finite fields is fairly understood. As in [8], for a quadratic polynomial $f = aX^2 + bX + c \in \mathbb{F}_q[X]$, $a \neq 0$, we define $\gamma = -b/2a$ as the unique critical point of f (that is, the zero of the derivative f') and consider the set, called the *critical orbit* of f,

$$\mathrm{Orb}(f) = \{f^{(n)}(\gamma) \ : \ n = 2, 3, \ldots, t_f\},$$

where t_f is the smallest value of t such that $f^{(t)}(\gamma) = f^{(s)}(\gamma)$ for some positive integer $s < t$. The following result is well known [3, 8]:

Theorem 1. *Let $f = aX^2 + bX + c \in \mathbb{F}_q[X]$ and γ as above. Then f is stable if and only if the adjusted critical orbit*

$$\overline{\mathrm{Orb}}(f) = \{-f(\gamma)\} \cup \mathrm{Orb}(f)$$

contains no squares, that is, if and only if, $\chi\left(f^{(n)}(\gamma)\right) = -1$, $n = 2, \ldots, t_f$, where χ is the quadratic character of $\mathbb{F}_q$.

Theorem 1 shows that the stability of quadratic polynomials over $\mathbb{F}_q$ can be tested in at most q steps by simply examining $-f(\gamma)$ and each element of $\mathrm{Orb}(f)$. In [11], using Theorem 1 and methods from analytic number theory, we significantly reduced this bound.

Theorem 2. *For any odd q and any stable quadratic polynomial $f \in \mathbb{F}_q[X]$ we have*

$$t_f = O\left(q^{3/4}\right).$$

However, the case of an arbitrary polynomial $f \in \mathbb{F}_q[X]$ is not yet settled. The only known result in this case has been proved in [6] using new techniques based on resultants of polynomials together with the Stickelberger's theorem [13].

Theorem 3. *Let $f \in \mathbb{F}_q[X]$ be a stable polynomial with leading coefficient a_d, non constant derivative f', $\deg f' = k \le d-1$. Let us suppose that γ_i, $i = 1,\ldots,k$, are the roots of the derivative f'. Then*

1. *if $d = \deg f$ is even,*
$$\mathcal{S}_1 = \left\{ a_d^k \prod_{i=1}^{k} f^{(n)}(\gamma_i) \mid n > 1 \right\} \cup \left\{ (-1)^{\frac{d}{2}} a_d^k \prod_{i=1}^{k} f(\gamma_i) \right\}$$
contains only nonsquares in $\mathbb{F}_q$;
2. *if $d = \deg f$ is odd,*
$$\mathcal{S}_2 = \left\{ (-1)^{\frac{(d-1)}{2}+k}(k+1)a_{k+1}a_d \prod_{i=1}^{k} f^{(n)}(\gamma_i) \mid n \ge 1 \right\},$$
where a_{k+1} is the coefficient of X^{k+1} in f, contains only squares in $\mathbb{F}_q$.

Applying now the same technique with multiplicative character sums as in [11, Theorem 1] (as the argument does not depend on the degree of the polynomial f), we have the following estimate, see [6]:

Theorem 4. *For any odd q and any stable polynomial $f \in \mathbb{F}_q[X]$ with irreducible derivative f', $\deg f' = k$, we have*
$$\#\mathcal{S}_1, \#\mathcal{S}_2 = O\left(q^{3k/4}\right).$$

Gomez and Nicolás [5] have proved that there are $O\left(q^{5/2}(\log q)^{1/2}\right)$ stable quadratic polynomials over $\mathbb{F}_q$ for an odd prime power q, while in [6] it is proved that there are $O(q^{d+1-1/\log(2d^2)})$ stable polynomials of degree $d \ge 2$ over $\mathbb{F}_q$ (where $\log z$ denotes the binary logarithm of z).

Irreducible divisors of iterated polynomials

In [7] we continue to study the arithmetic properties of iterated polynomials and show that for almost all polynomials f of a fixed degree d over $\mathbb{F}_q$, the nth iteration $f^{(n)}$ has a square-free factor of degree of order at least $n^{1+o(1)}$ as $n \to \infty$ (uniformly over q). This result is a combination of two different approaches.

First, we combine the method of Gomez and Nicolás [5] with some new ideas to show that for almost all quadratic polynomials $f \in \mathbb{F}_q[X]$ the number $r_n(f)$ of irreducible divisors of the nth iterate $f^{(n)}$ grows at least linearly with n if n is of order up to $\log q$. This immediately implies that the largest degree of the irreducible divisors of $f^{(n)}$ grows with n as well. Our tools to prove this are *resultants* of iterated polynomials, the *Stickelberger's Theorem* [13] and estimates of certain *character sums*, see [7].

Theorem 5. *If q is odd then for any fixed $\varepsilon > 0$ for all but $o(q^{d+1})$ polynomials $f \in \mathbb{F}_q[X]$ of degree d, we have*
$$r_n(f) \ge (0.5 + o(1))n,$$
when $n \to \infty$ and $L \ge n$, where
$$L = \left\lceil \left(\frac{1}{2\log d} - \varepsilon \right) \log q \right\rceil.$$

Let $f = f_d X^d + \ldots + f_1 X + f_0 \in \mathbb{F}_q[X]$ be a polynomial of degree $d \ge 2$ with leading coefficient f_d and non-constant derivative f' of degree $k \le d-1$. It is convenient to introduce the following notation
$$G_k(f_d,\ldots,f_0) = \prod_{i=1}^{k} f^{(n)}(\gamma_i), \quad n \ge 1,$$

where γ_i, $i = 1, \dots, k$, are the roots of f', which is clearly a polynomial in $f_d, \dots, f_0$.

For even d and an integer n, we consider the character sum

$$T_1(n) = \sum_{f_0 \in \mathbb{F}_q} \cdots \sum_{f_d \in \mathbb{F}_q} \left| \sum_{\ell=1}^{n} \chi(G_\ell(f_d, \dots, f_0) G_{\ell+1}(f_d, \dots, f_0)) \right|^2,$$

where χ is a quadratic character.

For odd d, we consider the character sum

$$T_2(n) = \sum_{f_0 \in \mathbb{F}_q} \cdots \sum_{f_d \in \mathbb{F}_q} \left| \sum_{\ell=1}^{n} \chi\left(f_d^{k\ell} G_\ell(f_d, \dots, f_0) \right) \right|^2,$$

where $k \leq d-1$ is the degree of the derivative f'.

In [7], using the same technique as in [5], we prove the following bounds:

Lemma 6. *Let $f = f_d X^d + \ldots + f_1 X + f_0 \in \mathbb{F}_q[X]$ be defined as above. For any integer $n \geq 1$, we have the following bounds:*

$$T_i(n) = O\left(n^2 d^n q^{d+1/2} + n^2 d^{2n} q^d + n q^{d+1} \right), \quad i = 1, 2.$$

Using now Lemma 6 and the Stickelberger's theorem [13] to show that $r_k(f)$ and $r_{k+1}(f)$ are of different parity for $n/2 + O(n^{2/3})$ values of $k = 1, \dots, n$, we prove Theorem 5.

Beyond this threshold, in [7] we use a different technique, related to Mason's proof of the *ABC*-conjecture in its polynomial version, see [9, 12], to prove a lower bound on the largest degree $D_n(f)$ of the irreducible divisors of $f^{(n)}$.

Theorem 7. *Let $f \in \mathbb{F}_q[X]$ be of degree d with $\gcd(d,q) = 1$ and such that $f \neq f_d X^d$. Then*

$$D_n(f) \gg \frac{1}{\log q} n.$$

Note that Theorem 7 becomes nontrivial for n of about the same level when Theorem 5 stops working. So they can be combined in the following result that provides some nontrivial information about the arithmetic structure of iterations that applies to all n and q, see [7]. Let $\Delta_n(f)$ denotes the largest degree of square-free fractors of $f^{(n)}$.

Theorem 8. *If q is odd and $\gcd(d,q) = 1$ then, for any fixed $\varepsilon > 0$, for all but $o(q^{d+1})$ polynomials $f \in \mathbb{F}_q[X]$ of degree d, for $n \geq 1$, we have*

$$\Delta_n(f) \gg n^{1-\varepsilon}.$$

Open questions

We note that in Theorem 3 only a necessary condition for the stability of a polynomial f over $\mathbb{F}_q$ was given. However, no necessary and sufficient condition is known for the stability of arbitrary polynomials over a finite field.

Moreover, we note that the results of [6] hold only over a field of odd characteristic. Study the stability of $f \in \mathbb{F}_{2^s}[X]$, $s \geq 1$, of degree $d \geq 3$, is certainly of interest. We note that no quadratic polynomial is stable over binary finite fields, see [1].

Another interesting question is to extend the bound of Theorem 5 to any n (beyond of the current threshold $n = O(\log q)$).

The critical orbit of quadratic polynomials f, $\overline{\mathrm{Orb}}(f)$, coincides with the following set

$$\{G_n(f_0, f_1, f_2) \mid n \geq 1\}.$$

It is certainly interesting to investigate various properties of the sequence $u_n = G_n(f_0,\dots,f_d)$ for $f_0,\dots,f_d \in \mathbb{F}_q$ fixed corresponding to a polynomial $f = f_d X^d + \dots + f_0 \in \mathbb{F}_q[X]$.

At this moment, only results for quadratic polynomials are known. For example, the sequence u_n becomes eventually periodic when $d = 2$. If f' is a irreducible polynomial of degree k, then $G_n(f_0,\dots,f_d) = \mathrm{Nm} f^{(n)}(\gamma)$ is the norm of $f^{(n)}(\gamma)$ in $\mathbb{F}_q$. Apart from these two cases, very little is known for general polynomials f.

The sparsity, or number of monomials, is another important characteristic of polynomials and it is certainly interesting to obtain lower bounds on the number of monomials of the iterations $f^{(n)}$. For iterations of polynomials and even rational functions over a field of characteristic zero such bounds can be derived from the results of [4].

References

[1] O. Ahmadi, F. Luca, A. Ostafe, and I. E. Shparlinski. On stable quadratic polynomials. *Glasgow Math. J.*, (in press).

[2] N. Ali, ‘Stabilité des polynômes’. *Acta Arith.*, 119: 53–63, 2005.

[3] M. Ayad and D. L. McQuillan. Irreducibility of the iterates of a quadratic polynomial over a field. *Acta Arith.*, 93: 87–97, 2000; Corrigendum: *Acta Arith.*, 99: 97, 2001.

[4] C. Fuchs and U. Zannier. Composite rational functions expressible with few terms. *J. Eur. Math. Soc.*, 14: 175–208, 2012.

[5] D. Gomez and A. P. Nicolás. An estimate on the number of stable quadratic polynomials. *Finite Fields and Appl.*, 16: 329–333, 2010.

[6] D. Gomez-Perez, A. P. Nicolás, A. Ostafe, and D. Sadornil. Stable polynomials over finite fields. *Submitted*, 2011.

[7] D. Gomez-Perez, A. Ostafe, and I. E. Shparlinski. On irreducible divisors of iterated polynomials. *In progress*, 2012.

[8] R. Jones and N. Boston. Settled polynomials over finite fields. *Proc. Amer. Math. Soc.*, 140: 1849–1863, 2012.

[9] R. C. Mason. *Diophantine Equations over Functions Fields*. Cambridge, Cambridge Univ.Press, 1984.

[10] A. Ostafe. Iterations of rational functions: Some algebraic and arithmetic aspects. *Finite Fields and Their Applications. Character Sums and Polynomials*, De Gruyter, (in press).

[11] A. Ostafe and I. E. Shparlinski. On the length of critical orbits of stable quadratic polynomials. *Proc. Amer. Math. Soc.*, 138: 2653–2656, 2010.

[12] N. Snyder. An alternate proof of Mason’s theorem. *Elemente Math.*, 55: 93–94, 2000.

[13] L. Stickelberger. Über eine neue Eigenschaft der Diskriminanten algebraischer Zahlkörper. *Verh. 1 Internat. Math. Kongresses, 1897*, Leipzig, 182–193, 1898.

Domingo Gómez-Pérez University of Cantabria
domingo.gomez@unican.es

Alina Ostafe Macquarie University
alina.ostafe@mq.edu.au

Igor E. Shparlinski Macquarie University
igor.shparlinski@mq.edu.au

Divisors of $\binom{n}{2}$ and prime powers

Luis Hernández Encinas, Agustín Martín Muñoz and Jaime Muñoz Masqué

As is well known, there exist algorithms for detecting perfect powers (e.g., see [3]) and also for detecting prime powers (e.g., see [2, Algorithm 1.7.5]). Below a characterization of non-prime-powers is presented, namely,

Theorem 1. *A positive integer m is not a prime power if and only if there exists an integer n satisfying the following two conditions:*

(i) $1 < n < m$,

(ii) $\frac{1}{2}n(n-1) \equiv 0 \pmod{m}$.

The previous result suggests to attach to each positive integer m, the set $\mathcal{S}(m)$ of all positive integers n satisfying the conditions (i) and (ii) in Theorem 1. Obviously, $\mathcal{S}(m)$ is a finite subset of $\mathbb{N}$, which is empty if and only if m is a prime power. If $\mathcal{S}(m)$ is not empty, then we denote by $L(m)$ the least element of $\mathcal{S}(m)$.

Figure 1 shows the graph of the function $m \mapsto L(m)$, $m \leq 500$. The gaps (the blank vertical straight lines) correspond to the prime power values of m.

Moreover, in Figure 2 several regularities of this graph, for $m \leq 10000$ can be observed; some of them can suitably be justified.

For example, it is observed that for many values of m the value of $L(m)$ is very close to $m/2$. In fact, if m is twice a prime power, say $m = 2p^e$, then we have either $L(m) = m/2$, or $L(m) = (m+2)/2$. Similarly, there are many values of m for which $L(m)$ is very close to $m/3$. This corresponds to the numbers $m = 3p^e$, $p \geq 5$ (see below).

Let r be the number of distinct prime factors of a positive integer m. The set $\mathcal{S}(m)$ enjoys the following properties:

(i) If m is odd, then $\#\mathcal{S}(m) = 2^r - 2$. If $n \in \mathcal{S}(m)$, then $m - n + 1 \in \mathcal{S}(m)$.

(ii) If m is even, then $\#\mathcal{S}(m) = 2^{r-1} - 1$.

(iii) If $p \geq 3$ is a prime number, then for every $e \in \mathbb{Z}^+$, either $L(2p^e) = p^e$, or $L(2p^e) = 1 + p^e$.

(iv) If $p \geq 5$ is a prime number, then for every $e \in \mathbb{Z}^+$, either $L(3p^e) = p^e$, or $L(3p^e) = 1 + p^e$.

(v) If m is odd, then $m - L(m) + 1$ is the greatest element in $\mathcal{S}(m)$.

The next goal is to analyze how the knowledge of $\mathcal{S}(m)$ can help one to factor m. If m is even, then by dividing finitely many times m by 2, we obtain $m = 2^{e_1}m'$, where m' is odd, and this task can be performed in polynomial time.

Let m be an odd positive integer. Two elements $n, n' \in \mathcal{S}(m)$ are said to be *complementary* if there exists a sequence $i_1 < \ldots < i_s$, such that, $n = n_{i_1,\ldots,i_s}$ and $n' = n_{j_1,\ldots,j_{r-s}}$. In other words,

$$\begin{aligned} n &= a_{i_1,\ldots,i_s} p_{i_1}^{e_{i_1}} \cdots p_{i_s}^{e_{i_s}} = 1 + b_{j_1,\ldots,j_{r-s}} p_{j_1}^{e_{j_1}} \cdots p_{j_{r-s}}^{e_{j_{r-s}}}, \\ a_{i_1,\ldots,i_s} &< p_{j_1}^{e_{j_1}} \cdots p_{j_{r-s}}^{e_{j_{r-s}}}, \end{aligned} \tag{1}$$

This work has been partially supported by Ministerio de Ciencia e Innovación (Spain) under the grant TIN2011-22668.

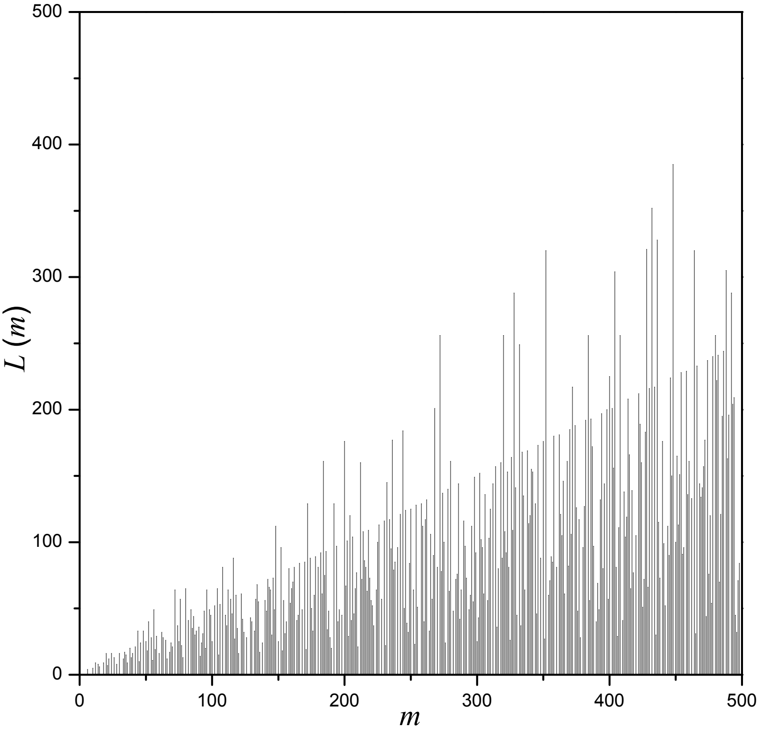


Figure 1: Plot of each value of $L(m)$ with vertical straight lines and $m \leq 500$.

$$\begin{aligned} &n' = a_{j_1,\dots,j_{r-s}} p_{j_1}^{e_{j_1}} \cdots p_{j_{r-s}}^{e_{j_{r-s}}} = 1 + b_{i_1,\dots,i_s} p_{i_1}^{e_{i_1}} \cdots p_{i_s}^{e_{i_s}}, \\ &a_{j_1,\dots,j_{r-s}} < p_{i_1}^{e_{i_1}} \cdots p_{i_s}^{e_{i_s}}. \end{aligned} \tag{2}$$

Bearing this definition in mind, we have

Theorem 2. *Let m be an odd positive integer. If $n, n' \in \mathcal{S}(m)$ are two complementary elements, then*

(i) $\gcd(\gcd(m, n'(n-1)), \gcd(m, n(n'-1))) = 1$,

(ii) $\gcd(m, n'(n-1)) \cdot \gcd(m, n(n'-1)) = m.$

Hence, once a complementary pair is known, a partial factorization of m can be obtained in $O((\log \mu)^3)$ operations, where

$$\mu = \min\{\max\{m, n'(n-1)\}, \max\{m, n(n'-1)\}\}.$$

Conversely, if $n, n' \in \mathcal{S}(m)$ are two elements for which property (i) *above holds true, then n and n' are complementary.*

Example 3. *If $m = 4725 = 3^3 \cdot 5^2 \cdot 7$, then*

$$\mathcal{S}(m) = \{n_1 = 351, n_2 = 1351, n_3 = 1701, n_4 = 3025, n_5 = 3375, n_6 = 4375\},$$

and factoring,

$$\begin{array}{ll} \gcd(m, n_2(n_1-1)) = 5^2 \cdot 7, & \gcd(m, n_1(n_2-1)) = 3^3 \cdot 5^2, \\ \gcd(m, n_3(n_1-1)) = 3^3 \cdot 5^2 \cdot 7, & \gcd(m, n_1(n_3-1)) = 3^3 \cdot 5^2, \\ \gcd(m, n_4(n_1-1)) = 5^2 \cdot 7, & \gcd(m, n_1(n_4-1)) = 3^2 \cdot 5^2 \cdot 7, \\ \gcd(m, n_5(n_1-1)) = 3^3 \cdot 5^2 \cdot 7, & \gcd(m, n_1(n_5-1)) = 3^3 \cdot 7, \\ \gcd(m, n_6(n_1-1)) = 5^2 \cdot 7, & \gcd(m, n_1(n_6-1)) = 3^3. \end{array}$$

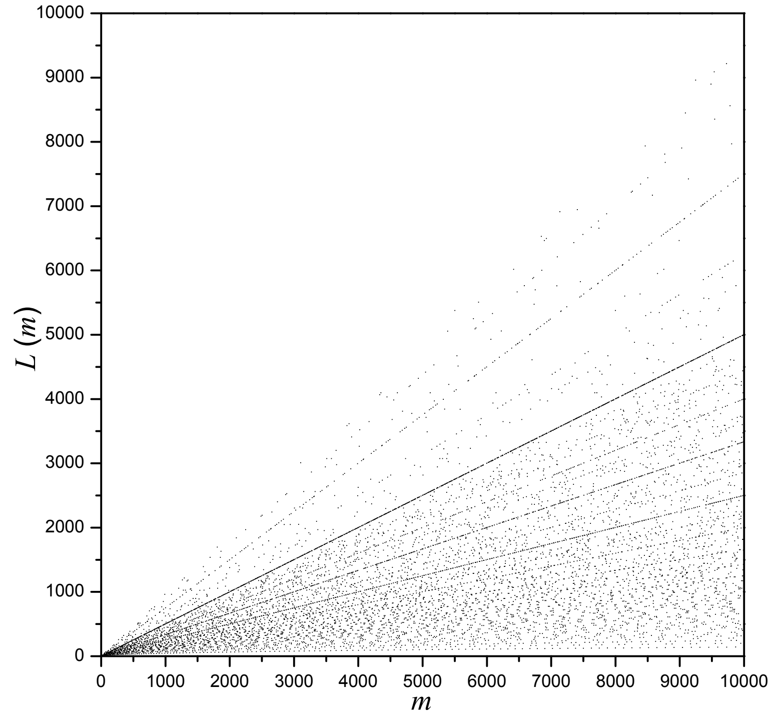


Figure 2: Plot of each value of $L(m)$ without vertical straight lines and $m \leq 10000$.

Consequently, n_1 and n_6 are complementary.

In order to factor m from $\mathcal{S}(m)$, Theorem 2 assumes that two complementary elements are known. The following elementary characterization of complementary pairs, can be useful.

Proposition 4. *Let m be an odd positive integer. Two elements $n, n' \in \mathcal{S}(m)$ are complementary if and only if $n + n' = m + 1$.*

Corollary 5. *Let m, m' be two positive integers which are not prime powers and assume that m is odd. If $\mathcal{S}(m) = \mathcal{S}(m')$, then $m = m'$.*

Example 6. *As a second example let us now consider the prime number*

$$\begin{aligned} p_{97} =& 3002073757\,4287773822\,7385792238\,5512797763\,7927232664 \\ & 1765602502\,1527116989\,7799529501\,8255653754\,1850817, \end{aligned}$$

obtained as a factor in the factorization of the number $(2^{488}+1)/257$ ([1]). Then $p = 2 \cdot 2269 \cdot p_{97} + 1$, and $q = 2 \cdot 349 \cdot p + 1$, are prime, and for $m = pq$ we obtain

$$\begin{aligned} n =& 9509140676\,4258306490\,0961954174\,6249039223\,9597819717\,3631948300 \\ & 0287659715\,9357556885\,7405221402\,0713467267\,807. \end{aligned}$$

The computation is simple as $n(n-1)/(2m) = 349$, and hence, the equation $n(n-1)/2 = km$ is proved to have an integer solution by simply letting $k = 1, \ldots, 349$.

References

[1] D.J. Bernstein and A.K. Lenstra, A general number field sieve implementation, The development of the number field sieve, A.K. Lenstra and H.W. Lenstra, Jr. (eds.), Lecture Notes in Math. 1554 (1993), 103–126.

[2] H. Cohen, A course in computational algebraic number theory, Graduate Texts in Mathematics, 138, Second Corrected Printing, Springer-Verlag, Berlin, 1995.

[3] D. Bernstein, H.W. Lenstra, J. Pila, Detecting perfect powers by factoring into coprimes, Math. Comp. 76 (2007), no. 257, 385–388.

L. Hernández Encinas Information Security Institute (ISI), CSIC
luis@iec.csic.es
A. Martín Muñoz Information Security Institute (ISI), CSIC
agustin@iec.csic.es
J. Muñoz Masqué Information Security Institute (ISI), CSIC
jaime@iec.csic.es

Multicollisions against tree- and graph-based hash functions

Kimmo Halunen

Introduction

Hash functions play an important role in modern cryptographic protocols. Many of the most widely used hash functions are becoming insecure for the needs of the society. Thus there is a need for more secure hash functions and a competition by the National Institute for Standards in Technology (NIST) to find a new secure hash function standard (SHA-3) is ending this year [6].

Cryptographic hash functions need to possess security properties to be applicable in security protocols. Most commonly required properties are preimage resistance, second preimage resistance and collision resistance. There are also other notions such as indistinguishability from a random oracle and more specific notions of the three properties mentioned earlier [12, 15]. The most common strategy for building hash functions has been the Merkle-Damgrard paradigm, where a compression function with fixed input and output length is iterated over the message to achieve a hash function for arbitrary length messages [13, 2].

One fairly powerful attack against iterated hash functions was discovered by Joux [5]. With Joux's method, one can construct multicollisions, i.e. sets of messages with the same hash value, for iterated hash functions much more efficiently than was previously expected. Furthermore, these multicollisions can be used against constructions that were considered fairly secure before Joux's attack [5]. Multicollisions have also been utilised in further attacks against iterated hash functions [7].

Several methods have been proposed to overcome the weakness that Joux's method utilises, e.g [1, 11]. Some of these have been adopted in the SHA-3 competition candidates and some have been found susceptible to similar weaknesses as the original iterated hash functions. One of the ideas to overcome Joux's attack was the introduction of generalised iterated hash functions in [14, 4] and tree-based hash functions in [14]. However, there are multicollision attacks also against most of these variants already displayd in [14, 4]. These attacks have been further improved and generalised in [8, 3, 9]. In this paper we give a further generalisation to these hash functions and show that there is a multicollision attack against even this very general class of hash functions.

Multicollisions and generalisations of iterated hash functions

A multicollision for a hash function h is a set $\{m_1, m_2, \ldots, m_k\}$ of distinct messages such that $h(m_i) = h(m_j)$ for all $i, j \in \{1, \ldots, k\}$. A multicollision with k elements is called a k-collision.

Joux's method for finding a 2^k-collision for an iterated hash function is the following [5]. Let f be the compression function used by h and denote by h_0 the initial value of the hash function. Now the attacker may use the birthday attack to find two values x_1 and y_1 for which $f(h_0, x_1) = f(h_0, y_1) := h_1$. By applying another birthday attack the attacker obtains x_2 and y_2 with $f(h_1, x_2) = f(h_1, y_2) := h_2$. After only k birthday attacks the attacker has k pairs x_i, y_i out of which she can form altogether 2^k different messages that all have the same hash value (namely h_k).

The generalisations presented in [14] and [4] give rise to the class of generalised iterated hash functions. This construction allows the message blocks to be used several times and in permuted order. In [14] the authors show that when each message block is used in the computation of the hash value at most twice, there is an efficient multicollision attack against the hash function. Examples of these types of hash functions are the Hash Twice construction and the Zipper hash [10].

Hoch and Shamir generalise the previous results in [4]. They show that even when the message blocks can be used $q \in \mathbb{N}$ times, there exists a multicollision attack against the hash function that is polynomial in the length of the hash function and the size of the collision. However, their result is triple exponential in the parameter q and is thus very impractical.

The results of [4] have been improved and slightly corrected in several papers [8, 3, 9]. These improvements show that the triple exponential complexity of the multicollision method in [4] can be made much more efficient. It is quite possible that a completely polynomial time method for finding multicollisions against generalised iterated hash functions can be formulated by applying more sophisticated analysis.

Graph-based hash functions

In [14] the authors propose a very general class of hash functions. The class $\mathcal{D}$ is informally defined in [14] as follows:

Let f be a compression function. A hash function H from $\mathcal{D}$ behaves in the following way:

1. H invokes f a finite number of times

2. The entire output of any intermediate invocation (not the final invocation) is fed into the input of other invocations of f

3. Each bit of the message to be hashed is fed into at least one invocation of f

4. The output of the final invocation of f is the output of the hash function H

In [14] only two subclasses of $\mathcal{D}$ are investigated, namely the generalised iterated hash functions and binary tree-based hash functions. However, in this paper we show that this class of hash functions can be completely defined by extending the work of [14] and [4]. These hash functions are called graph-based hash functions and can be defined with the help of graphs.

In this paper we give a formal definition of the hash functions in the class $\mathcal{D}$ as graph-based hash functions. Furthermore, we show that the results of [14] can be extended from binary tree-based hash functions to t-ary tree-based hash functions. Also these multicollision attacks generalise to graphs that exhibit enough tree-like properties. We also conjecture that these results will generalise to all graph-based hash functions i.e. all hash functions in the class $\mathcal{D}$.

We also discuss some future research problems. For example, there are improvements on the complexity of finding multicollisions for generalised iterated hash functions [9]. For graph-based hash functions, there are no similar improvements at the moment. Improving the complexity of the multicollision attacks against graph-based hash functions is one future research direction.

In addition, the results concerning the generalisations of iterated hash functions bound the multiplicity of each message block linearly. An interesting theoretical research problem could be to see how relaxing this restriction would affect the complexity of the attacks. One example is to allow the multiplicity to grow as a polynomial of the number of message blocks. This would have only theoretical interest as this would not be a practical construction.

References

[1] E. Biham and O. Dunkelman. A framework for iterative hash functions - HAIFA. Cryptology ePrint Archive, Report 2007/278, 2007. `http://eprint.iacr.org`.

[2] I. B. Damgrard. A design principle for hash functions. In G. Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

[3] K. Halunen, J. Kortelainen, and T. Kortelainen. Combinatorial multicollision attacks on generalized iterated hash functions. In C. Boyd and W. Susilo, editors, *Eighth Australasian Information Security Conference (AISC 2010)*, volume 105 of *CRPIT*, pages 86–93, Brisbane, Australia, 2010. ACS.

[4] J. J. Hoch and A. Shamir. Breaking the ICE - finding multicollisions in iterated concatenated and expanded (ICE) hash functions. In M. J. B. Robshaw, editor, *Fast Software Encryption - FSE '06*, volume 4047 of *Lecture Notes in Computer Science*, pages 179–194. Springer, 2006.

[5] A. Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In M. K. Franklin, editor, *Advances in Cryptology - CRYPTO '04*, volume 3152 of *Lecture Notes in Computer Science*, pages 306–316. Springer, 2004.

[6] R. M. Kayser. NIST SHA-3 hash function competition announcement. *Federal Register Notices*, 72(212):62212–62220, 2007.

[7] J. Kelsey and T. Kohno. Herding hash functions and the Nostradamus attack. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT '06*, volume 4004 of *Lecture Notes in Computer Science*, pages 183–200. Springer, 2006.

[8] J. Kortelainen, K. Halunen, and T. Kortelainen. Multicollision attacks and generalized iterated hash functions. *Journal of Mathematical Cryptology*, 4, 2010.

[9] J. Kortelainen, T. Kortelainen, and A. Vesanen. Unavoidable regularities in long words with bounded number of symbol occurrences. In B. Fu and D.-Z. Du, editors, *Computing and Combinatorics*, volume 6842 of *Lecture Notes in Computer Science*, pages 519–530. Springer Berlin / Heidelberg, 2011.

[10] M. Liskov. Constructing an ideal hash function from weak ideal compression functions. In E. Biham and A. Youssef, editors, *Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 358–375. Springer Berlin / Heidelberg, 2007.

[11] S. Lucks. A failure-friendly design principle for hash functions. In B. K. Roy, editor, *Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security*, volume 3788 of *Lecture Notes in Computer Science*, pages 474–494. Springer, 2005.

[12] U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[13] R. C. Merkle. One way hash functions and DES. In G. Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer-Verlag, 1990.

[14] M. Nandi and D. R. Stinson. Multicollision attacks on generalized hash functions. Cryptology ePrint Archive, Report 2004/330, 2004. `http://eprint.iacr.org/`.

[15] P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In B. Roy and W. Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer Berlin / Heidelberg, 2004.

Kimmo Halunen University of Oulu
khalunen@ee.oulu.fi

Subspace fuzzy-vault

Kyle Marshall, Joachim Rosenthal, Davide Schipani, and Anna-Lena Trautmann

Background

Fuzzy vault is the term used by Juels and Sudan in [3] to describe a cryptographic primitive in which a key κ is hidden by a set of features A in such a way that any witness B which is close enough to A under the set difference metric can decommit κ. Fuzzy vault is a generalization of fuzzy commitment [4].

The motivation for fuzzy vault is largely predicated on an inherent flaw in the processing of biometric data. In early biometric authentication systems, comparison of a biometric was done against a database stored locally on the machine, rather than in some hashed form. Passwords are normally stored in hashed form to prevent an adversary from seeing the password even in the event that the adversary were able to reverse engineer the device on which it is stored. Since biometric data is irreplacable in the sense that once compromised it cannot be changed, storing the data locally in un-hashed form can pose a significant security risk [2]. The reason that biometric data was not stored in hashed form was a result of the comparative methods for analyzing the data. Consider the case when the biometric is a fingerprint. Although individuals have different fingerprints, environmental and technological issues prevent exact duplication of a fingerprint image even by the same individual. Therefore, if the template image was stored in a hashed form, the authentication image would not match perfectly, and therefore be assigned an entirely different hash value. Some of these issues can be resolved using pre-alignment techniques [6].

The fuzzy vault scheme proposed in [3] is as follows and will henceforth be called the JS scheme. Let $A \subset \mathbb{F}_q$ and let $\kappa = (k_0, k_1, \ldots, k_{\ell-1}) \in \mathbb{F}_q^t$ be the secret key. We require that $|A| = t \geq \ell$. Furthermore, choose $r > t$ and select a set $C \subset \mathbb{F}_q$ to consist of $r - t$ points not in A. Construct the polynomial $\kappa(x) = k_0 + k_1 x + \ldots k_{\ell-1} x^{\ell-1}$ and the sets $\mathcal{A}, \mathcal{B} \subset \mathbb{F}_q \times \mathbb{F}_q$ according to

$$
\begin{aligned}
\mathcal{A} &= \{(x, \kappa(x)) \mid x \in A\}, \\
\mathcal{B} &= \{(y, \kappa(y) + \varepsilon_y) \mid y \in C, \varepsilon_y \neq 0\}.
\end{aligned}
$$

Define $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$. The points $\mathcal{A}$ are called the authentic points, and the points $\mathcal{B}$ are called chaff points. Lastly, an appropriate Reed-Solomon decoder `decode` is selected and $\mathcal{V}$ and `decode` are then made public.

If a witness attempts to gain access to the vault, then the witness submits a set $B \subset \mathbb{F}_q$ which is close to A under the set difference metric and then constructs the polynomial f by interpolating the points of $\mathcal{V}$ whose x-coordinates correspond to B. The witness then uses `decode` to correct f to the nearest codeword in the Reed-Solomon code. If this is given by $\kappa(x)$, then the witness recovers the secret key.

A Fuzzy Vault Scheme Using Network Coding

It was shown in [8] that certain reasonable parameters for the fuzzy vault scheme in its original form cause the system to be susceptible to a brute force attack. Choi et al. in [1] speed up the attack by using a fast polynomial reconstruction algorithm. In the JS scheme, the number of keys and thus the complexity of a brute-force attack is determined by the choice of ℓ. Since the number of features must be larger than ℓ, the security, in practice, depends on the number of features than can be extracted from a biometric. Moon et al. consider the problem of improving the security for small degree polynomials in [9].

Recently, much work has been done in the area of error correcting codes in projective space. These codes turn out to be appropriate for error correction in networks under the setting of Kötter and Kschichang, and are referred to as linear network codes [5]. Extending the construction of the fuzzy vault in the JS scheme to arbitrary linear codes is not entirely straightforward, however, linear network codes can be used to create a fuzzy vault in an analogous way.

In this alternative fuzzy vault scheme, we will utilize techniques from linear network coding and restrict our attention to constant dimension codes [5]. A constant dimension linear network code is a subset of the Grassmanian $\mathcal{G}_q(n,k)$, the set of all k-dimensional subspaces of $\mathbb{F}_q^n$. The subspace distance defines a metric on $\mathcal{G}_q(n,k)$ given by

$$d_S(U,V) = \dim(U+V) - \dim(U \cap V),$$

for $U,V \in \mathcal{G}_q(n,k)$. While finding good linear network codes is still an open research problem, there are many candidates now, including the Reed-Solomon and spread code constructions [7, 5].

In this work, we present the construction of the fuzzy vault based on linear network coding as well as algorithms, security analysis, and considerations for implementation. Furthermore, we show that the fuzzy vault scheme based on linear network coding has several advantages over the JS scheme.

References

[1] W. Y. Choi, S. Lee, D. Moon, Y. Chung, and K. Y. Moon. A fast algorithm for polynomial reconstruction of fuzzy fingerprint vault. *IEICE Electronics Express*, 5(18):725–731, 2008.

[2] T. C. Clancy. Secure smartcard-based fingerprint authentication. In *ACM Workshop on Biometrics: Methods and Applications*, pages 45–52, 2003.

[3] A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, Feb. 2006.

[4] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, CCS ’99, pages 28–36, New York, NY, USA, 1999. ACM.

[5] R. Koetter. Coding for errors and erasures in random network coding. In *in Proc. IEEE Int. Symp. Information Theory*, 2007.

[6] P. Li, X. Yang, K. Cao, X. Tao, R. Wang, and J. Tian. An alignment-free fingerprint cryptosystem based on fuzzy vault scheme. *J. Netw. Comput. Appl.*, 33(3):207–220, May 2010.

[7] F. Manganiello, E. Gorla, and J. Rosenthal. Spread codes and spread decoding in network coding. *CoRR*, abs/0805.0507, 2008.

[8] P. Mihailescu, A. Munk, and B. Tams. The fuzzy vault for fingerprints is vulnerable to brute force attack. In *BIOSIG*, pages 43–54, 2009.

[9] D. Moon, W. yong Choi, and K. Moon. Fuzzy fingerprint vault using multiple polynomials.

K. Marshall University of Zurich
kyle.marshall@math.uzh.ch

J. Rosenthal University of Zurich
rosenthal@math.uzh.ch

D. Schipani University of Zurich
davide.schipani@math.uzh.ch

A.-L. Trautmann University of Zurich
anna-lena.trautmann@math.uzh.ch

The TriTon Transformation

Daniel Smith-Tone

Abstract

Many new systems have been proposed which hide an easily invertible multivariate quadratic map in a larger structure by adding more variables and introducing some mixing of a random component to the structured system. While many systems which have been formed by attempting to hide the hidden structure of equations have been broken by observing symmetric properties of the differential of the public key, the dichotomy between the roles of the different types of variables, or even the different types of monomials in the systems, have given rise to differential invariant attacks which distinguish between subspaces corresponding to one type of variable or the other. In this monologue, we take a general approach, and describe a basic construction, TriTon, of which several of the above types of systems are special cases. We analyse this system, and conclude that such constructions are weak with naive choices of parameters.

Introduction

Since 1994, when Peter Shor discovered the key to factoring large composite integers and computing discrete logarithms in polynomial time on a quantum computer, see [21], there has been an ongoing challenge to develop a secure and practical public key replacement for RSA and Diffie-Hellman. This quest to find quantum-resistant mechanisms to replace the current public key infrastructure is wraught with difficulties. In addition to the challenges of designing asymmetric schemes which are immune to classical attack, the task of the post-quantum cryptographer is to create cryptographic tools which are invulerable in a computational model, the understanding of which is constantly evolving.

As a result of such difficulties, the main approach is to design public key cryptosystems in the classical model of computing which do not admit efficient analysis by known quantum techniques. This process often results in cryptosystems which suffer from massive public keys. In light of Grover's search algorithm, see [14], and the apparent trade-offs among performance, key length, and security which are ubiquitous in the literature, it is entirely possible that we may have no other option in this matter. What we can do, however, is construct schemes which are extremely fast.

Speed is one of the motivating factors for the development of a secure Multivariate Public Key Cryptosystem (MPKC). In addition to its other virtues— such as extreme parametrizability, the ease of adaptability to low power devices, the NP-completeness of the fundamental problem of inverting a system of multivariate equations, and the fact that empirically this problem seems difficult in the average case— multivariate systems, and in particular the "big field" schemes, are extremely efficient, often having speeds dozens of times faster than RSA, [4, 3, 27].

The big question about many purportedly quantum-resistant schemes is whether we can be assured of the security of the system while retaining the desired performance. Many schemes from Multivariate Public Key Cryptography, such as C^*, SFLASH, PMI, ℓIC-, Oil and Vinegar, and the various Square variants, have been broken by uncovering some of the structure inherent to the public key. See [5, 1, 2, 9, 20, 24]. Although there are some general theoretical results about the security of such cryptosystems, see [22, 23], the resistance of these systems against structural attack is not well understood.

In this paper, we analyze an approach to the construction of schemes which involve variables of multiple types. We call such schemes "TriTon," because they contain three colors, or flavors, of monomials— the structure monomials, the obfuscation monomials, and the mixing monomials. We endeavor to reveal some fundamental structural weaknesses of such schemes to further the development of security theory; in particular, we break some instances with naive parameters.

The paper is organized as follows. In the next section we present the TriTon transformation of a multivariate cryptosystem and describe why such a modification might seem beneficial. In the subsequent section, we express several well-known schemes as TriTon transformations of more basic systems. The following sections describes an attack against certain TriTon schemes with poorly chosen parameters. Finally, we draw conclusions about the trustworthiness of systems derived from such a design philosophy.

TriTon Construction

Let q be a prime power, and let $\mathbb{F}_q$ be a finite field with q elements. Given an effectively invertible quadratic function, $f:\mathbb{F}_q^n\to\mathbb{F}_q^m$, a quadratic function, $g:\mathbb{F}_q^l\to\mathbb{F}_q^m$, and $A:\mathbb{F}_q^{n+l}\to\mathbb{F}_q^m$ bilinear, the TriTon construction produces the function $\tilde{f}:\mathbb{F}_q^{n+l}\to\mathbb{F}_q^m$ as follows:

$$\tilde{f}(x,y)=f(x)+g(y)+A(x,y),$$

where $x\in\mathbb{F}_q^n$ and $y\in\mathbb{F}_q^l$.

To complete the scheme, we compose two affine transformations, $T:\mathbb{F}_q^m\to\mathbb{F}_q^m$ and $U:\mathbb{F}_q^{n+l}\to\mathbb{F}_q^{n+l}$, to produce:

$$P(x)=T\circ\tilde{f}\circ U(x),$$

where $x\in\mathbb{F}_q^{n+l}$.

This construction has a great deal of algebraic structure, as can be seen by determining its differential. The discrete differential of an univariate function, f, is the bivariate function $Df(a,x)=f(a+x)-f(x)-f(a)+f(0)$. Since we are only interested in encryption functions which are quadratic, the differential will always be bilinear, and therefore each coordinate of the differential is a bilinear form. The differential of each coordinate of the core map, $\tilde{f}$, has the following structure:

$$D\tilde{f}_i=\begin{bmatrix}Df_i & A_i\\ A_i^T & Dg_i\end{bmatrix}.$$

The motivating force behind this transformation strategy is to hide any structure present in f without producing any new invariants or rank weaknesses. In addition, the ability to make A, or g, or both maps random may provide effective means of hiding the structure of f, and potentially enhance the security of the scheme.

While any system of multivariate equations can be defined using two sets of variables and separating the monomials into three categories, it is only reasonable to consider the system a TriTon construction if the system relies on this delegation of monomials into the three categories, structure, obfuscation, and mixing, for the effective inversion or analysis of the system. Several schemes have been proposed over the years which fit this description. In particular, any of the variants of the Oil and Vinegar scheme, see [17, 20], the C^* modification, PMI, see [15] for C^* and [6] for PMI, and any of the Stepwise Triangular Schemes(STS), see for example, [13] with the Trivial Mixing Methodology(TMM).

Well-known TriTon Systems

While any system of multivariate equations can be defined using two sets of variables and separating the monomials into three categories, it is only reasonable to consider the system a TriTon construction if the system relies on this delegation of monomials into the three categories, structure, obfuscation, and mixing, for the effective inversion or analysis of the system. Several schemes have been proposed over the years fitting this description. Here we express a few well-known schemes which fit the above description, and give an example of a scheme which cannot effectively be considered in such a context.

Oil and Vinegar

The prototypical scheme differentiating between two types of variables is Oil and Vinegar, see [17]. In this scheme, the central map is defined in such a way that quadratic monomials in one type of variable, the oil variables, never occur. Thus the structured component is zero, the obfuscation component is comprised of monomials with random coefficients which are quadratic in the vinegar variables, and the mixing component is similarly random. Once the values of the vinegar variables have been fixed, the system is linear in the oil variables and they can be uniquely determined.

The differential of each single core map formula has the following form:

$$Df_i = \begin{bmatrix} 0 & Df_{i1} \\ Df_{i1}^T & Df_{i2} \end{bmatrix}.$$

Clearly, any vector of the form:

$$\begin{bmatrix} * \\ 0 \end{bmatrix},$$

that is, in the oil subspace, is mapped by Df_i to a vector of the form:

$$\begin{bmatrix} 0 \\ * \end{bmatrix},$$

in the vinegar subspace. Therefore, the product of a matrix in the span of the differential coordinate forms with the inverse of another such matrix will leave the oil subspace invariant, a fact which was exploited to break the balanced oil and vinegar scheme, see [20].

One may note that the unbalanced oil and vinegar scheme similarly admits a TriTon structure, as do several other vinegar variants of multivariate schemes. The main distinction between such systems and the balanced oil and vinegar scheme, is that they never have a trivial quadratic component of such a high, detectable dimension.

PMI

The C^* cryptosystem, developed by Matsumoto and Imai in [15], is the prototypical multivariate public key cryptosystem based on the structure of a large extension field. Given a degree $n+l$ extension, k, of our scalar field, the scheme expressed the composition of a hidden monomial map, $f:\mathbb{F}_q^{n+l}\rightarrow\mathbb{F}_q^{n+l}$, of the form $f(x)=x^{q^\theta+1}$, where $gcd(n+l,\theta)=1$, and two affine transformations, $U,T:\mathbb{F}_q^{n+l}\rightarrow\mathbb{F}_q^{n+l}$, as a system of multivariate equations over the base field. The scheme, however, was later broken by Patarin, see [16], by virtue of a bilinear relation in the input and output of f.

The internally perturbed C^* scheme, PMI, see [6], uses the idea of adding a random summand of low dimensional support to the core map. Given the standard parameters of C^*, internal perturbation augments the core map, f, with a summand $g\circ L$, where $g:\mathbb{F}_q^l\rightarrow\mathbb{F}_q^{n+l}$ is a random quadratic map and $L:\mathbb{F}_q^{n+l}\rightarrow\mathbb{F}_q^l$ is a random linear map. Thus the entire encryption map is given by:

$$P(x)=T\circ f\circ U(x)+T\circ g\circ L\circ U(x).$$

The strategy here is to randomize the obfuscation monomials while retaining structure in the majority of the function. Once the randomized component is removed, the structure of the entire remaining map is utilized to find a preimage.

Specifically, the map $y=P(x)$ can be “inverted” by computing all possible outputs, z, of the random quadratic, g, subtracting Tz from $P(x)$, and applying the decryption routine of C^* to the result. If the output, x, of this procedure matches a preimage of z under $g\circ L\circ U$, then $P(x)=y$ and x is legitimately an inverse of y. If none of the q^l values of z share such a preimage with the C^* portion of the map, then y is not in the image of P.

With a change of basis we can express L as:

$$\tilde{L} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}.$$

We then have:

$$P(x) = \tilde{T} \circ \tilde{f} \circ \tilde{U}(x) + \tilde{T} \circ \tilde{g} \circ \tilde{L} \circ \tilde{U}(x),$$

and in this basis the differential of each formula in the central map has the form:

$$D\tilde{f}_i + D(\tilde{g}_i\tilde{L})_i = \begin{bmatrix} D\tilde{f}_{i1} & D\tilde{f}_{i2} \\ D\tilde{f}_{i2}^T & D\tilde{g}_i + D\tilde{f}_{i3} \end{bmatrix}.$$

One may note that for $n+l$ odd, without the g component, each differential coordinate form has corank 1. If g is truely randomly selected, then often when LUx is nonzero, the rank of the differential coordinate form will be smaller. An equivalent observation involving the associated bilinear form of each public equation, along with some additional probabilistic methods resulted in an attack discovering the "noise kernel," effectively removing the obfuscation, see [10]. Notice that for $\begin{bmatrix} x & y \end{bmatrix}^T \in \cap_i ker(D\tilde{g}_i)$ we have for all i:

$$\begin{bmatrix} D\tilde{f}_{i1} & D\tilde{f}_{i2} \\ D\tilde{f}_{i2}^T & D\tilde{g}_i + D\tilde{f}_{i3} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} D\tilde{f}_{i1} & D\tilde{f}_{i2} \\ D\tilde{f}_{i2}^T & D\tilde{f}_{i3} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

pSFLASH - A Non-Example

pSFLASH is another scheme based on the original C^* scheme of Matsumoto and Imai, see [15]. After the discovery of Patarin's linearization attack, see [16], a new modification, the idea of discarding public equations, was suggested, [18]. This method was later shown to be weak in an attack exploiting a multiplicative symmetry exhibited by the differential of the public key by Dubois et al. from [9]. The results of this paper, and the subsequent generalization of the attack to other schemes, see [11], for example, further popularized differential methods in multivariate cryptanalysis and inspired several theoretical veins of inquiry, see [8, 22, 23].

The practical suggestion was proposed by Ding et al. in [7], that using the projection modifier, which is equivalent to making the affine transformation U singular, may prevent the attack using multiplicative symmetry. The resulting scheme is known as pSFLASH. The encryption map is formed as follows:

$$P(x) = T \circ f \circ S(x),$$

where f is a C^* monomial, and both S and T are singular with corank 1 and r, respectively.

The system is inverted by choosing a nonsingular map which agrees with T on the range of T, applying the inverse of this map, inverting f, and finding a preimage of S. Each of these operations is efficient for anyone with the knowledge of T, f, and S.

We may attempt to view this system as a TriTon scheme by choosing a change of basis which maps the image of S to the first $n-1$ basis vectors. The resulting scheme looks like:

$$P(x) = \tilde{T} \circ \tilde{f} \circ \tilde{S},$$

where $\tilde{S}$ is of the form:

$$\tilde{S} = \begin{bmatrix} \tilde{S}_1 & \tilde{S}_2 \\ 0 & 0 \end{bmatrix}.$$

As a result, the input of the hidden monomial map always has zero as the last coordinate, and we can equivalently regard the core map as including a projection onto the first $n-1$ coordinates, in which case the differential of the ith core coordinate formula has the form:

$$\begin{bmatrix} D\tilde{(f)}_{i1} & 0 \\ 0 & 0 \end{bmatrix}.$$

In light of this fact, one may choose such a basis and consider the system as having one fewer variables. This has the effect of allowing a marginally smaller public key, and since an adversary can easily complete this computation there is no reason not to take advantage of this benefit. As a result, however, there is no advantage to considering this scheme as a TriTon construction.

Trivial Mixing Method and Analysis

In the previous section, we witness the strategies of adding a random component for obfuscation and of making the structured component trivial so that it does not interfere with the inversion of the mixing component. In this section, we describe another strategy called the Hidden Pair of Bijections(HPB) scheme which has been proposed recently by Gotaishi et al., see [13], and present a cryptanalysis. The approach there advocated requires the obfuscation component, g, to be invertible, and for the mixing component, A, to be of full rank. The resulting function defines a signature scheme analogous to the oil and vinegar scheme, in that one fixes the values of a set of variables, rendering the mixing component trivial, and inverts the resultant expression. The exposition of the scheme mentions that any form of structured quadratic components f and g could be used; for example, both f and g could be C^* monomials.

Specifically, to sign a message m, one begins by seting $z = H(m)$, a hash of the message. One then flips a coin determining which of x and y to fix to zero, and solves either $z = f(x) + g(0) + A(x,0) = f(x)$, or $z = f(0) + g(y) + A(0,y) = g(y)$.

The claim is that the scheme is secure because for any particular signature an attacker is unaware whether the first n variables, x, are set to zero, or the second n variables, y; therefore, given a large number of signatures, it cannot be known which ones were signed with x set to zero and which were signed with y set to zero. This claim is false.

Consider the collection of all possible signatures, $\mathcal{S}$. $\mathcal{S}$ consist of two components: $\mathcal{S}_1$, the collection of all signatures which were derived from setting $x = 0$, and $\mathcal{S}_2$, the collection of all signatures which were derived from setting $y = 0$. Both $\mathcal{S}_i$ have dimension n, and therefore we are guaranteed that once an adversary intercepts $2n+1$ signatures, the last signature will be in the span of n of the previous signatures, identifying the domain of either f or g. Projecting the entire scheme onto this subspace reduces the encryption map to the composition of two affine maps with f or g. Thus the scheme is no more difficult to invert than f or g, and it is broken.

In the rump session of PQCRYPTO ’11, Gotaishi suggested a modification to repair the scheme [12]. His suggestion was to add a third type of variable and a third quadratic map, h, which is invertible, but which has no mixing with the other types of variables. The problem with this method, which Gotaishi suggested seemed precarious, is that the domain of this third quadratic map is a differential invariant, i.e. the differential of the core map has the form:

$$\begin{bmatrix} Df_i & A_i & 0 \\ A_i^T & Dg_i & 0 \\ 0 & 0 & Dh_i \end{bmatrix}.$$

Therefore, we can attack the scheme by finding the n-dimensional subspace which is simultaneously invariant under all differential coordinate forms, and projecting onto its cosummand, reducing the scheme to the original HPB primitive.

Generalization of the Trivial Mixing Method

The system of the previous section suffers from another fatal flaw. The requirement that the value to which x or y is fixed is zero is very restrictive, so that there are only $2q^n$ possible signatures, while the domain contains q^{2n} elements. Therefore the proportion of used bits is only $\frac{2}{q^n}$, indicating that the scheme is extremely inefficient.

This strategy of fixing the values of some of the inputs of the core map to render the mixing component trivial can still be used while fixing the inefficiency problem and avoiding the above attack by making the mixing component, A, of low rank. Since randomly choosing which affine half-dimensional space on which to project did not enhance the security of the HPB scheme, we can remove this feature and allow the obfuscation component g to take an arbitrary form. Thus the generalized core map takes the form () with A of corank k and the quadratic function g of whichever form optimizes security.

To sign a message, one randomly selects an element $z \in \cap_x ker(A(x,*))$, and, given a hash y, returns $U^{-1}\begin{bmatrix} f^{-1}(T^{-1}y - g(z)) \\ z \end{bmatrix}$. One checks that $T(f(f^{-1}(T^{-1}y-g(z)))+g(z)+A(f^{-1}(T^{-1}y-g(z)),z)) = y$. The TMM schemes in [25] and [13] are special cases in which k is zero.

Each coordinate of the differential of this core map admits the presentation:

$$\begin{bmatrix} Df_i & A_i \\ A_i^T & Dg_i \end{bmatrix}.$$

Now, as before, an adversary can collect a maximal collection of linearly independent signatures, revealing $\cap_x ker(A(x,*))$. Since no signature is contained in the cokernel, we may project onto this kernel to obtain an equivalent map with a smaller domain. In this manner, the the induced map on the differential produces the following bilinear form:

$$\begin{bmatrix} D\tilde{f}_i & 0 \\ 0 & D\tilde{g}_i \end{bmatrix}.$$

Now each of these differential coordinate forms share an n-dimensional invariant subspace and a k-dimensional invariant subspace. Since the n-dimensional subspace, V, corresponds to the input of f, we compose yet another projection with the system and recover a system of equations linearly equivalent to $T \circ f \circ U|_V$. At this point, the inversion of the entire scheme is reduced to an inversion of the hidden map, f, and thus the construction is broken.

Conclusion

The basic idea of the Triton construction is to combine two disparate quadratic systems, mixing the variables together in such a way that the distillation of a single component is difficult. In many instances, however, the division of variables into classes and the delegation of particular monomials into certain required structures has caused a detectable change in the rank, or invariant structure of the differential of the encryption map.

In particular, the trivial mixing methodology seems fundamentally flawed, in that we can effectively develop a distinguisher which can separate the types of variables based on the properties of each class of monomial, regardless of the dimension associated with each type of variable. In comparison to the case of oil and vinegar, which resists the standard cryptanalysis when sufficiently unbalanced, trivial mixing seems particularly weak.

As a result of these facts, there is good reason to remain skeptical about techniques involving the division of variables into classes, or the introduction of intermediate variables, such as in the case of PMI. If rank methods and differential invariant methods continue to prove effective against such schemes, then none of these TriTon transformations of cryptosystems will be trusted.

References

[1] J. Baena, C. Clough, and J. Ding. Square-vinegar signature scheme. *PQCRYPTO 2008, LNCS*, 5299:17–30, 2008.

[2] O. Billet and G. Macario-Rat. Cryptanalysis of the square cryptosystems. *ASIACRYPT 2009, LNCS*, 5912:451–486, 2009.

[3] A. I.-T. Chen, C.-H. O. Chen, M.-S. Chen, C.-M. Cheng, and B.-Y. Yang. Practical-sized instances of multivariate pkcs: Rainbow, tts, and ℓic-derivatives. *Post-Quantum Crypto, LNCS*, 5299:95–106, 2008.

[4] A. I.-T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee, and B.-Y. Yang. Sse implementation of multivariate pkcs on modern x86 cpus. *CHES 2009, LNCS, Springer, IACR*, 5747:33–48, 2009.

[5] C. Clough, J. Baena, J. Ding, B.-Y. Yang, and M.-S. Chen. Square, a New Multivariate Encryption Scheme. In M. Fischlin, editor, *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pages 252–264. Springer, 2009.

[6] J. Ding. A new variant of the matsumoto-imai cryptosystem through perturbation. In *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, pages 305–318, 2004.

[7] J. Ding, B.-Y. Yang, C.-M. Cheng, O. Chen, and V. Dubois. Breaking the Symmetry: a Way to Resist the New Differential Attack. Cryptology ePrint Archive, Report 2007/366, 2007. http://eprint.iacr.org/.

[8] J. Ding, V. Dubois, B.-Y. Yang, O. C.-H. Chen, and C.-M. Cheng. Could sflash be repaired? *Automata, Languages and Programming*, 4450:691–701, 2009.

[9] V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical Cryptanalysis of SFLASH. In A. Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2007.

[10] P.-A. Fouque, L. Granboulan, and J. Stern. Differential cryptanalysis for multivariate schemes. *EUROCRYPT 2005, LNCS*, 3494:341–353, 2005.

[11] P. A. Fouque, G. Macario-Rat, L. Perret, and J. Stern. Total break of the ℓic- signature scheme. *PKC 2008, LNCS*, 4939:1–17, 2008.

[12] M. Gotaishi. Hidden pair of bijection signature (Part II). *Presentation: Rump Session PQCRYPTO 2011*, 2011. http://troll.iis.sinica.edu.tw/pqc11/recent.shtml.

[13] M. Gotaishi and S. Tsujii. Hidden pair of bijection signature scheme. Cryptology ePrint Archive, Report 2011/353, 2011. http://eprint.iacr.org/.

[14] L. K. Grover. A Fast quantum mechanical algorithm for database search. 1996. Proceedings STOC 1996, 212-219.

[15] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature verification and message-encryption. *Eurocrypt '88, Springer*, 330:419–545, 1988.

[16] J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. *Crypto 1995, Springer*, 963:248–261, 1995.

[17] J. Patarin. The oil and vinegar algorithm for signatures. *Presented at the Dagsthul Workshop on Cryptography*, 1997.

[18] J. Patarin, L. Goubin, and N. Courtois. C $^{*}_{-+}$ and HM: Variations around two schemes of T.Matsumoto and H.Imai. *Asiacrypt 1998, Springer*, 1514:35–49, 1998.

[19] N. Sendrier, editor. *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings*, volume 6061 of *Lecture Notes in Computer Science*, 2010. Springer.

[20] A. Shamir and A. Kipnis. Cryptanalysis of the oil & vinegar signature scheme. *CRYPTO 1998. LNCS*, 1462:257–266, 1998.

[21] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Sci. Stat. Comp.*, 26, 1484, 1997.

[22] D. Smith-Tone. Properties of the discrete differential with cryptographic applications. In Sendrier [19], pages 1–12.

[23] D. Smith-Tone. On the differential security of multivariate public key cryptosystems. In Yang [26], pages 130–142.

[24] E. Thomae and C. Wolf. Roots of square: Cryptanalysis of double-layer square and square+. In Yang [26], pages 83–97.

[25] S. Tsujii, M. Gotaishi, K. Tadaki, and R. Fujita. Proposal of a signature scheme based on sts trapdoor. In Sendrier [19], pages 201–217.

[26] B.-Y. Yang, editor. *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, volume 7071 of *Lecture Notes in Computer Science*, 2011. Springer.

[27] B.-Y. Yang, C.-M. Cheng, B.-R. Chen, and J.-M. Chen. Implementing minimized multivariate public-key cryptosystems on low-resource embedded systems. *3rd Security of Pervasive Computing Conference, LNCS*, 3934:73–88, 2006.

D. Smith-Tone University of Louisville & NIST
daniel.smith@nist.gov

Cubic sieve congruence of the Discrete Logarithm Problem, and fractional part sequences

Srinivas Vivek and C. E. Veni Madhavan

The Cubic Sieve is a variant of the Index Calculus Method for the Discrete Logarithm Problem (DLP) in fields of prime order. It was proposed by Coppersmith et. al. in [1]. Working of the cubic sieve method requires a nontrivial solution (in positive integers) to a Diophantine equation called the Cubic Sieve Congruence (CSC, for short) $x^3 \equiv y^2 z \pmod{p}$, where p is a given prime number. A nontrivial solution to CSC must satisfy

$$x^3 \equiv y^2 z \pmod{p}, \quad x^3 \neq y^2 z, \quad 1 \leq x, y, z < p^{\alpha}, \tag{1}$$

where α is a given real number that satisfies $\frac{1}{3} < \alpha \leq \frac{1}{2}$. Henceforth the above equation will be referred to as CSC (1). When x, y, and z are of the order $O(p^{\alpha})$, then the heuristic expected running time of the cubic sieve is $L_p\left[\gamma = \frac{1}{2}, c = \sqrt{2\alpha}\right] = \exp\left((c + o(1))\,(\ln p)^{\gamma}\,(\ln \ln p)^{1-\gamma}\right)$, where $\ln p$ denotes the natural logarithm of p. Hence smaller values of α lead to faster running times. It is important to note that this estimate of the running time of cubic sieve does not take into account the time required for finding a nontrivial solution to CSC. Therefore, an important open problem concerning the cubic sieve method is to develop an efficient algorithm to determine a nontrivial solution to CSC, given p and α. We shall refer to this problem as the *CSC problem*.

The Number Field Sieve is the current best algorithm for DLP in prime fields with the heuristic expected running time of $L_p\left[\frac{1}{3}, \left(\frac{64}{9}\right)^{\frac{1}{3}}\right]$. Hence the cubic sieve method is mostly of theoretical interest to cryptography. Apart from the cryptographic connection, the CSC problem is a challenging problem in computational number theory and is interesting in its own right. Some attempts to solve this problem have been made in [2, 3]. Recently, the parametrization $x \equiv v^2 z \pmod{p}$ and $y \equiv v^3 z \pmod{p}$ was introduced by Maitra et. al. [3]. Hence CSC (1) can be equivalently written as

$$x \equiv v^2 z \pmod{p},\ y \equiv v^3 z \pmod{p},\ x^3 \neq y^2 z,\ 1 \leq x, y, z < p^{\alpha},\ 1 \leq v < p. \tag{2}$$

We refer to the above equation as CSC (2).

In this paper, we make further progress towards finding an efficient algorithm for the CSC problem by showing that we can determine in deterministic polynomial time whether a solution to CSC (2) exists for a given v $(1 \leq v < p)$. If one exists, we show that we can also compute it efficiently. Previously, the only way to determine this was to check all the values of z from 1 to p^{α}. As a consequence, we show in the $\alpha = \frac{1}{2}$ case of CSC (1) that for primes "close" to i^{ε} (integer i, real $\varepsilon \in [3, 4]$), a solution to CSC exists and it can be computed deterministically in $\widetilde{O}\left(p^{\frac{1}{3}}\right)$ bit operations, while the previous best is $\widetilde{O}\left(p^{\frac{1}{2}}\right)$. The implicit logarithmic factor hidden in the soft-oh notation $\widetilde{O}$ is $\ln^3 p$. Interestingly, we have empirically observed that about one-third of all the primes are covered by the above class.

We were able to accomplish this by relating the above problem to the *gap problem* of fractional part sequences, where we need to determine the non-negative integers N satisfying the fractional part inequality $\{\theta N\} < \phi$ (θ and ϕ are given real numbers) [4]. The correspondence between the CSC problem and the gap problem is that determining the parameter z in the former problem corresponds to determining N in the latter problem, whereas the parameter θ is either $\frac{v^2 \,(\mathrm{mod}\, p)}{p}$ or $\frac{v^3 \,(\mathrm{mod}\, p)}{p}$. In particular, we apply the previous results on the distribution of the non-negative integers N satisfying the fractional part inequality $\{\theta N\} < \phi$ (rational θ, real ϕ are given) to show how to efficiently determine the least common N satisfying both $\{\theta N\} < \phi$ and $\left\{\hat{\theta} N\right\} < \hat{\phi}$ (both θ and $\hat{\theta}$ are rational), when certain conditions on θ, $\hat{\theta}$, ϕ, $\hat{\phi}$ and N are satisfied.

References

[1] D. Coppersmith, A. M. Odlzyko, and R. Schroeppel. Discrete logarithms in $GF(p)$. *Algorithmica*, 1(1):1–15, 1986.

[2] A. Das and C. E. Veni Madhavan. On the Cubic Sieve Method for Computing Discrete Logarithms over Prime Fields. *International Journal of Computer Mathematics*, 82(12):1481–1495, 2005.

[3] S. Maitra, Y. V. S. Rao, P. Stanica, and S. Gangopadhyay. Nontrivial Solutions to the Cubic Sieve Congruence Problem: $x^3 \equiv y^2 z \bmod p$. *Computación y Sistemas*, 12(3):253–266, 2009.

[4] N. B. Slater. The distribution of the integers N for which $\{\theta N\} < \phi$. In *Proceedings of the Cambridge Philosophical Society*, volume 46, pages 525–534, 1950.

Srinivas Vivek University of Luxembourg
srinivasvivek.venkatesh@uni.lu

C. E. Veni Madhavan Indian Institute of Science
cevm@csa.iisc.ernet.in

Invited Lectures

SCC

Invited Lectures

Approximate common divisors via lattices
Nadia Heninger

In the approximate common divisor problem, one is given several multiples of a number with added error, and asked to find their ”approximate common divisor”. The case of two approximate multiples was formulated by Howgrave-Graham, and is a lovely example of lattice-based cryptanalysis with many applications, particularly to partial key recovery problems for RSA. It turns out that these results fit into a broader context of analogies between cryptanalysis and coding theory. Generalizing these techniques leads us to algorithms and challenges for fully homomorphic encryption, private information retrieval, and several families of error-correcting codes.

N. Heninger University of California San Diego
nadiah@cs.ucsd.edu

Structured linear systems and some of their applications
Éric Schost

Introduction

Exploiting the structure of data is a key idea to develop fast algorithms. In the context of linear algebra, this principle is at the heart of algorithms for *structured matrices*. These algorithms can speed up (for instance) the inversion of a given matrix, whenener this matrix has “almost” the structure of e.g. a Toeplitz, Hankel or Vandermonde matrix. For definiteness, we recall that a Toeplitz (resp. Hankel) matrix is invariant along diagonals (resp. anti-diagonals); the $m \times n$ Vandermonde matrix associated to $\mathsf{x} = (x_1,\dots,x_m)$ has entries $[x_i^j]_{i=1,\dots,m,\, j=0,\dots,n-1}$.

In a nutshell, the central idea in this context is to represent structured matrices in a compact manner, by means of their *generators* with respect to suitable *displacement operators*, and operate on this compact data structure.

In this talk, we will focus on the operators for Toeplitz, Hankel and Vandermonde matrices. For φ in a field $\mathbb{F}$, it is customary to define the cyclic down-shift matrix of size n by

$$\mathbb{Z}_{n,\varphi} = \begin{bmatrix} 0 & & & \varphi \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbb{F}^{n\times n}.$$

Then, a matrix $\mathsf{A} \in \mathbb{F}^{m\times n}$ will be called *Toeplitz-like* if

$$\mathbb{Z}_{m,\varphi}\mathsf{A} - \mathsf{A}\mathbb{Z}_{n,\psi}$$

has a low rank compared to n (this rank is independent of the choice of φ and ψ, up to a constant). Roughly speaking, this means that shifting A one unit down is “close” (in terms of rank) to shifting it one unit to the left. Similarly, we will say that A is *Vandermonde-like* if

$$\mathbb{D}(\mathsf{x})\mathsf{A} - \mathsf{A}\mathbb{Z}_{n,\psi}$$

has low rank, where $\mathbb{D}(\mathsf{x})$ is the diagonal matrix with entries $x_1,\dots,x_m$. This condition means that multiplying the rows A by respectively $x_1,\dots,x_m$ is close to shifting it one unit to the left.

A pair of matrices (G,H) in $\mathbb{F}^{m\times\alpha} \times \mathbb{F}^{n\times\alpha}$ will be called *generators* for A, with respect to an operator $\mathcal{L}$ as above, if $\mathcal{L}(\mathsf{A}) = \mathsf{G}\mathsf{H}^t$. When α is small, they can thus play the role of a compact data structure to represent and operate on A (note that in the cases above, we can easily reconstruct A from its generators).

In the rest of this abstract, we give an overview of some algorithms for solving such systems, with a focus on two applications: polynomial interpolation (motivated by list decoding algorithms), and algebraic approximation (motivated by polynomial systems arising in point-counting problems).

Solving structured linear systems

A natural question is to understand how displacement methods can help for tasks such as inverting $\mathsf{A} \in \mathbb{F}^{n\times n}$ (assuming it is invertible), or finding a vector in its nullspace — and more generally a solution of the system $\mathsf{A}\mathsf{u} = \mathsf{v}$. For the displacement operators considered in this abstract, numerous algorithms exist for these tasks; they can be classified into two categories:

- iterative algorithms, which typically compute an LU-factorization of A, or of its inverse;
- divide-and-conquer algorithms, which use a structured version of Strassen's matrix inversion algorithm [24] to compute generators of A^{-1}.

Algorithms in the first category can be traced back to [15, 10]; they usually run in time $O(\alpha n^2)$, for matrices of size n and displacement rank α.

Here, we will focus on divide-and-conquer algorithms. Bitmead and Anderson [4] and Morf [18] gave the first such algorithm, for Toeplitz-like systems, under strong non-degeneracy assumptions. Kaltofen [16] then showed how to lift the non-degeneracy assumptions, using randomization and an extension of Morf's and Bitmead and Anderson's inequalities on the displacement rank of submatrices.

The algorithms in these references run in time $O(\alpha^2 \mathscr{M}(n)\log(n))$, where $\mathscr{M}$ is such that degree n polynomials in $\mathbb{F}[x]$ can be multiplied in $\mathscr{M}(n)$ operations in $\mathbb{F}$. Using Fast Fourier Transform, $\mathscr{M}(n)$ can be taken quasi-linear in n: using the results of [23, 8], we can take $\mathscr{M}(n) \in O(n\log(n)\log\log(n))$, so the previous running time becomes $O(\alpha^2 n\log(n)\log\log(n))$.

Similar results of the form $O(\alpha^2 \mathscr{M}(n)\log(n))$ or $O(\alpha^2 \mathscr{M}(n)\log(n)^2)$ were later obtained for Vandermonde and Cauchy displacement operators, either by a direct approach [21] or by using known equivalences between the various displacement operators [20].

In the two sections below, we are interested in "intermediate" situations, where the displacement rank may be more than constant, but still small compared to n. Then, the previous results are satisfactory (quasi-linear) with respect to n, but not to α: when α is very close to n, their running time is close to $O(n^2 \mathscr{M}(n)\log(n))$, whereas fast dense linear algebra techniques take time only $O(n^\omega)$ (we denote by ω a feasible exponent for linear algebra, that is, a real number such that $n \times n$ matrices over k can be multiplied in $O(n^\omega)$ operations in k; one can take $\omega \leq 2.38$ [27]).

It is actually possible to improve on this by reintroducing dense linear algebra techniques into algorithms for structured matrices. This reduces the cost to $O(\alpha^{\omega-1}\mathscr{M}(n)\log(n))$ for Toeplitz-like matrices [6, 5].

List decoding

As a first application, we consider list decoding algorithms for Reed-Solomon codes and folded Reed-Solomon codes.

We first recall the definition of the codes we will consider. Let k,n be integers, with $k \leq n$, and let $\gamma \in \mathbb{F} - \{0\}$ be an element of order at least n. For $i \geq 0$, we write $x_i = \gamma^i$. Given message symbols $(f_0,\dots,f_{k-1}) \in \mathbb{F}^k$, the Reed Solomon code $\mathsf{RS}_\gamma[k,n]$ maps the polynomial $f(x) = \sum_{i=0}^{k-1} f_i x^i$ to the values $(f(x_0),\dots,f(x_{n-1})) \in \mathbb{F}^n$. The sender sends the n values $(f(x_i))$ to the receiver; we will write $(y_0,\dots,y_{n-1})$ for the received message.

When there are few transmission errors (less than half the minimum distance), the Berlekamp-Massey algorithm allows us to recover the message f. In presence of many errors, to go beyond the error-correction bound, one can resort to list decoding techniques: return several (but hopefully few) polynomials, among which should be the original f.

Following Sudan's breakthrough [25], most algorithms for this task proceed in two steps: an *interpolation* phase, where a multivariate polynomial $Q \in \mathbb{F}[x,y]$ is computed from the received data, and a *root-finding* phase, where the message polynomial f is recovered as a "root" of Q. Here, we focus on the question of computing Q.

In Sudan's original algorithm, the question is to find Q such that $Q(x_i,y_i) = 0$ for all i, with suitable degree and weighted degree constraints (that we do not discuss here). The Guruswami-Sudan algorithm [14] imposes that $Q(x_i,y_i) = 0$ with order $s \geq 1$ (that is, the derivatives of Q of order up to $s-1$ should vanish at all points (x_i,y_i)); this is also the case for further extensions known as *folded codes* [13].

There exist a huge literature dedicated to finding the polynomial Q, see for instance [22, 1, 17, 26, 2, 28, 9, 3]. Roughly speaking, two trends can be distinguished, depending on whether one does linear algebra over $\mathbb{F}[x]$, or over the base field $\mathbb{F}$.

In the former approach, the problem is often reduced to finding short vectors in a polynomial lattice, for which one can rely on algorithms from [12]. Looking at the problem over $\mathbb{F}$, one is led to description by means of Vandermonde-like matrices, or generalizations thereof [19], or by means of block-Hankel matrices [28]. The latter description seems to be the most amenable to the techniques described in the previous paragraph: with $\deg(Q,y) = \ell$, the interpolation at n points, with order s, turns into a Hankel-like system of displacement rank ℓ; it can thus can be done in time $O(\ell^{\omega-1}\mathscr{M}(s^2n)\log(sn))$.

Algebraic approximants

Another family of examples originates from solving systems of polynomial equations depending on parameters. Consider the following situation:

- we want to solve equations $f_1(\mathbf{u},\mathbf{x}),\dots,f_n(\mathbf{u},\mathbf{x}) = 0$, where $\mathbf{x} = (x_1,\dots,x_n)$ are our indeterminates and $\mathbf{u} = (u_1,\dots,u_m)$ are parameters
- we know one value $\mathbf{u}^{(0)}$ and a corresponding solution $\mathbf{x}^{(0)}$
- the Jacobian matrix of $\mathbf{f} = (f_1,\dots,f_n)$ with respect to $\mathbf{x}$ has full rank at $(\mathbf{u}^{(0)},\mathbf{x}^{(0)})$.

Let further $\mathbf{u}^{(1)}$ be the parameter value corresponding to the system we actually want to solve. Writing $\mathbf{u}^{(t)} = (1-t)\mathbf{u}^{(0)} + t\mathbf{u}^{(1)}$, we can use Newton iteration to compute one solution $\mathbf{x}^{(t)}$ of the system $\mathbf{f}(\mathbf{u}^{(t)},\mathbf{x}) = 0$ with entries that are power series in t.

To solve the system at $t = 1$ (or at least find some solutions), we can then reconstruct the minimal polynomial P of $x_n^{(t)}$, which belongs to $\mathbb{F}(t)[X]$. This is where structured linear algebra techniques come into play, since the coefficients of such a polynomial are solutions of a Toeplitz-like linear system; the displacement rank of this system is roughly equal to the degree of P in X. Such a computation is sometimes called *algebraic approximation*, since it generalizes Padé approximation.

Once P is known, setting $t = 1$ gives us the values of x_n above $\mathbf{u}^{(1)}$; similar ideas then give us the corresponding values of $x_1,\dots,x_{n-1}$. This idea is explained in detail in [7] for the particular case where $f_i = u_i - \varphi_i(\mathbf{x})$, for some polynomials φ_i.

As an application, let us mention some problems coming from point-counting in cryptology. Schoof's algorithms and its extensions to higher genus [11] require to compute torsion divisors in the Jacobian of the curve we are considering. This amounts to solve various families of polynomial systems, which fall into the category described here. Typically, we are attempting to do division-by-ℓ (of a torsion divisor D, with ℓ a prime) in the Jacobian — which is naturally seen as parametrized by the divisor D.

References

[1] M. Alekhnovich. Linear diophantine equations over polynomials and soft decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 51(7):2257 –2265, 2005.

[2] P. Beelen and K. Brander. Key equations for list decoding of Reed-Solomon codes and how to solve them. *Journal of Symbolic Computation*, 45(7):773 – 786, 2010.

[3] D. J. Bernstein. Simplified high-speed high-distance list decoding for alternant codes. In *PQCrypto'11*, volume 7071 of *Lecture Notes in Computer Science*, pages 200–216. Springer, 2011.

[4] R. R. Bitmead and B. D. O. Anderson. Asymptotically fast solution of Toeplitz and related systems of linear equations. *Linear Algebra Appl.*, 34:103–116, 1980.

[5] A. Bostan. Algorithmes rapides pour les polynômes, séries formelles et matrices. In *Les cours du CIRM*, volume 1, pages 75–262. 2010.

[6] A. Bostan, C.-P. Jeannerod, and É. Schost. Solving Toeplitz- and Vandermonde-like linear systems with large displacement rank. In *ISSAC'07*, pages 33–40. ACM, 2007.

[7] A. Cafure, G. Matera, and A. Waissbein. Efficient inversion of rational maps over finite fields. In *Algorithms in algebraic geometry*, IMA Vol. Math. Appl., pages 55–77. Springer, 2008.

[8] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[9] H. Cohn and N. Heninger. Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding. In *ICS*, pages 298–308. Tsinghua University Press, 2011.

[10] B. Friedlander, M. Morf, T. Kailath, and L. Ljung. New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices. *Linear Algebra and its Applications*, 27 (0):31 – 60, 1979.

[11] P. Gaudry and É. Schost. Genus 2 point counting over prime fields. *J. Symb. Comput.*, 47(4): 368–400, 2012.

[12] P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In *ISSAC'03*, pages 135–142. ACM, 2003.

[13] V. Guruswami and A. Rudra. Error-correction up to the information-theoretic limit. *Communications of the ACM*, 52(3):87–95, 2009.

[14] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(6):1757 – 1767, Sep–1999.

[15] T. Kailath, S. Y. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *J. Math. Anal. Appl.*, 68(2):395–407, 1979.

[16] E. Kaltofen. Asymptotically fast solution of Toeplitz-like singular linear systems. In *ISSAC'94*, pages 297–304. ACM, 1994.

[17] K. Lee and M. E. O'Sullivan. List decoding of Reed-Solomon codes from a Gröbner basis perspective. *Journal of Symbolic Computation*, 43(9):645 – 658, 2008.

[18] M. Morf. Doubling algorithms for Toeplitz and related equations. *IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 954–959, 1980.

[19] V. Olshevsky and M. A. Shokrollahi. A displacement approach to efficient decoding of algebraic-geometric codes. In *STOC'99*, pages 235–244. ACM, 1999.

[20] V. Y. Pan. On computations with dense structured matrices. *Math. Comp.*, 55(191):179–190, 1990.

[21] V. Y. Pan and A. Zheng. Superfast algorithms for Cauchy-like matrix computations and extensions. *Linear Algebra Appl.*, 310:83–108, 2000.

[22] R. Roth and G. Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246 –257, 2000.

[23] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[24] V. Strassen. Gaussian elimination is not optimal. 13:354–356, 1969.

[25] M. Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[26] P. Trifonov. Efficient interpolation in the Guruswami-Sudan algorithm. *IEEE Transactions on Information Theory*, 56(9):4341 –4349, 2010.

[27] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *STOC*, pages 887–898, 2012.

[28] A. Zeh, C. Gentner, and D. Augot. An interpolation procedure for list decoding Reed-Solomon codes based on generalized key equations. *IEEE Transactions on Information Theory*, 57(9): 5946 –5959, sept. 2011.

É. Schost Univeristy of Western Ontario
eschost@uwo.ca

Lattice reduction and cryptanalysis of lattice-based cryptosystems

Damien Stehlé

Lattice-based cryptography relies the apparent hardness of standard algorithmic problems over euclidean lattices [9]. It provides unmatched security assurances resulting from worst-case to average-case reductions [1, 11], seems to enjoy a great efficiency potential as hinted by several primitives having quasi-optimal asymptotic performances [7, 8], and allows to realize fascinating primitives such as homomorphic encryption [5, 2]. This combination of attractive features has made it a vibrant field of research.

The best generic tool currently known for attacking this family of cryptographic primitives is lattice reduction [10]. Lattice reduction is a representation paradigm: it consists in finding a representation (a basis) of a given lattice that provides easier access to intrinsic properties of that lattice.

In this talk, we will survey the state of the art on lattice reduction algorithms, from both theoretical and practical perspectives [3, 6, 4]. We will then describe how lattice reduction may be used to solve standard problems from lattice-based cryptography, such as the Learning With Errors (LWE) and Small Integer Solution (SIS) problems.

References

[1] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC (Symposium on Theory of Computing)*, pages 99–108, ACM, 1996.

[2] Z. Brakerski and V. Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *FOCS (Symposium on Foundations of Comput. Sci.)*, pages 97–106, IEEE, 2011.

[3] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *ASIACRYPT*, volume 7073 of *Lecture Notes in Comput. Sci.*, pages 1–20, Springer, 2011.

[4] D. Cadé, X. Pujol and D. Stehlé. fplll-4.0, a floating-point LLL implementation. Available at `http://perso.ens-lyon.fr/xavier.pujol/fplll`

[5] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC (Symposium on Theory of Computing)*, pages 169–178, ACM, 2009.

[6] G. Hanrot, X. Pujol and D. Stehlé. Analyzing Blockwise Lattice Algorithms Using Dynamical Systems. In *CRYPTO*, volume 6841 of *Lecture Notes in Comput. Sci.*, pages 447–464, Springer, 2011.

[7] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, volume 5912 of *Lecture Notes in Comput. Sci.*, pages 598–616, Springer, 2009.

[8] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Comput. Sci.*, pages 1–23, Springer, 2010.

[9] D. Micciancio and O. Regev. Lattice-based cryptography. Book chapter in *Post-quantum Cryptography* (D. J. Bernstein and J. Buchmann (eds.)), Springer, 2008.

[10] P. Q. Nguyen. Hermite's Constant and Lattice Algorithms. Book chapter in *The LLL Algorithm: Survey and Applications* (P. Q. Nguyen and B. Vallée (Eds.)), Springer, 2009.

[11] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Journal of the ACM*, 56(6), 2009.

D. Stehlé CNRS – Laboratoire LIP (U. Lyon, CNRS, ENS de Lyon, INRIA, UCBL)
damien.stehle@ens-lyon.fr

Contributed Talks

SCC

On the complexity of the Arora-Ge Algorithm against LWE

Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret

Abstract

Arora & Ge [5] recently showed that solving LWE can be reduced to solve a high-degree non-linear system of equations. They used a linearization to solve the systems. We investigate here the possibility of using Gröbner bases to improve Arora & Ge approach.

Introduction

The Learning With Errors (LWE) Problem was introduced by Regev in [27, 26]. It is a generalisation for large primes of the well-known LPN (Learning Parity with Noise) problem. Since its introduction, LWE has become a source of many innovative cryptosystems, such as the oblivious transfer protocol by Peikert et al. [25], a cryptosystem by Akavia et al. [1] that is secure even if almost the entire secret key is leaked, homomorphic encryption [21, 10, 4], etc... Reasons of LWE's success in cryptography include its simplicity as well as convincing theoretical arguments regarding its hardness, i.e. a reduction from (worst-case) assumed hard lattice problems to (average-case) LWE.

The purpose of this paper is to investigate whether algebraic techniques (e.g. [16, 17, 18, 19, 3, 2, 20]) can be used in the context of LWE. This is motivated by a recent result Arora & Ge [5] who showed that solving LWE can be reduced to solve a high-degree non-linear system of equations.

Learning With Errors

We reproduce below the definition of the LWE problem from [27, 26].

Definition 1 (LWE). *Let $n \geq 1$ be the number of variables, q be an odd prime integer, χ be a probability distribution on $\mathbb{Z}_q$ and $\mathbf{s}$ be a secret vector in $\mathbb{Z}_q^n$. We denote by $L_{\mathbf{s},\chi}^{(n)}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ at random, choosing $e \in \mathbb{Z}_q$ according to χ, and returning $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.* LWE *is the problem of finding $\mathbf{s} \in \mathbb{Z}_q^n$ given pairs $\mathbb{Z}_q^n \times \mathbb{Z}_q$ sampled according to $L_{\mathbf{s},\chi}^{(n)}$.*

The modulus q is typically taken to be polynomial in n, and χ is the discrete Gaussian distribution on $\mathbb{Z}_q$ with mean 0 and standard deviation $\sigma = \alpha \cdot q$, for some α. To *discretize* the Gaussian distribution $\mathbb{N}0, \sigma^2$ modulo q, we sample according to $\mathbb{N}0, \sigma^2$ and round to the nearest integer mod q. In what follows, $\chi_{\alpha,q}$ will then denote this discretized distribution.

A typical setting for the standard deviation (std) is $\sigma = n^{\varepsilon}$, with $\varepsilon, 0 \leq \varepsilon \leq 1$. For example, [27] suggests $q \approx n^2$ and $\alpha = 1/(\sqrt{n} \cdot \log^2 n)$. Indeed, as soon as $\varepsilon \geq 1/2$ (worst-case) $\text{GAPSVP} - \tilde{o}(n/\alpha)$ reduces to (average-case) LWE[1]. Thus, any algorithm solving LWE (when $\varepsilon \geq 1/2$) can be used for $\text{GAPSVP} - \tilde{o}(n/\alpha)$. We emphasize that it is widely believed that only exponential algorithm exists for solving $\text{GAPSVP} - \tilde{o}(n/\alpha)$.

Recently, Arora & Ge [5] introduced a variant of LWE with *structured* errors. In this setting, you have given an oracle such that given LWE samples returns polynomials which vanish on the errors.

[1] The reduction is quantum if q is polynomial but can be made [24] classical if q is super polynomial.

They showed that the (discretized) Gaussian intrinsically induced a structure on the errors. This feature can be used to reduce LWE to the problem of solving a non-linear system of multivariate of equations.

The total complexity (time and space) of their approach is $2^{\tilde{O}(n^{2\varepsilon})}$. It is then subexponential when $\varepsilon < 1/2$, but remains exponential when $\varepsilon \geq 1/2$. It is interesting that Arora&Ge reach with a completely different approach the $\varepsilon = 1/2$ hardness limit advised by Regev [27, 26].

Note that an improvement on Arora&Ge could allow to challenge the '*subexponetiality*' of $\text{GAPSVP} - \tilde{O}(n/\alpha)$. Remark that [5] uses linearization to solve the non-linear system. It is then natural to investigate whether more advanced tools, such as Gröbner bases [11, 12, 13], could improve the algorithm of Arora&Ge.

In this note, we will show that Gröbner bases can bring a practical improvement on the complexity of [5]. We also briefly discuss whether Gröbner bases can (or can not) allow to change the complexity class of Arora&Ge. Before that, we need to recall some basic complexity results about Gröbner bases.

Gröbner bases – Complexity Results

Gröbner basis is probably the main tool allowing to solve non-linear system of finite fields. From an algorithmic point of view, Lazard [22] showed that computing the Gröbner basis for a system of polynomials is equivalent to perform a Gaussian elimination on the *Macaulay matrices* [23] $\mathcal{M}^{\text{acaulay}}_{d,m}$ for $d, 1 \leq d \leq D$ for some integer D. Moreover, the most efficient known algorithms such as F_5 [15] reduce Gröbner basis computations to a series of Gaussian eliminations on matrices of increasing sizes.

Definition 2 (Macaulay Matrix [23])**.** *Let* $f_1, \ldots, f_m \in \mathbb{Z}_q[x_1, \ldots, x_n]$. *The* Macaulay matrix $\mathcal{M}^{\text{acaulay}}_{d,m}(f_1, \ldots, f_m)$ *of degree d is defined as follows: list "horizontally" all the degree d monomials from smallest to largest sorted by some fixed admissible monomial ordering. The smallest monomial comes last. Multiply each* f_i *by all monomials* $t_{i,j}$ *of degree* $d - d_i$ *where* $d_i = \deg(f_i)$*. Finally, construct the coefficient matrix for the resulting system:*

$$\mathcal{M}^{\text{acaulay}}_{d,m}(f_1,\ldots,f_m) := \begin{array}{c} \\ (t_{1,1}, f_1) \\ (t_{1,2}, f_1) \\ \vdots \\ (t_{m,1}, f_m) \\ (t_{m,2}, f_m) \\ \vdots \end{array} \begin{array}{c} \text{monomials of degree } \leq d \textit{ sorted for } < \\ \left(\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array} \right) \end{array}$$

Theorem 3 ([22])**.** *Let* $\mathbf{f} = (f_1, \ldots, f_m) \in (\mathbb{Z}_q[x_1, \ldots, x_n])^m$ *and* $<$ *be a monomial ordering. There exists a positive integer D for which Gaussian elimination on all* $\mathcal{M}^{\text{acaulay}}_{d,m} = (f_1, \ldots, f_m)$ *matrices for* $d, 1 \leq d \leq D$ *computes a Gröbner basis of* $\langle f_1, \ldots, f_m \rangle$ *w.r.t. to* $<$*. The degree D will be called* degree of regularity *of* $f_1, \ldots, f_m$*.*

Consequently, the complexity of computing a Gröbner basis is bounded by the complexity of performing Gaussian elimination on the Macaulay matrix in some degree D. Roughly, the complexity of computing a Gröbner basis with an algorithm based on the degree of regularity (such as – but not limited too – Buchberger's algorithm, F_4, F_5 [15, 11, 12, 14]) is:

$$O\left(\binom{n+D}{D}^{\omega} \right), \tag{1}$$

where $2 \leq \omega < 3$ is the linear algebra constant, and D is the degree of semi-regularity of the system.

In general, computing the degree of regularity of a system is a difficult problem. However, it is known for a specific family of polynomial systems [6, 8, 7, 9].

Definition 4 (Semi-regular Sequence [8])**.** *Let $m > n$, and $f_1, \ldots, f_m \in \mathbb{Z}_q[x_1, \ldots, x_n]$ be homogeneous polynomials of degrees $d_1, \ldots, d_m$ respectively and $\mathcal{I}$ the ideal generated by these polynomials. The system is said to be a* semi-regular sequence *if the Hilbert series [13] of $\mathcal{I}$ w.r.t. the grevlex order is:*

$$H_{\mathcal{I}}(z) = \left[\frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-z)^n}\right]_+, \tag{2}$$

where $[S]_+$ denotes the series obtained by truncating S before the index of its first non-positive coefficient. Thus, the degree of regularity D involved in Theorem 3 *for a semi-regular sequence is:*

$$1 + \deg(H_{\mathcal{I}}).$$

Improving Arora-Ge Approach

We briefly detail below the linearization approach of Arora-Ge. We then discuss whether Gröbner bases can be used in this context.

Basic Arora-Ge Algorithm – A Linerization Approach

The idea of [5] is to generate a non-linear noise-free system of equations from LWE samples. This is due to the following well-known feature of a Gaussian noise:

Lemma 5. *Let $C > 0$ be a constant. It holds that:*

$$\Pr[e \xleftarrow{\$} \chi_{\alpha,q} : |e| > C \cdot \sigma] \leq e^{\mathcal{O}(-C^2)}.$$

As a consequence, elements sampled from a Gaussian distribution only takes values on a (small) subset $[-C\cdot\sigma, \ldots, C\cdot\sigma]$ of $\mathbb{Z}_q$ with high probability. From now on, we set $t = C\cdot\sigma$. We can reinterpret Lemma 5 algebraically by considering the polynomial:

$$P(X) = X\prod_{i=1}^{t}(X+i)(X-i).$$

Clearly P is of degree $2t+1 \in \mathcal{O}(\sigma)$. Thus, if $e \xleftarrow{\$} \chi_{\alpha,q}$, then $P(e) = 0$ with probability at least $1 - e^{\mathcal{O}(-C^2)}$.

For $i \geq 1$, let $(\mathbf{a}_i, \langle \mathbf{a}_i, s\rangle + e_i) = (\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. If $e_i \xleftarrow{\$} \chi_{\alpha,q}$, then

$$P\big(\mathbf{a}_i, \langle \mathbf{a}_i, s\rangle - b_i\big) = 0, \tag{3}$$

with probability at least $1 - e^{\mathcal{O}(-C^2)}$. As a consequence, each sample $(\mathbf{a}_i, \langle \mathbf{a}_i, s\rangle + e_i) = (\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ allows to generate a non-linear equation of degree $2t+1$ in the n components of the secret $\mathbf{s}$.

The idea of Arora & Ge is then to generate sufficiently many equations as in (3) to perform a linearization. However, one has to choose the constant – denoted by C_{AG} – occurring in Lemma 5 sufficiently big so that all errors generated lies with high probability in $[-C_{\text{AG}}\cdot\sigma, \ldots, C_{\text{AG}}\cdot\sigma] \subseteq \mathbb{Z}_q$, i.e. the secret $\mathbf{s}$ is indeed a common solution of the M_{AG} equations constructed as in (3). To this end, we set:

$$p_f = \frac{M_{\text{AG}}}{e^{\mathcal{O}(C_{\text{AG}}^2)}}.$$

This is the probably that the secret $s \in \mathbb{Z}_q^n$ is not solution of the system $\mathcal{S}_{\text{AG}}$ generated from M_{AG} equations as in (3), i.e. the probability of failure of Arora-Ge approach. Let also $D_{AG} = 2C_{\text{AG}}\cdot\sigma + 1$ be the degree of the equations occuring in $\mathcal{S}_{\text{AG}}$. According to [5], taking $C_{\text{AG}} \in \tilde{\mathcal{O}}(\sigma)$ allows to make the probability of failure negligible.

To summarize, Arora-Ge approach reduces to linearize at system of M_{AG} equations of degree $D_{AG} = 2C_{\text{AG}}\cdot\sigma + 1 \in \tilde{\mathcal{O}}(\sigma^2)$. Moreover, correctness of this approach can be proven:

Theorem 6. *[5] Let $D_{\mathrm{AG}} < q$. The system obtained by linearizing $M_{\mathrm{AG}} = O\left(q \cdot \log(q)\binom{n+D_{\mathrm{AG}}}{D_{\mathrm{AG}}}\sigma\right) = n^{O(D_{\mathrm{AG}})} = 2^{\tilde{O}(D_{\mathrm{AG}})}$ equations as in* (3) *has at most one solution with high probability.*

The time complexity of the basic Arora-Ge approach is then

$$\mathcal{C}_{AG}^{\mathrm{plx}} = n^{O(D_{AG})} = 2^{\tilde{O}(\sigma^2)} = 2^{\tilde{O}(n^{2\varepsilon})}.$$

Note also this algorithm also requires $2^{\tilde{O}(n^{2\varepsilon})}$ LWE samples for performing the linearization.

From Linerization to Gröbner Bases

The question we try to address here is whether the complexity $\mathcal{C}_{AG}^{\mathrm{plx}}$ can be improved by using Gröbner bases instead of linearization. The rational is that you can decrease the constant C_{AG} (and so the degree of the equations) to a value smaller than $\tilde{O}(n^{2\cdot\varepsilon})$ by considering less equations (whilst keeping the probability p_f of failure similar in bother approaches). However, the cost of the solving step increase since one has to compute a Gröbner basis. The question is then to find – if any – a tradeoff allowing to improve upon linearization.

To do so, we will consider a number of equations of the form $\sqrt[\theta]{M_{\mathrm{AG}}}$, with $\theta > 1$ ($\theta = 1$ is the basic Arora-Ge). We want to keep the probability of failure similar for the linearization and Gröbner basis approaches. As a consequence, we need to take a constant C_θ such that:

$$p_f = \frac{\sqrt[\theta]{M_{\mathrm{AG}}}}{e^{O(C_\theta^2)}}.$$

An easy calculation leads to $C_\theta \in \tilde{O}\left(\frac{C_{\mathrm{AG}}}{\sqrt{\theta}}\right)$. Thus, decreasing the number of equations from M_{AG} to $\sqrt[\theta]{M_{\mathrm{AG}}}$ allows to divide the constant C_{AG} by a factor $\sqrt{\theta}$. The degree of the equations we are doing to consider is then equal to $2\sigma \cdot C_\theta + 1 \in \tilde{O}\left(\frac{\sigma^2}{\sqrt{\theta}}\right)$.

The question is now to find a good candidate for θ. Typically, if θ is too big then you will greatly decrease the number of equations, but the cost of the solving step will become prohibitive and the total complexity will be worth than for a linearization.

We have considered a θ of the form: $\theta = n^{2\cdot\beta}$, for some $\beta \geq 0$ (note that we get the basic Arora-Ge by taking $\beta = 0$). In this new setting, we get a constant $C_\beta = n^{\varepsilon-\beta}$. We have then to solve a system having $M_\beta = \sqrt[n^{2\cdot\beta}]{M_{\mathrm{AG}}} \in 2^{\tilde{O}(n^{2(\varepsilon-\beta)})}$ equations of degree $D_\beta = \tilde{O}(n^{2\cdot\varepsilon-\beta})$. We denote such system by $\mathcal{S}_{\mathrm{GB}}(\beta)$.

The question is to determine the complexity $\mathcal{C}_{GB-AG}^{\mathrm{plx}}(\beta)$ of solving $\mathcal{S}_{\mathrm{AG}}(\beta)$. This reduces to study its degree of regularity D_{reg}^{β}. Given current algorithms, the specific structure of the system does not allow to solve it faster than random systems. As a consequence, we assume that D_{reg}^{β} is not bigger than the degree of regularity of a semi-regular system of the same size[2], namely:

$$D_{reg}^{\beta} \leq 1 + \deg(H_\beta),$$

where:

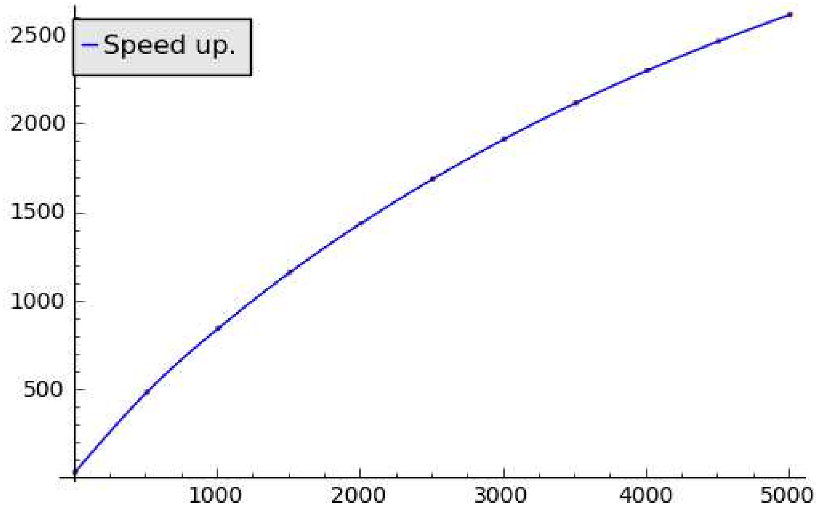
$$H_\beta(z) = \left[\frac{(1-z^{D_\beta})^{M_\beta}}{(1-z)^n}\right]_+,$$

where $[.]_+$ denotes the series obtained by truncating before the index of its first non-positive coefficient.

We present below some experiments performed for $\beta = 1/5$. We have computed explicitly the complexities for both approaches: linearization and Gröbner bases. As suggested in [27],

[2] We have performed few experiments for small parameters. The experiments seem to confirm this hypothesis.

we considered $q \approx n^2$ and $\alpha = 1/(\sqrt{n} \cdot \log^2 n)$ We plotted below the speed-up we obtained, i.e. $\log_2 \left(\frac{C^{\text{plx}}_{GB-AG}(\beta)}{C^{\text{plx}}_{AG}} \right)$ (y-axis) for $n, 0 \leq n \leq 5000$. We can see that Gröbner bases allow to improve the complexity of the basic Arora-Ge when $n \leq 5000$ (x-axis). Note that further experiments are required to confirm this behavior when n tends to infinity[3]



However, the form of the speed-up also tends to suggest that we only improve from a constant C^{plx}_{AG}. change the asymptotical behavior of the Arora&Ge approach. we mention that we are currently considering several forms for the β. In particular, β which is not a constant but a function of n. As a conclusion, we also emphasize that Arora-Ge needs exponential (or subexponetial) number of LWE samples. For most cryptosystems based on LWE, you have access to much less samples, typically polynomially-many. In this situation, you have then not enough samples to perform the linearization and the only option to mount the Arora&Ge approach is to solve the system by using Gröbner bases.

References

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer Verlag, 2009.

[2] M. R. Albrecht and C. Cid. Cold boot key recovery by solving polynomial systems with noise. In J. Lopez and G. Tsudik, editors, *ACNS*, volume 6715 of *Lecture Notes in Computer Science*, pages 57–72, 2011.

[3] M. R. Albrecht and K. G. Paterson. Breaking an identity-based encryption scheme based on dhies. In L. Chen, editor, *IMA Int. Conf.*, volume 7089 of *Lecture Notes in Computer Science*, pages 344–355. Springer Verlag, 2011.

[4] M. R. Albrecht, P. Farshim, J.-C. Faugère, and L. Perret. Polly Cracker, revisited. In *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 179–196, Berlin, Heidelberg, New York, 2011. Springer Verlag. full version available as Cryptology ePrint Archive, Report 2011/289, 2011 `http://eprint.iacr.org/`.

[3] Note that the degree of the equations involved being huge, it becomes rather costly to just expand the Hilbert series for the systems considered.

[5] S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer Verlag, 2011.

[6] M. Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université de Paris VI, 2004.

[7] M. Bardet, J.-C. Faugère, and B. Salvy. Complexity study of Gröbner basis computation. Technical report, INRIA, 2002. `http://www.inria.fr/rrrt/rr-5049.html`.

[8] M. Bardet, J.-C. Faugère, and B. Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *Proc. International Conference on Polynomial System Solving (ICPSS)*, pages 71–75, 2004.

[9] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Proc. of MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry*, 2005.

[10] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In R. Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 97–106. IEEE, 2011.

[11] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.

[12] B. Buchberger, G. E. Collins, R. G. K. Loos, and R. Albrecht. Computer algebra symbolic and algebraic computation. *SIGSAM Bull.*, 16(4):5–5, 1982.

[13] D. A. Cox, J. B. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Springer Verlag, 2005.

[14] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139:61–88, June 1999.

[15] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In T. Mora, editor, *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation ISSAC*, pages 75–83. ACM Press, July 2002. isbn: 1-58113-484-3.

[16] J.-C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (hfe) cryptosystems using gröbner bases. In D. Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60. Springer Verlag, 2003.

[17] J.-C. Faugère and L. Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In S. Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 30–47. Springer Verlag, 2006.

[18] J.-C. Faugère, F. L. dit Vehel, and L. Perret. Cryptanalysis of minrank. In Wagner [28], pages 280–296.

[19] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 279–298. Springer Verlag, 2010.

[20] J.-C. Faugère, L. Perret, C. Petit, and G. Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 27–44. Springer Verlag, 2012.

[21] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. Available at `http://crypto.stanford.edu/craig`.

[22] D. Lazard. Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of the European Computer Algebra Conference on Computer Algebra*, volume 162 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 1983. Springer Verlag.

[23] F. S. Macaulay. On some formula in elimination. *London Mathematical Society*, 1(33):3–27, 1902.

[24] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 333–342. ACM, 2009.

[25] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In Wagner [28], pages 554–571.

[26] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *STOC*, pages 84–93. ACM, 2005.

[27] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[28] D. Wagner, editor. *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, 2008. Springer Verlag.

M. R. Albrecht INRIA, LIP6, CNRS
malb@lip6.fr

C. Cid Royal Holloway, University of London
carlos.cid@rhul.ac.uk

J.-C. Faugère INRIA, LIP6, CNRS
jean-charles.faugere@inria.fr

R. Fitzpatrick Royal Holloway, University of London
robert.fitzpatrick.2010@live.rhul.ac.uk

L. Perret INRIA, LIP6, CNRS
ludovic.perret@lip6.fr

On the complexity of the BKW Algorithm on LWE

Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret

Abstract

In this paper we present a study of the complexity of the Blum-Kalai-Wasserman (BKW) algorithm when applied to the Learning with Errors (LWE) problem, by providing refined estimates for the data and computational effort requirements for solving *concrete* instances of the LWE problem. We apply this refined analysis to suggested parameters for various LWE-based cryptographic schemes from the literature and as a result, provide new upper bounds for the concrete hardness of these LWE-based schemes.

Introduction

LWE (Learning with Errors) is a generalisation for large primes of the well-known LPN (Learning Parity with Noise) problem. It was introduced by Regev in [27] and has provided cryptographers with a remarkably flexible tool for building cryptosystems. For example, Gentry, Peikert and Vaikuntanathan presented in [17] LWE-based constructions of trapdoor functions, digital signature schemes, universally composable oblivious transfers and identity-based encryption. Moreover, in his recent seminal work Gentry [16] resolved one of the longest standing open problems in cryptography with a construction related to LWE: the first fully homomorphic encryption scheme. This was followed by further constructions of homomorphic encryption schemes based on the LWE problem, e.g. [1, 11]. Reasons for the popularity of LWE as cryptographic primitive include its simplicity as well as convincing theoretical arguments regarding its hardness, namely, a (quantum) reduction from worst-case lattice problems, such as the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP), to average-case LWE.

We reproduce the definition of the LWE problem from [27].

Definition 1 (LWE)**.** *Let $n \geq 1$ be the number of variables, q be an odd prime integer, χ be a probability distribution on $\mathbb{Z}_q$ and $\mathbf{s}$ be a secret vector in $\mathbb{Z}_q^n$. We denote by $L_{\mathbf{s},\chi}^{(n)}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ at random, choosing $e \in \mathbb{Z}_q$ according to χ, and returning $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.* LWE *is the problem of finding $\mathbf{s} \in \mathbb{Z}_q^n$ given pairs $\mathbb{Z}_q^n \times \mathbb{Z}_q$ sampled according to $L_{\mathbf{s},\chi}^{(n)}$.*

The modulus q is typically taken to be polynomial in n, and χ is the discrete Gaussian distribution on $\mathbb{Z}_q$ with mean 0 and standard deviation $\sigma = \alpha \cdot q$, for some α. Regev proved [27] that if $\sigma \geq \sqrt{n}$, then (worst-case) GAPSVP $- \tilde{O}(n/\alpha)$ reduces to (average-case) LWE. This reduction is quantum when $q \in \text{poly}(n)$; it can however be made classical [25] if the modulus is super-polynomial, i.e., $q \in 2^{\tilde{O}(n)}$.

MOTIVATION. While there is a reduction of LWE to (assumed) hard lattice problems [27], little is known about the *concrete* hardness of particular LWE instances. That is, given particular values for the prime q and the security parameter n, what is the number of bit operations required to recover the secret using currently known algorithms? As a consequence of this gap, most proposals based on LWE do not provide concrete choices for parameters and restrict themselves to asymptotic statements about security, which can be considered unsatisfactorily vague for practical purposes. In fact we see this lack of precision as one of the several obstacles to the consideration of LWE-based schemes for real-world deployment.

RELATED WORK. Since LWE can be reduced to hard lattice problems, advances in and concrete estimates for lattice algorithms typically carry over to LWE. Indeed, the expected complexity of lattice algorithms is often exclusively considered when parameters for LWE-based schemes are discussed. However, while the effort on improving lattices algorithms is intense [28, 12, 23, 15, 24, 18, 22, 26], direct algorithms for tackling the LWE problem remain rarely investigated from an algorithmic point of view. For example, the main subject of this paper – the BKW algorithm – specifically applied to the LWE problem has so far received no treatment in the literature[1]. Furthermore, it is only recently that an alternative to BKW has been proposed for LWE: Arora and Ge [2] proposed a new algebraic technique for solving LWE, with total complexity (time and space) of $2^{\tilde{O}(\sigma^2)}$ (it is thus subexponential when $\sigma < \sqrt{n}$, remaining exponential when $\sigma \geq \sqrt{n}$). It is worth noting that Arora and Ge achieve the $\sqrt{n}$ hardness-threshold found by Regev [27], but with a constructive approach. However, currently the main relevance of Arora-Ge's algorithm is asymptotic; it is an open question whether one can improve its practical efficiency.

For comparison, the situation is much different in code-based cryptography. That is, improvements on the Information Set Decoding (ISD) algorithm – the classical technique for decoding random linear codes – are continuously reported, e.g. [13, 6, 7, 20, 5], allowing to rather easily determine concrete parameters for code-based schemes. From a more general perspective, we emphasise that improving the constants of exponential algorithms solving hard computational problems is emerging as a new important research area in computer science. For computational problems related to cryptography, we mention recent results on solving knapsack [4, 19], solving set of non-linear equations [8, 9], as well as lattices problems.

CONTRIBUTION. We present a detailed study of a dedicated version of the Blum, Kalai and Wasserman [10] (BKW) algorithm for LWE with discrete Gaussian noise. To our knowledge, this is the first time that such detailed description appears in the literature. Given an instance of the LWE problem as described in Definiton 1, let a and b be two parameters such that $a = \lceil n/b \rceil$. BKW can then be viewed as consisting of three stages: sample reduction, hypothesis testing to recover a subset of the secret, and combining of candidate solutions. On a high level, the first stage of the BKW algorithm can be described as a form of Gaussian elimination which, instead of treating each column independently, considers 'blocks' of b columns per iteration. Following this reduction, the second stage performs hypothesis tests to recover components of the secret vector $\mathbf{s}$. The third stage combines these components to recover the full secret $\mathbf{s}$.

By studying in detail each of these stages, we take the first steps to 'de-asymptotic-ify' our understanding of the hardness of LWE under the BKW algorithm. That is, by investigating the exact complexity of the algorithm, we provide concrete values for the expected number of bit operations for solving instances of the LWE problem. The BKW algorithm is known to have complexity $2^{\mathcal{O}(n)}$ when applied to LWE instances with a prime modulus polynomial in n [27]; in this paper we provide both the leading constant of the exponent in $2^{\mathcal{O}(n)}$ and concrete costs of BKW when applied to LWE. More precisely, we first show the following theorem.

Theorem 2 (informal). *Let $(\mathbf{a}_i, c_i)$ be LWE samples following $L_{\mathbf{s},\chi}^{(n)}$, let $0 < b \leq n$, $r \geq 0$ and $d \leq b$ be parameters, and define $a = \lceil n/b \rceil$. The expected cost of the BKW algorithm to recover* $\mathbf{s}$ *is upper-bounded by*

$$\left\lceil \frac{n}{d} \right\rceil \cdot \left(\frac{q^d}{(q^d-1)} \cdot \left(\frac{a}{2} \cdot (n+3) \cdot (a \cdot q^b + m) + m \cdot q^d \right) \right) + 4n(r+1)^{(\lceil n/d \rceil)}$$

arithmetic operations in $\mathbb{Z}_q$ and $\frac{\lceil n/d \rceil \cdot q^d}{(q^d-1)} \left(a \cdot q^b + m \right) + 2n$ calls to the LWE oracle, where m is a value depending on $L_{\mathbf{s},\chi}^{(n)}$, a, r and d with typically $m \ll 2^n$.

[1] However, a detailed study of the algorithm to the LPN-case was provided [14], which inspired this work.

We then discuss how to select the parameters b,r,d and how to compute m. Finally, we apply our results to various parameter choices from the literature [27, 21, 1].

The BKW Algorithm

The BKW algorithm was proposed by Blum, Kalai and Wasserman [10] as a method for solving the LPN problem, with sub-exponential complexity, requiring $2^{\mathcal{O}(n/\log n)}$ samples and time. The algorithm can be adapted for tackling the LWE problem, with complexity $2^{\mathcal{O}(n)}$, when the modulus is taken to be polynomial in n. The BKW algorithm can be viewed as consisting of three stages: (a) sample reduction, (b) hypothesis testing to recover a subset of the secret and (c) combining of candidate solution. On a high level, the first stage of the BKW algorithm can be described as a form of Gaussian elimination which, instead of treating each column independently, considers 'blocks' of b columns per iteration, where b is a parameter of the algorithm. Following this reduction, the second stage performs hypothesis tests to recover components of the secret vector $\mathbf{s}$. The third stage combines these components to recover the full secret $\mathbf{s}$. The main idea of the algorithm is to minimise the number of row operations (additions) in the first stage, as this has a strong influence in the number of samples required in the later stages for reliably recovering each of the components of $\mathbf{s}$.

The way we study the complexity of the BKW algorithm for solving the LWE problem is closely related to the method described in [14]: given an oracle that returns samples according to the probability distribution $L^{(n)}_{\mathbf{s},\chi}$, we use the algorithm's first stage to construct an oracle returning samples according to another distribution, which we call $B^{(n)}_{\mathbf{s},\chi,a}$, where $a = \lceil n/b \rceil$ denotes the number of 'levels' of addition. The complexity of the algorithm is related to the number of operations performed in this transformation, to obtain the required number of samples for hypothesis testing.

Details regarding the complexity of the initial and intermediate stages of the algorithm are given in the full version of this paper.

For the final hypothesis-testing stage, we wish to know the expected position of a counter for the correct guess $\mathbf{v} = \mathbf{s}'$ among the entries of S (where S is a collection of counters, in bijection with the possible guesses for a subset $\mathbf{s}'$ of elements of $\mathbf{s}$), since the expected position or rank of the correct counter determines the expected number of final hypothesis tests required to obtain $\mathbf{s}$. We clearly have a trade-off between the expected rank of the correct counter in S and the cost of the final hypothesis-testing stage. For instance, if we could guarantee that with probability 1 the correct counter was always in the highest position of each list, then the final hypothesis-testing stage could be omitted. If, on the other hand we could say that with probability 95% the correct counter was within the top 3 elements of each list S, then we would be required to carry out a non-trivial final hypothesis-testing stage, examining a certain number of combinations of elements from our lists to obtain $\mathbf{s}$.

It should be noted that this (more general) approach is not considered in the original presentation of the BKW algorithm and that, in the original presentation, it is assumed that the correct counter always assumes the highest position in each list S. We introduce a further parameter r, the expected rank of a correct counter within each list S. Clearly, for the original presentation of BKW, $r = 0$.

Thus, what remains to be established is the size $m = |F|$ (where F denotes the set of LWE samples available) needed such that the counter for the right guess $\mathbf{v} = \mathbf{s}'$ is expected among the largest r entries in S. By the Central Limit theorem, the distribution of $S_{\mathbf{v}}$ approaches a Normal distribution as m increases. Hence, for sufficiently large m we assume that we may approximate the discrete distribution $S_{\mathbf{v}}$ by a normal distribution [3]. If $\mathbb{N}\mu,\sigma^2$ denotes a Normal distribution with mean μ and standard deviation σ we denote the distribution for $\mathbf{v} = \mathbf{s}'$ by $D_c = \mathbb{N}\mathbb{E}_c, \mathrm{Var}_c$ and for $\mathbf{v} \neq \mathbf{s}'$ by $D_w = \mathbb{N}\mathbb{E}_w, \mathrm{Var}_w$.

Establishing m hence first of all means establishing $\mathbb{E}_c, \mathbb{E}_w, \mathrm{Var}_c$, and Var_w (see full version of paper).

Now, to estimate the rank of the counter $S_{\mathbf{s}'}$ in S given m samples, we compute the probability that $\mathbb{N}\mathbb{E}_w, \mathrm{Var}_w$ takes a smaller value than $\mathcal{N}(\mathbb{E}_c, \mathrm{Var}_c)$ if sampled $q^d - 1$ times (since there are $q^d - 1$ wrong guesses). That is, we compute the rank r by computing probability that the difference distribution $D_c - D_w$ takes a value ≤ 0 and by multiplying this value by $q^d - 1$. Hence, given a target rank r we can estimate m.

Lemma 3. *Let* $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ *be the Gaussian error function. Then, solving*

$$r \quad = \quad \frac{q^d - 1}{2} \left(1 + \mathrm{erf}\left(\frac{\mathbb{E}_w - \mathbb{E}_c}{\sqrt{2(\mathrm{Var}_c + \mathrm{Var}_w)}} \right) \right) \tag{1}$$

for m recovers the number of non-zero samples needed such that $S_{\mathbf{s}'}$ is expected to be among the first r entries of S.

Using Lemma 3 we can hence estimate the number of non-zero samples m we need to recover partial information about the secret $\mathbf{s}$. Finally, we need to extend this result to recover $\mathbf{s}$.

BKW Third Stage: Combination In the final stage, we need to examine the expected cost of the search needed through the $g := \lceil n/d \rceil$ lists in order to obtain the full secret. To decide on our final guess for $\mathbf{s}$, we need to set a threshold acceptance value for the distance between the actual noise distribution and the hypothetical noise distribution obtained through testing a guess for $\mathbf{s}$ against samples from $L^{(n)}_{\mathbf{s},\chi}$. Once we find a combination $\mathbf{s}_{(0,d)} \mid\mid \mathbf{s}_{(d,2d)} \mid\mid \dots \mid\mid \mathbf{s}_{(n-n \bmod d,n)}$ which falls beneath this threshold, we terminate our search and return $\mathbf{s} = \mathbf{s}_{(0,d)} \mid\mid \mathbf{s}_{(d,2d)} \mid\mid \dots \mid\mid \mathbf{s}_{(n-(n \bmod d),n)}$. We have $g := \lceil n/d \rceil$ lists for each of which we expect that the right guess has rank r.

More precisely, we denote by Y_h the random variable determined by the rank of a correct counter S_c in a list of h elements. Now, for a list of length q^d and a given rank r ($0 \leq r < q^d$), we have the (binomial-normal) compound distribution

$$\begin{aligned}
\Pr[Y_{q^d} = r] \quad &= \quad \int_x \left(\binom{q^d}{r+1} \cdot \Pr[e \leftarrow_{\$} D_w : e < x]^{(r+1)} \cdot \Pr[e \leftarrow_{\$} D_w : e \geq x]^{(q^d - r - 1)} \cdot \Pr[e \leftarrow_{\$} D_c : e = x] \right) \\
&= \quad \int_x \left(\binom{q^d}{r+1} \cdot p_x^{(r+1)} \cdot (1 - p_x)^{(q^d - r - 1)} \cdot \left(\frac{1}{\sqrt{2\pi \mathrm{Var}_c}} e^{-\frac{(x - \mathbb{E}_c)^2}{2\mathrm{Var}_c}} \right) \right) dx,
\end{aligned}$$

where $p_x = \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{x - \mathbb{E}_w}{\sqrt{2\mathrm{Var}_w}}\right)\right)$.

The following combination strategy has been devised with this distribution in mind: Let $i_0 \dots, i_{g-1}$ be indices pointing to the g subvectors currently considered for combination to the full solution. We initialise all $i_j = 0$ and test this candidate. If it fails our test we traverse in such a way that $\sum_{j=0}^{g-1} i_j$ strictly increases, i.e., we consider all indices that sum to 1 first, then all indices that sum to 2, etc. Overall, we expect to test $(r+1)^g$ candidates (recall, that we start counting at zero) until we test the correct one. To have a unique solution we need to test about $2n$ samples, each test costing $2n$ operations in $\mathbb{Z}_q$.

BKW: Complexity

We can now state our main theorem.

Theorem 4. *Let* $(\mathbf{a}_i, c_i)$ *be samples following* $L^{(n)}_{\mathbf{s},\chi}$, $0 < b \leq n$ *and* $d \leq b$ *be parameters,* $a = \lceil n/b \rceil$ *and* m, r *as in Lemma* 3*. The expected cost of the BKW algorithm to recover* $\mathbf{s}$ *is upper-bounded by*

$$\left\lceil \frac{n}{d} \right\rceil \cdot \left(\frac{q^d}{(q^d - 1)} \cdot \left(\frac{a}{2} \cdot (n+3) \cdot (a \cdot q^b + m) \right) \right) \tag{2}$$

additions in $\mathbb{Z}_q$ for the elimination step,

$$\left\lceil \frac{n}{d} \right\rceil \cdot \left(m \cdot q^d \right) \tag{3}$$

arithmetic operations in $\mathbb{Z}_q$ for the hypothesis-testing step, and

$$4n(r+1)^{(\lceil n/d \rceil)} \tag{4}$$

arithmetic operations in $\mathbb{Z}_q$ for the final combination step. Furthermore, at most

$$\lceil n/d \rceil \cdot \frac{q^d}{(q^d-1)} \left(a \cdot q^b + m\right) + 2n \tag{5}$$

calls to $L_{\mathbf{s},\chi}^{(n)}$ are needed.

Picking Parameters & Applications

In this section we apply Theorem 4 to various sets of parameters suggested in the literature. We stress that we always allow an unbounded number of queries to $L_{\mathbf{s},\chi}^{(n)}$, an assumption which does not carry over to any cryptosystem considered in this section. We also note that in order to compute concrete costs we require numerical approximations in various places, such as the computation of p_j and solving for m in Lemma 3. We used $2n$ bits of precision which seems to be sufficient for our purposes, i.e., increasing the precision further did not change our results. Finally, we stress that the results in section should be considered as upper bounds on the cost of running BKW on LWE instances considered here. That is, we do not claim that the parameter choices in this section are optimal, although they are based on extensive experiments.

In all cases below, we need to pick the parameters a, d and r. We pick $d > 1$ but r small to ensure that stage 3 does not dominate the overall computation; in particular, $d = 2$ and $r = 2$ seem to be good choices in our experiments. Furthermore, we set $a := t \cdot \log_2 n$ where t is a small constant, hence choosing t implies a. The parameter t is chosen to minimise additions while keeping $m \ll 2^n$. In this section, we assume that one operation in $\mathbb{Z}_q$ costs q bit operations.

Regev's original parameters

In [27] Regev also proposes a simple public-key encryption scheme with the suggested parameters $q \approx n^2$ and $\alpha = 1/(\sqrt{n} \cdot \log_2^2 n)$. We consider the parameter range $n = 64, \dots, 256$. In our experiments $t = 2.6$ produced the best results, i.e., higher values of t resulted in m growing too fast. Plugging these values into the formulas of Theorem 4 we get an overall complexity of

$$\frac{mn\left(n^8 - n^4 + \left(3.38n^9 + 10.14n^8\right)\log_2^2 n\right)}{2(n^4-1)} 2^{(2/2.6n)} + 4n3^{\left(\frac{1}{2}n\right)} + \frac{\left(0.65m^2n^{10} + 1.95m^2n^9\right)}{(n^4-1)} \log_2 n$$

If $m \in o(2^{\left(\frac{2}{2.6}n\right)})$ then this expression is dominated by $\frac{mn\left(n^8-n^4+\left(3.38n^9+10.14n^8\right)\log^2(n)\right)}{2(n^4-1)} 2^{(2/2.6n)}$ and hence $\in \mathcal{O}(2^{\left(\frac{2}{2.6}n\right)})$. However, since we compute m numerically, we have to rely on experimental evidence to verify this behaviour. Table 1 lists the estimated number of calls to $L_{\mathbf{s},\chi}^{(n)}$ ("$\log_2 \#L_{\mathbf{s},\chi}^{(n)}$"), the estimated number of required ring ("$\log_2 \#\mathbb{Z}_q$") and bit ("$\log_2 \#\mathbb{Z}_2$") operations, the costs in terms of ring operations for each of the three stages, and the number of "rows" in the "BKW matrix" ("$\log_2 n_r$"). To compare the observed costs with asymptotic complexity, Figure 1 plots $\Delta \log_2 \#\mathbb{Z}_q$, i.e., the ratio of $\log_2 \#\mathbb{Z}_q$ for consecutive values of n, and compares it with $2/2.6 \approx 0.7692$.[2]

Conclusion & Further Work

In this work we have provided what we believe is the first concrete analysis of the cost of running the BKW algorithm on LWE instances and applied this analysis to various sets of parameters found in the literature. Although we were unable to provide a closed form for the complexity of the

[2] To avoid a possible misunderstanding: Figure 1 does not show $\log_2 \#\mathbb{Z}_q/n$ but $\Delta \log_2 \#\mathbb{Z}_q$, i.e., it disregards the constant coefficient.

n	$\log_2 m$	$\log_2 n_r$	$\log_2 \#\mathbb{Z}_q$ in				$\log_2 \#\mathbb{Z}_2$	$\log_2 \#L_{\mathbf{s},\chi}^{(n)}$
			stage 1	stage 2	stage 3	total		
48	33.28	40.80	54.86	60.21	45.62	60.25	71.42	45.39
64	38.84	53.20	68.18	67.84	58.72	69.03	81.03	58.20
96	49.87	77.95	94.23	81.80	84.66	94.23	107.40	83.53
128	61.12	102.66	119.86	95.12	110.44	119.86	133.87	108.66
160	72.08	127.33	145.23	107.69	136.12	145.23	159.88	133.65
192	83.05	152.00	170.48	119.97	161.74	170.48	185.65	158.58
224	94.34	176.65	195.62	132.37	187.32	195.62	211.24	183.46
256	105.30	201.30	220.69	144.30	212.88	220.69	236.69	208.30

Table 1: Cost of finding $\mathbf{s}$ for parameters suggested in [27] with $d = 2, t = 2.6, r = 2$.

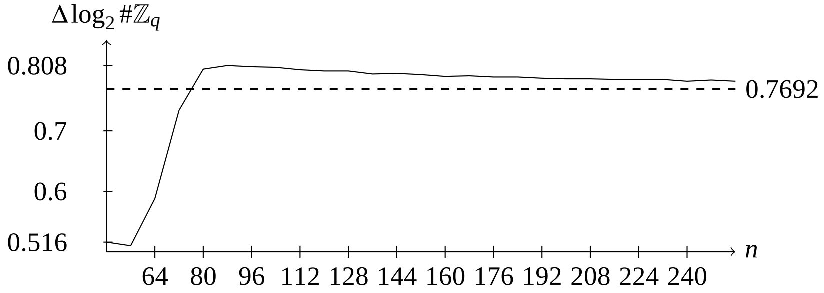


Figure 1: $\Delta \log_2 \#\mathbb{Z}_q$ vs 0.7692.

BKW algorithm, since the value m in Theorem 4 is computed using numerical approximation, we believe that our work presents an important contribution to the better understanding of both the theoretic aspects of the algorithm as well as the security provided by LWE-based cryptographic schemes. Besides potential further refinements in our analysis, we consider providing such a closed, explicit expression for the complexity of the BKW algorithm on LWE as a pressing research question for future work. Finally, finding optimal parameters and comparing the results with lattice-based solutions and the Arora-Ge algorithm are logical next steps.

References

[1] M. R. Albrecht, P. Farshim, J.-C. Faugère, and L. Perret. Polly Cracker, revisited. In *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 179–196, Berlin, Heidelberg, New York, 2011. Springer Verlag. full version available as Cryptology ePrint Archive, Report 2011/289, 2011 `http://eprint.iacr.org/`.

[2] S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer Verlag, 2011.

[3] T. Baigneres, P. Junod, and S. Vaudenay. How far can we go beyond linear cryptanalysis? *Advances in Cryptology - Asiacrypt 2004*, 2004.

[4] A. Becker, J.-S. Coron, and A. Joux. Improved generic algorithms for hard Knapsacks. In K. G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 364–385. Springer Verlag, 2011.

[5] A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves Information Set Decoding. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 520–536. Springer Verlag, 2012.

[6] D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In J. Buchmann and J. Ding, editors, *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer Verlag, 2008.

[7] D. J. Bernstein, T. Lange, and C. Peters. Smaller decoding exponents: Ball-collision decoding. In P. Rogaway, editor, *Advances in Cryptology - CRYPTO 2011*, volume 6841, pages 743–760. Springer Verlag, 2011.

[8] L. Bettale, J.-C. Faugère, and L. Perret. Hybrid Approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2010.

[9] L. Bettale, J.-C. Faugère, and L. Perret. Solving polynomial systems over finite fields: Improved analysis of the Hybrid Approach. In *ISSAC '12: Proceedings of the 2012 international symposium on Symbolic and algebraic computation*, ISSAC '12, pages 1–12, New York, NY, USA, 2012. ACM.

[10] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In R. Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 97–106. IEEE, 2011.

[12] Y. Chen and P. Q. Nguyen. BKZ 2.0: better lattice security estimates. In *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20, Berlin, Heidelberg, 2011. Springer Verlag.

[13] M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 88–105. Springer Verlag, 2009.

[14] P.-A. Fouque and É. Levieil. An improved LPN algorithm. In R. D. Prisco and M. Yung, editors, *Security and Cryptography for Networks, 5th International Conference, SCN 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer Verlag, 2006.

[15] N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 257–278. Springer Verlag, 2010.

[16] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. Available at `http://crypto.stanford.edu/craig`.

[17] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[18] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 447–464. Springer Verlag, 2011.

[19] N. Howgrave-Graham and A. Joux. New generic algorithms for hard Knapsacks. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 235–256. Springer Verlag, 2010.

[20] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In D. H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 107–124. Springer Verlag, 2011.

[21] D. Micciancio and O. Regev. Lattice-based cryptography. In D. J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer Verlag, Berlin, Heidelberg, New York, 2009.

[22] I. Morel, D. Stehlé, and G. Villard. H-LLL: using householder inside LLL. In J. R. Johnson, H. Park, and E. Kaltofen, editors, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2009*, pages 271–278. ACM, 2009.

[23] P. Q. Nguyen. Lattice reduction algorithms: Theory and practice. In K. G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 2–6. Springer Verlag, 2011.

[24] P. Q. Nguyen and D. Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Transactions on Algorithms*, 5(4), 2009.

[25] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 333–342. ACM, 2009.

[26] X. Pujol and D. Stehlé. Solving the shortest lattice vector problem in time $2^{2.465n}$. *IACR Cryptology ePrint Archive*, 2009:605, 2009.

[27] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[28] M. Rückert and M. Schneider. Estimating the security of lattice-based cryptosystems. *IACR Cryptology ePrint Archive*, 2010:137, 2010.

M. R. Albrecht INRIA, LIP6, CNRS
malb@lip6.fr
C. Cid Royal Holloway, University of London
carlos.cid@rhul.ac.uk
J.-C. Faugère INRIA, LIP6, CNRS
jean-charles.faugere@inria.fr
R. Fitzpatrick Royal Holloway, University of London
robert.fitzpatrick.2010@live.rhul.ac.uk
L. Perret INRIA, LIP6, CNRS
ludovic.perret@lip6.fr

Aperiodic logarithmic signatures

Barbara Baumeister and Jan de Wiljes

Introduction

In the early 2000's Magliveras, Stinson and Tran van Trung introduced two public key cryptosystems, MST_1 and MST_2, based on factorizations, covers and logarithmic signatures, of finite non-abelian groups [8]. Recently, Lempken, Magliveras, Tran van Trung and Wei [5] developed a third cryptosystem, MST_3. Several authors have dealt with the security of these schemes, see for instance [3] or [1]. As a reaction Svaba and Tran van Trung published a newer, strengthened version of MST_3 [12].

A main question is how to produce covers and logarithmic signatures for a group. Blackburn et al. [1] suggested a construction of so called amalgamated transversal logarithmic signatures *ATLS* from exact transversal logarithmic signatures. Based on the use of these amalgamated transversal logarithmic signatures they presented a successful attack on the system MST_3.

In this paper we propose a method to construct logarithmic signatures which are not amalgamated transversal and further do not even have the property of being periodic. The *ATLS* are periodic, see [1, Lemma 2.1], and this property was crucial for breaking the system MST_3 (see cases 2 and 3 in subsection 4.3 in [1]). The idea for our construction is based on the theory in Szabó's book about group factorizations [13].

Covers and logarithmic signatures

Throughout this paper, G denotes a finite group and every set is assumed to be finite. Further information can be found in [2], [5], [6], [7] and [8].

Let $K \subseteq G$ and $\alpha = [A_1, \ldots, A_s]$ be a sequence of sequences $A_i = [a_{i,1}, \ldots, a_{i,r_i}]$ with $a_{i,j} \in G$, such that $\sum_{i=1}^{s} |A_i|$ is bounded by a polynomial in $\lceil \log |K| \rceil$. Then α is a *cover for* $K \subseteq G$, if every product $a_{1,j_1} \cdots a_{s,j_s}$ lies in K and if every $g \in K$ can be written as

$$g = a_{1,j_1} \cdots a_{s,j_s} \tag{1}$$

with $j_i \in \{1, \ldots, |A_i|\}$. If, moreover, the tuple $(j_1, \ldots, j_s)$ is unique for every $k \in K$ then α is called a *logarithmic signature for* K. We call the product $a_{1,j_1} \cdots a_{s,j_s}$ in (1) a *factorization* of g w.r.t. α. Two factorizations $a_{1,j_1} \cdots a_{s,j_s}$ and $a_{1,h_1} \cdots a_{s,h_s}$ of g are *different* if $(j_1, \ldots, j_s) \neq (h_1, \ldots, h_s)$. (Note that for $\alpha = [[a,a],[b,b]]$ the element ab has four different factorizations $a \cdot b$.)

If $\alpha = [A_1, \ldots, A_s]$ is a logarithmic signature of K) with $r_i := |A_i|$ for all $i \in \{1, \ldots, s\}$, then the sequence A_i is called a *block of* α and the sequence $(r_1, \ldots, r_s)$ the *type of* α. The *length of* α is

$$l(\alpha) := \sum_{i=1}^{s} r_i.$$

Covers of minimal length are noteworthy due to the fact that less memory capacity has to be used. The interested reader is referred to [6], [10] and [11] for information on this issue.

For the application in cryptography the following distinction is made. A logarithmic signature β for K is *tame* if every $g \in K$ can be factorized in polynomial time (polynomial in $\lceil log|K| \rceil$) w.r.t. to β, otherwise β is called *wild*. Last not least we call a logarithmic signature β of G *aperiodic* if none of the blocks B_i is periodic. The set of all aperiodic logarithmic signatures for a group G is denoted by $\mathcal{A}(G)$.

Let $\alpha = [A_1, \ldots, A_s])$ be a cover for $K \subseteq G$ of type $(r_1, \ldots, r_s)$ with $A_i = [a_{i,1}, \ldots, a_{i,r_i}]$. Let τ_α be the canonic bijection from $\mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_s}$ to $\mathbb{Z}_m$, where $m := \prod_{i=1}^{s} r_i$, $m_1 := 1$ and $m_i := \prod_{l=1}^{i-1} r_l$ for $i \geq 2$, i. e.

$$\tau_\alpha : \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_s} \to \mathbb{Z}_m, \tau_\alpha(j_1, \ldots, j_s) := \sum_{i=1}^{s} j_i m_i$$

That is a generalization of n-ary representations. Let $\breve{\alpha} : \mathbb{Z}_m \to K$ be the surjection:

$$\breve{\alpha}(x) := a_{1,j_1+1} \cdots a_{s,j_s+1}, \text{ where } (j_1, \ldots, j_s) = \tau_\alpha^{-1}(x).$$

Note that τ_α^{-1} can be computed efficiently (using Euclid's algorithm) and therefore the same is true for $\breve{\alpha}$.

The cryptosystem MST$_3$

Alice chooses a public non-abelian group G with large center Z and generates

- a tame logarithmic signature $\beta = [B_1, \ldots, B_s]$ of Z of type $(r_1, \ldots, r_s)$
- and a random cover $\alpha = [A_1, \ldots, A_s]$ for a subset K of G with $a_{i,j_i} \in G \backslash Z$ for all $i \in \{1, \ldots, s\}$ and $j_i \in \{1, \ldots, r_i\}$, which is of the same type as β.

Then she chooses random elements $t_0, \ldots, t_s \in G \backslash Z$ and computes the following covers:

- $\tilde{\alpha} = [\tilde{A}_1, \ldots, \tilde{A}_s]$, whereat $\tilde{A}_i = t_{i-1}^{-1} A_i t_i$ for all $i \in \{1, \ldots, s\}$,
- $\gamma := [H_1, \ldots, H_s]$ with $H_i := [b_{i,1}\tilde{a}_{i,1}, \ldots, b_{i,r_i}\tilde{a}_{i,r_i}]$ for all $i \in \{1, \ldots, s\}$.

The public key is (α, γ) and the private key is $(\beta, t_0, \ldots, t_s)$.

To encrypt an element $x \in \mathbb{Z}_{|Z|}$, Bob computes $y_1 = \breve{\alpha}(x)$ and $y_2 = \breve{\gamma}(x)$ and sends $y = (y_1, y_2)$ to Alice.

Alice decrypts y by calculating $\breve{\beta}^{-1}(y_2 t_s^{-1} y_1^{-1} t_0)$ which equals x. As β is tame, the decryption-algorithm is efficient.

The cryptographic hypothesis is the problem of factorizing w. r. t. the random cover α. Furthermore it has to be hard for the attacker to reconstruct the private key by using the public key. For information on these two issues we refer the reader to [1] and [9].

Constructing aperiodic tame logarithmic signatures

Now we will concentrate on the construction of β and we will restrict us to elementary abelian 2-groups (denoted by 2^n), although all results in Section hold for every abelian group. Note that β is supposed to be secret. As in a logarithmic signature β every group element is at most once in a block, we will consider sets instead of sequences in the first two paragraphs of this section to simplify the notation.

Szabó showed in [13, Theorem 7.3.1]:

Theorem 1 ([13, Theorem 7.3.1])**.** *Let G be an elementary abelian 2-group. There exists an aperiodic logarithmic signature β of type $(r_1, \ldots, r_s)$ with $r_1 \geq \cdots \geq r_s \geq 2$ if*

- *$s = 2$ and $r_2 \geq 8$ or*
- *$s \geq 3$ and $r_1 \geq 8$, $r_s \geq 4$ holds.*

We use the idea of the proof of this theorem to construct tame aperiodic logarithmic signatures for elementary abelian 2-groups, for example for the center of a Suzuki 2-Group.

The algorithm

The following algorithm constructs a new logarithmic signature out of a subgroup and a left transversal of that subgroup. The realization of some rather vague steps in the algorithm, namely the construction of δ and all $\alpha^{(j_1,\dots,j_s)}$, are filled by considering some special subgroups of G.

Algorithm 1: Construction of aperiodic Logarithmic Signatures

```
Output: $\beta \in \mathcal{A}(G)$.
Choose an abelian group $G$, a subgroup $U$ of $G$ and a transversal $R$ of $U$ in $G$;
Generate a logarithmic signature of $R$
    $\delta = [D_1, \dots, D_s]$ with $D_i = \{d_{i,1}, \dots, d_{i,r_i}\}$
of type $(r_1, \dots, r_s)$ and logarithmic signatures of $U$
    $\alpha^{(j_1,\dots,j_s)} := \left[A_1^{(j_1)}, \dots, A_s^{(j_s)}\right]$
for all $(j_1, \dots, j_s) \in \{1, \dots, r_1\} \times \cdots \times \{1, \dots, r_s\}$
Compute $\beta := [B_1, \dots, B_s]$ by
    $B_1 := d_{1,1}A_1^{(1)} \cup \cdots \cup d_{1,r_1}A_1^{(r_1)}, \cdots ,$
    $B_s := d_{s,1}A_s^{(1)} \cup \cdots \cup d_{s,r_s}A_s^{(r_s)}.$
```

Notice that all logarithmic signatures $\alpha^{(j_1,\dots,j_s)}$ are used for the construction of β.

Example 2. *We choose $G := \langle u, v, w, x, y, z \rangle = 2^6$, $U := \langle u, v, w, x \rangle$, $R := \{1, y, z, yz\}$ and set $D_1 := \{1, z\}$, $D_2 := \{1, y\}$, and*

$$A_1^{(1)} := \{1, u, v, uv\}, A_1^{(2)} := \{1, w, x, wx\},$$
$$A_2^{(1)} := \{1, uw, vx, uvwx\}, A_2^{(2)} := \{1, ux, uvw, vwx\}.$$

We get $B_1 = \{1, u, v, uv, z, wz, xz, wxz\}, B_2 = \{1, uw, vx, uvwx, y, uxy, uvwy, vwxy\}$. Neither of these two blocks is periodic. It follows that $\beta \in \mathcal{A}(G)$ of type $(8, 8)$.

Theorem 3. *The sequence β constructed by the algorithm is a logarithmic signature for G of type $(l_1, \dots, l_s)$, where $l_i := \sum_{j=1}^{r_i} |A_i^{(j)}|$.*

Proposition 4. *The logarithmic signature β is tame if δ and all $\alpha^{(j_1,\dots,j_s)}$ are tame and if $|R|$ is bounded by a polynomial in $\lceil \log |G| \rceil$ (then for every $g \in G$ the coset representative in R which lies in the same coset as g can be found efficiently).*

Remark 5. *The last assumption of Proposition 4 is not required if G is given in form of a maximal set of generators $\{g_1, \dots, g_t\}$ with the property, that every element can be represented uniquely, where $U = \langle g_1, \dots, g_i \rangle$ and $R = \langle g_{i+1}, \dots, g_t \rangle$. In that case we get the desired coset representative by using a projection.*

Concrete construction of β.

Let $G = 2^n$ be an elementary abelian group of order $n \in \mathbb{N}_{>7}$. Then we may consider G as an $\mathbb{F}_2$-vector space. Let $\mathcal{B} = (g_1, \dots, g_n)$ be an $\mathbb{F}_2$-basis for G and let $v = (v_1, \dots, v_{2s})$ be a partition of n. Then we consider the following decomposition, using the notation $\mathfrak{v}_i := \sum_{k=1}^{i} v_k$:

$$G = \underbrace{\underbrace{\langle g_1, \dots, g_{\mathfrak{v}_1}\rangle}_{U_1} \times \dots \times \underbrace{\langle g_{\mathfrak{v}_{s-1}+1}, \dots, g_{\mathfrak{v}_s}\rangle}_{U_s}}_{U} \times$$
$$\underbrace{\underbrace{\langle g_{\mathfrak{v}_s+1}, \dots, g_{\mathfrak{v}_{s+1}}\rangle}_{D_1} \times \dots \times \underbrace{\langle g_{\mathfrak{v}_{2s-1}+1}, \dots, g_{\mathfrak{v}_{2s}}\rangle}_{D_s}}_{R},$$

Moreover, we choose a tame logarithmic signature β' for an elementary abelian group of order $2^{v_1+v_{s+1}}$. Then we choose subsets $K_i := \{k_i^{(1)}, \dots, k_i^{(r_i)}\} \subseteq (U_1 \times \dots \times U_{i-1})^{\#}$ of size r_i for every $i \in \{2, \dots, s\}$ and we construct the logarithmic signature $\beta = [\beta', B_2, \dots, B_s]$ using Algorithm 1 by setting

$$\delta := [D_2, \dots, D_s],$$
$$A_i^{(j)} := [k_i^{(j)} u_{i,1}, \dots, k_i^{(j)} u_{i,m_i}, 1], \text{ for } i = 2, \dots, s \text{ and } j = 1, \dots, r_i,$$

where $U_i = [u_{i,1}, \dots, u_{i,m_i}, 1]$. Then β is an aperiodic, tame logarithmic signature for G. Some immediat questions arise. For instance:

Question Can we store the group G represented by $\mathcal{B}$ without revealing β?

Moreover we present an algorithm for the factorization of a group element g w.r.t. the just constructed logarithmic signature β.

Complexity issues are shortly discussed.

References

[1] S. R. Blackburn, C. Cid and C. Mullan, Cryptanalysis of the MST_3 Public Key Cryptosystem, *J. Math. Cryptol.*, 3 (4): 321–328, 2009.

[2] C. A. Cusack, *Group factorizations in cryptography*, Ph.D. thesis, University of Nebraska, 2000.

[3] M. I. González Vasco, A. L. Pérez del Pozo and P. Taborda Duarte, A note on the security of MST_3, *Des. Codes Cryptogr.*, 55 2-3:189–200, 2010.

[4] B. Huppert and N. Blackburn, *Finite Groups II*, Springer, Berlin–Heidelberg–New York, 1982.

[5] W. Lempken, S. S. Magliveras, Tran van Trung and W. Wei, A Public Key Cryptosystem Based on Non-abelian Finite Groups, *J. Cryptol.*, 22 (1):62-74, 2009.

[6] W. Lempken and Tran van Trung, On Minimal Logarithmic Signatures of Finite Groups, *Experimental Mathematics*, 14(3):257–269, 2005.

[7] S. S. Magliveras and N. D. Menon, Algebraic Properties of Cryptosystem *PGM, J. Cryptol.*, 5 (3):167-183, 1992.

[8] S. S. Magliveras, D. R. Stinson and Tran van Trung, New Approaches to Designing Public Key Cryptosystems Using One-Way Functions and Trapdoors in Finite Groups, *J. Cryptol.* , 15(4):285–297, 2002.

[9] S. S. Magliveras, P. Svaba, Tran van Trung and P. Zajac, On the security of a realization of cryptosystem MST_3, *Tatra Mt. Math. Publ.*, 41:65–78, 2008.

[10] Nidhi Singhi, Nikhil Singhi and S. S. Magliveras, Minimal logarithmic signatures for finite groups of Lie type, *Des. Codes and Cryptogr.* **55** (2010), nos. 2–3, 243–260.

[11] Nikhii Singhi and Nidhi Singhi, Minimal logarithmic signatures for classical groups, *Des. Codes and Cryptogr.*, 60 (2):183–195, 2010.

[12] Pavol Svaba and Tran van Trung, Public key cryptosystem MST_3: cryptoanalysis and realization, *J. Math. Cryptol.*, 4 (3):271–315, 2010.

[13] S. Szabó, *Topics in Factorization of Abelian Groups*, Birkhäuser, Basel, 2004.

B. Baumeister Universität Bielefeld
B.Baumeister@math.uni-bielefeld.de
J. de Wiljes Universität Hildesheim
wiljes@uni-hildesheim.de

Using symmetries and fast change of ordering in the Index Calculus for Elliptic Curves Discrete Logarithm.

Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault

This abstract presents results on polynomial systems involved in an algebraic attack on elliptic curves cryptosystems. The security of these cryptosystems is based on the difficulty to solve the *elliptic curves discrete logarithm problem (ECDLP)*: let E be an elliptic curve defined over a finite field $\mathbb{K}$. The set of its rational points forms a commutative group, $E(\mathbb{K})$. Given two points P and Q of $E(\mathbb{K})$ the ECDLP is to find if it exists, an integer x such that $Q=[x]P$. The notation $[x]P$ denotes, as usual, the multiplication of P by x.

Except for few *weak* curves (as curves with small enough embedding degree or curves defined over $\mathbb{F}_p$ of order p), the best known algorithms to solve the ECDLP are generic algorithms. A generic algorithm is an algorithm to solve the DLP in any group. A result from Shoup [18] shows that these algorithms are exponential in general. Among this algorithms, the Pollard rho method [17] is the most optimal and its complexity is given, up to a constant factor, by the square root of the order of the curve.

In [11], it is proposed an index calculus attack to solve the ECDLP defined over a non prime finite field $\mathbb{F}_{q^n}$ where $n>1$. Later on, Diem [1, 2] obtained rigorous proofs that for some particular families of curves the discrete logarithm problem can be solved in subexponential time.

Let us recall the principle of the algorithm: given P and Q, two points of $E(\mathbb{F}_{q^n})$, we look for x, if it exists, such that $Q=[x]P$

1. Compute the factor base $\mathcal{F}=\{(x,y)\in E(\mathbb{F}_{q^n})\mid x\in\mathbb{F}_q\}$.
2. Look for at least $\#\mathcal{F}+1$ relations of the form: $[a_j]P\oplus[b_j]Q=P_1\oplus\cdots\oplus P_n$, where $P_1,\cdots,P_n\in\mathcal{F}$ and a_j and b_j are randomly picked up in $\mathbb{Z}$.
3. Finally, by using linear algebra, recover the discrete logarithm x.

Using the double large prime variation [12] and for a fixed degree extension n, the complexity of this index calculus attack is $O(q^{2-\frac{2}{n}})$. It is thus faster than Pollard rho method in $O(q^{\frac{n}{2}})$ for $n\geq 3$. However, this complexity hides an exponential dependance in n in step 2 due to the resolution of the point decomposition problem.

Definition 1. *The* point decomposition problem, *denoted* **PDP** *in this paper, is: Given a point R in an elliptic curve $E(\mathbb{F}_{q^n})$ with a factor base $\mathcal{F}$ formed of the points with an $\mathbb{F}_q$-rational abscissa, find, if they exist, $P_1,\ldots,P_n$ in $\mathcal{F}$, such that $R=P_1\oplus\cdots\oplus P_n$.*

The group law of an elliptic curve being given by rational fractions in terms of the coordinates of the summing points, one way to solve the PDP is to model it as a polynomial system. Hence, the resolution of the PDP is equivalent to solve a polynomial system with coefficients in a finite field. To solve polynomial systems in finite field we use Gröbner basis. As usual, the resolution using Gröbner basis requires two steps. First, by using efficient algorithms to compute Gröbner basis as F_4 [3] or F_5 [4], we compute a DRL Gröbner basis of the system to solve. Then, by using a change of ordering algorithm as FGLM [5, 7], we compute a LEX Gröbner basis from which one can read off the solutions of the system. In this context, the PDP has a complexity in $O\left(\log(q)\left(\binom{n+d}{d}^{\omega}+n\cdot 2^{3n(n-1)}\right)\right)$ where $2\leq\omega<3$ is the linear algebra constant and d is a bound on the maximal degree reached during the computation of Gröbner basis with F_4 or F_5. The second part of the PDP complexity is due to the complexity of the FGLM algorithm which is polynomial in the number of solutions of the polynomial system with exponent at most 3. This step is often the bottleneck of the polynomial systems solving.

The main topic of this paper is to decrease the complexity of solving the PDP and thus the exponential dependance in n in the index calculus attack. To this end, we proceed in two steps.

First we give a new general change of ordering algorithm of interest independent of the PDP. This new algorithm follows the approach of [5] but we do not assume that the multiplication matrix is sparse. We replace the step of the sparse FGLM which uses the sparsity of the matrix by an usual approach introduced by Keller-Gehrig in [14]. The complexity of this change of ordering is still polynomial in the number of solutions of the system to solve but the exponent is decreases to ω up to logarithm factors.

Theorem 2. *Let $\mathcal{I}$ be a shape position ideal of $\mathbb{K}[x_1,\dots,x_n]$ with $\mathbb{K}$ a finite field. We denote by G_{DRL} the DRL Gröbner basis of $\mathcal{I}$. Given the matrix representation of the multiplication by the smallest variable in $\mathbb{K}[x_1,\dots,x_n]/\langle G_{DRL}\rangle$, computing the LEX Gröbner basis of an ideal in shape position can be done in $O(\log(D)(nD+D^{\omega}))$ where D is the degree of $\mathcal{I}$.*

Under the Moreno-Socias conjecture [16], it is shown in [6] that computing the multiplication matrix by the smallest variable in $\mathbb{K}[x_1,\dots,x_n]/\langle G_{\mathrm{DRL}}\rangle$ requires no arithmetic operations. Hence, we can extend our theorem.

Theorem 3. *Under the Moreno-Socias conjecture, the complexity of the change of ordering to pass from the DRL order to the LEX order for generic systems is given by $O(\log(D)(nD+D^{\omega}))$.*

However, polynomial systems coming from applications (in particular, the PDP problem) are often not generic and Theorem 3 can not be applied. To ensure that the construction of the multiplication matrix T is negligible compared to the change of ordering, we propose a new strategy to solve polynomial systems.

First we compute a DRL Gröbner basis of the system to solve. Then we try to compute T. If we can compute T for free then we compute the LEX basis. If we can not compute T for free then we consider the new ideal $\mathcal{I}^{(t)}$ generated by $G_{\mathrm{DRL}} \cup \{t-\lambda_1 x_1-\cdots-\lambda_n x_n\} \subset \mathbb{K}[x_1,\dots,x_n,t]$ where the λ_i's are randomly chosen in $\mathbb{K}$. Finally, we compute the DRL Gröbner basis of $\mathcal{I}^{(t)}$ and then we apply the change of ordering. This new strategy is summarized in Figure 1.

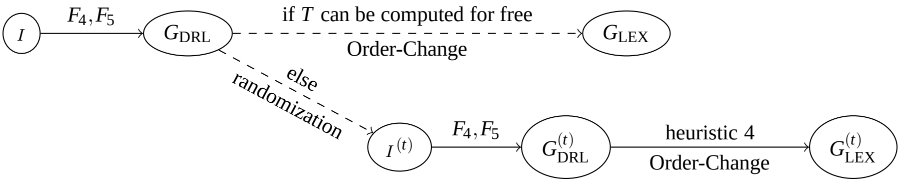


Figure 1: New strategy for polynomial systems solving.

From [15] the degree of regularity is not changed, when we add the variable t, and the number of solutions neither thus the asymptotic complexity of the new strategy to solve polynomial systems is the same that the original strategy. Our experiments (see Table 2) show that this modification allows to neglect the cost of computing the multiplication matrix.

Heuristic 4. *If $\mathcal{I}$ is a non-generic ideal, let $\mathcal{I}^{(t)}$ be the ideal generated by $G_{DRL} \cup \{t-\lambda_1 x_1-\cdots-\lambda_n x_n\} \subset \mathbb{K}[x_1,\dots,x_n,t]$ where the λ_1's are randomly chosen in $\mathbb{K}$. Then no arithmetic operations are required to compute the multiplication matrix by the variable t in $\mathbb{K}[x_1,\dots,x_n,t]/\langle \mathcal{I}^{(t)}\rangle$ w.r.t. DRL order.*

Conjecture 5. *The complexity of the change of ordering to pass from the DRL order to the LEX order for non-generic ideal in shape position is given by $O(\log(D)(nD+D^{\omega}))$.*

Remark 6. *Contrary to FGLM [7] algorithm only the multiplication matrix by the smallest variable is required in the algorithm introduced in this paper. So far, no known algorithm computes all the multiplication matrices in less than $O(nD^3)$ arithmetic operations. Hence, even if the change of ordering part of the FGLM algorithm can use the fast matrix multiplication, its total complexity can not be less than $O(nD^3)$.*

Then, we reveal some elliptic curves (Edwards or Jacobi intersections curves), where one can make use of the presence of a small rational subgroup to speed-up the index calculus algorithm, and especially the PDP step.

More precisely, the action of the 2-torsion of these curves induces some symmetries to the polynomial system to solve. Indeed, the action of the 2-torsion on the curve translates into the polynomial systems to solve in a very simple manner: any sign change on an even number of variables is allowed. Moreover, the order of the point in the decomposition of any point of the curve is not significant. This implies that all permutations of variables are also allowed. This correspond to the action of the well known symmetric group. These two actions combined gives the so called dihedral Coxeter group $D_n = (\mathbb{Z}/2\mathbb{Z})^{n-1} \rtimes \mathfrak{S}_n$. Using invariant theory techniques [19] we can thus expressed the system in adapted coordinates and therefore the number of solutions is reduced by a factor $2^{n-1} \cdot n!$ (the cardinality of the Dihedral Coxeter group). This yields a speed-up by a factor $2^{3(n-1)}$ (or $2^{\omega(n-1)}$ for the heuristic case) in the change of ordering step, compared to the general case.

Theorem 7. *Let E be an elliptic curve defined over a non binary field $\mathbb{F}_{q^n}$ where $n > 1$. If E can be put in twisted Edwards or twisted Jacobi intersections representation then the complexity of solving the PDP is*

- *(proven complexity)* $O\left(\log(q)\left(\binom{n+d}{d}^{\omega} + n \cdot 2^{3(n-1)^2}\right)\right)$
- *(heuristic complexity)* $O\left(\log(q)\left(\binom{n+d}{d}^{\omega} + n^2 \cdot 2^{\omega(n-1)^2}\right)\right)$

where $2 \leq \omega < 3$ is the linear algebra constant, and d is the degree of regularity which bounds the maximal degree reached by polynomials during the computation of Gröbner basis with F_5.

Usually, the step which dominates the complexity of Gröbner basis computation is the change of ordering. In theory, it is difficult to estimate this predominance. Indeed, except for some classes of polynomial systems as bilinear systems, regular systems *etc*, it is not easy to estimate the degree of regularity of the system. However, experimental results can help us to guess which step dominates in practice. We compare our new resolution of the PDP (denoted T_2 in Tables 3 and 4) with the original method (denoted W. [11]).

For $n = 4$, we can observe that taking into account the symmetries, dramatically decreases the computing time of the PDP resolution, by a factor of about 100, see Table 3. Moreover, from these experiments it seems that the computation of the DRL Gröbner basis is more expensive that the change of ordering algorithm.

One of the main improvement brought by this work, is that we are now able to solve the polynomial systems coming from the summation polynomials for $n = 5$ when the symmetries and the new strategy for polynomial systems solving (see Figure 1) are used. Still, these computation are not feasible with MAGMA and we use the FGb library. The timings are given in Table 4. One can notice that using symmetries is not sufficient to solve this instance of the PDP and the bottleneck is still the change of ordering step. Nevertheless, this instance can be solved by using the new strategy for Gröbner basis computation proposed in the first part of the paper. Here, the change of ordering step seems not to be the dominant step of the computation.

In practice, to solve more instances of the PDP, this new approach can be combined with that of Joux and Vitse [13]. Instead of looking for decomposition of a point in n points, they look for only $n-1$ points. This decreases the difficulty to solve one polynomial system, but this increases the number of polynomial systems to solve in the index calculus attack.

The results about change of ordering algorithm have been submitted to ISSAC for a poster presentation and the full paper about the resolution of the PDP [8] has been submitted to Journal of Cryptology.

References

[1] C. Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147: 75–104, 2011.

[2] C. Diem. On the discrete logarithm problem in class groups of curves. *Math. Comp*, 80: 443–475, 2011.

[3] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999.

[4] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM.

[5] J.-C. Faugère and C. Mou. Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices. In *ISSAC '11: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 1–8, New York, NY, USA, 2011. ACM.

[6] J.-C. Faugère and C. Mou. Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices (extended version), 2012. Article currently in progress.

[7] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[8] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Using symmetries in the index calculus for elliptic curves discrete logarithm. Cryptology ePrint Archive, Report 2012/199, 2012. `http://eprint.iacr.org/`.

[9] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Fast change of ordering with exponent ω, 2012. Available at `http://www-polsys.lip6.fr/~huot/unpublished/orderChange.pdf`.

[10] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Using symmetries and fast change of ordering in the index calculus for elliptic curves discrete logarithm., 2012. Available at `http://www-polsys.lip6.fr/~huot/pdf/scc2012_full.pdf`.

[11] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *Journal of Symbolic Computation*, 44(12):1690–1702, 2009.

[12] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Mathematics of Computation*, 76:475–492, 2007.

[13] A. Joux and V. Vitse. Elliptic Curve Discrete Logarithm Problem over Small Degree Extension Fields. Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$. To appear in Journal of Cryptology, Springer, DOI: 10.1007/s00145-011-9116-z., 2011.

[14] W. Keller-Gehrig. Fast algorithms for the characteristic polynomial. *Theor. Comput. Sci.*, 36: 309–317, June 1985.

	D	Density	Type	Const. T	Order-Change
Random $n = 16$	2^{16}	18.3%	I/II	228.6s	15005.3s
Edwards $n = 4$ $\mathfrak{S}_n + T_2$	512	27.61%	I/II/III	0.034s, 134NF	0.36s
Edwards $n = 4$ $\mathfrak{S}_n + T_2$ (new)	512	19.41%	I/II	0.00s	**0.02s**
Edwards $n = 5$ $\mathfrak{S}_n + T_2$	2^{16}		I/II/III	$>$ 2 days	$>$ 2 days
Edwards $n = 5$ $\mathfrak{S}_n + T_2$ (new)	2^{16}	9.31%	I/II	11.65s	**7865.67s**
Eco 14	2^{12}	11.50%	I/II/III	1100.08s, 2353NF	1102.55s
Eco 14 (new)	2^{12}	26.41%	I/II	0.08s	**1.97s**
[9, Example 1], $n = 11$	2^{11}	31.90%	I/II/III	7020.89s, 1023NF	7543.49s
[9, Example 1], $n = 11$ (new)	2^{11}	21.53%	I/II	0.15s	**5.30s**
[9, Example 1], $n = 16$	2^{16}		I/II/III	$>$ 2 days	$>$ 2 days
[9, Example 1], $n = 16$ (new)	2^{16}	18.33%	I/II	195.0s	**52558.7s**

Table 2: Computing time of LEX Gröbner basis with strategy in Figure 1 and construction of the multiplication matrix by the smallest variable for non generic systems. Computation with FGb on a 3.47 GHz Intel® Xeon® X5677 CPU.

$\log_2(q)$		F_4 (s)	Change of ordering (s)	Total (s)
16	W. [11]	4	531	535
	T_2	0	3	**3**
128	W. [11]	532	5305	5837
	T_2	31	23	**54**

Table 3: Computing time of Gröbner basis to solve the PDP with MAGMA (V2-17.1) on a 2.93 GHz Intel® Xeon® E7220 CPU for $n = 4$.

[15] D. Lazard. Gröbner bases, gaussian elimination and resolution of systems of algebraic equations. In J. van Hulzen, editor, *Computer Algebra*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer Berlin / Heidelberg, 1983.

[16] G. Moreno-Socias. Degrevlex gröbner bases of generic complete intersections. *Journal of Pure and Applied Algebra*, 180(3):263 – 283, 2003.

[17] J. Pollard. Monte carlo methods for index computation mod p. *Math. Comp.*, 32(143):918–924, July 1978.

[18] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, pages 256–266. Springer-Verlag, 1997.

[19] B. Sturmfels. *Algorithms in Invariant Theory (Texts and Monographs in Symbolic Computation)*. Springer Publishing Company, Incorporated, 2nd ed.; vii, 197 pp.; 5 figs. edition, 2008.

Tables

For a complete description of these tables or more details about results presented here, see the full paper [10] corresponding to this extended abstract.

$\log_2(q)$		F_5 (s)	Change of ordering (s)		Total (s)
			usual strategy	new strategy 1	
16	W. [11]	$>$ 2 days			
	T_2	12297	$>$ 2 days	7866	**20163**

Table 4: Computing time of Gröbner basis to solve the PDP with FGb on a 3.47 GHz Intel® Xeon® X5677 CPU for $n = 5$.

J.-C. Faugère PolSys project INRIA Paris-Rocquencourt; UPMC Paris 06; CNRS, UMR 7606; LIP6
jean-charles.faugere@inria.fr

P. Gaudry CARAMEL project INRIA Grand-Est; Université de Lorraine; CNRS, UMR 7503; LORIA
pierrick.gaudry@loria.fr

L. Huot PolSys project INRIA Paris-Rocquencourt; UPMC Paris 06; CNRS, UMR 7606; LIP6
louise.huot@lip6.fr

G. Renault PolSys project INRIA Paris-Rocquencourt; UPMC Paris 06; CNRS, UMR 7606; LIP6
guenael.renault@lip6.fr

New variants of algebraic attacks based on structured Gaussian elimination

Satrajit Ghosh and Abhijit Das

Introduction

In algebraic cryptanalysis, we express the encryption transform of a cipher as an overdefined system of multivariate polynomial equations in the bits of the plaintext, the ciphertext and the key, and then solve that system for the key bits from some known plaintext/ciphertext pairs. In general, solving such systems over finite fields is an NP-Complete problem. However, when the multivariate system is overdefined, some reasonable algorithms are known [1, 2, 3, 4, 5, 6, 7]. The XL_SGE algorithm [8] has been recently proposed to improve the complexity of the XL attack [4] by using structured Gaussian elimination (SGE) [9] during the expansion phase of XL. In this paper, we establish that XL_SGE suffers from some serious drawbacks. To avoid this problem, we propose three variants of XL_SGE, based upon partial monomial multiplication, handling of columns of weight two, and deletion of redundant equations. Our modified algorithms have been experimentally verified to be superior to XL_SGE.

We are given a sparse and consistent system $\mathbb{A}$ over $GF(2)$ of multivariate equations, some of which are quadratic and the rest of which are linear. Such systems are available from block ciphers like AES.

eXtended Linearization (XL)

In addition to the initial system $\mathbb{A}$, a degree bound D is also supplied as an input to XL [4].

Algorithm 1: Extended Linearization (XL) of multivariate systems

1. **Multiply:** Generate the new system $\mathbb{B} = \bigcup_{0\leq k\leq D-d_{max}} X^k\mathbb{A}$, where X^k stands for the set of all monomials of degree k, and d_{max} is the maximum degree of the initial system.
2. **Linearize:** Consider each monomial in the variables x_i of degree $\leq D$ as a new variable, and perform Gaussian elimination on the system $\mathbb{B}$. The ordering of the monomials must be such that all the terms containing single variables (like x_1) are eliminated last.
3. **Solve:** Assume that Step 2 yields at least one univariate polynomial equation in some variable x_1. Find the roots of this equation in the underlying finite field.
4. **Repeat:** Simplify the equations, and repeat the process to solve for the other variables.

Structured Gaussian Elimination (SGE)

Algorithm 2 describes one iteration of structured Gaussian elimination (SGE) [9].

Algorithm 2: Structured Gaussian Elimination (SGE)

1. Delete columns of weight 0 and 1.
2. Delete rows of weight 0 and 1.
3. Delete rows of weight 1 in the light part. After Step 2 and Step 3, update column weights.
4. Delete redundant rows.

A Heuristic Improvement of XL

The problem with XL is that the size of the system increases drastically with the increase in the degree bound D. Many linearly dependent equations are generated during the expansion process (Step 1) in XL. The equations generated by XL are very sparse. Moreover, the statistics of the systems obtained in XL (for $D = 2$) reveal that the columns of the generated systems can be distinguished as heavy-weight and light-weight. These observations lead us to propose a new heuristic (XL_SGE) [8] to reduce the number of linearized equations in XL. In XL_SGE, the intermediate systems are reduced using structured Gaussian elimination (SGE). The reduced systems are multiplied with monomials to get systems of higher algebraic degrees. XL_SGE uses only the first three steps of SGE.

Algorithm 3: Extended Linearization with Structured Gaussian Elimination (XL_SGE)

1. Expand the initial system $\mathbb{A}$ up to degree $d = 2$ using XL to obtain a linearized system $\mathbb{A}'$. Make a copy of the linearized system $\mathbb{A}'$ as $\mathbb{B}$.
2. Apply structured Gaussian elimination (SGE) on $\mathbb{A}'$ with avalanche-control parameter K to obtain a reduced system of equations $\mathbb{A}''$ of degree d.
3. Multiply each equation in $\mathbb{A}''$ by each monomial of degree 1 to get a system $\mathbb{A}'''$ of degree $d+1$. Append the equations of $\mathbb{A}'''$ to $\mathbb{B}$. $\mathbb{B}$ now has equations of degrees $\leq d+1$. Rename $\mathbb{A}'''$ as $\mathbb{A}'$.
4. If the degree of the system of equations $\mathbb{B}$ is D, end the process. Otherwise, go to Step 2 with d incremented by 1.

XL_SGE controls excessive reduction of intermediate systems due to avalanche effects by using a heuristic parameter K during the application of SGE. More specifically, the i-th row and the j-th column are eliminated if and only if the following three conditions are satisfied: (i) the j-th column has weight 1, (ii) the (i, j)-th entry is non-zero (1, to be precise), and (iii) the weight of the i-th row is at least K.

Improvements of XL_SGE

XL_SGE is designed to reduce the size of the final solvable system in comparison with XL. However, there are many instances where this size reduction is not substantial. Our experiments reveal that SGE on $\mathbb{A}'$ for $d = 2$ yields sizable reduction in the system size. Subsequently, for $d \geq 3$, SGE progressively loses effectiveness in bringing down the system size. This is the expected behavior of XL_SGE.

To ensure reduction of system sizes by SGE for all degrees of $\mathbb{A}'$, two possibilities are explored. First, we investigate how variables of column weight one may reappear in the system. Second, we modify SGE to work even when all variables have column weights ≥ 2.

- **Partial monomial multiplication:** Carefully skipping certain monomial multiplications during the expansion stage has some benefits. First, fewer equations are generated, and second, SGE may again discover variables of column weight one. On the darker side, generation of fewer equations may adversely affect the rank profile of the expanded system. If too many monomial multiplications are not skipped, we hope not to encounter a big trouble with the rank profile. Therefore, two important issues are of relevance in this context: which monomial multiplications would be skipped, and how many.

- **Deletion of variables with weight more than one:** Suppose that a variable z appears in $t \geq 2$ equations in an expanded system. If we add one of these equations to the remaining $t-1$ equations, the column weight of z reduces to one, so SGE (Algorithm 2) can remove this variable in Step 1. This, however, increases the weight of these $t-1$ equations. This increase in row weights may increase weights of certain columns. That is, an effort to forcibly eliminate z may stand in the way of the elimination of other variables. However, if $t = 2$, this processing of z

followed by the removal of the only equation containing z does not increase the total weight of the system. Still, the density (average weight per row or column) of the system increases (since one equation and one variable are now removed), but the expanded systems, particularly if large, are expected to absorb this problem without sufficient degradation of the performance of XL_SGE.

XL_SGE with Random Monomial Multiplication (XL_SGE-2)

As a first attempt, we skip monomial multiplications randomly, and the amount of skipping is governed by a probability $p \in (0,1]$. More precisely, each equation is multiplied by each monomial of degree one with probability p (and skipped with probability $1-p$). If $p=1$, we have the original XL_SGE algorithm. For $p<1$, we expect more size reduction than XL_SGE.

XL_SGE-2 accepts as input the initial system of equations $\mathbb{A}$, a degree bound $D \in \mathbb{N}$, the avalanche-control parameter $K \in \mathbb{N}$, and a multiplication probability $p \in (0,1]$.

Algorithm 4: XL_SGE with Random Monomial Multiplication (XL_SGE-2)

```
1. Expand the initial system $\mathbb{A}$ up to degree $d = 2$ using XL to obtain a linearized system $\mathbb{A}'$. Make
   a copy of the linearized system $\mathbb{A}'$ as $\mathbb{B}$.
2. Apply structured Gaussian elimination (SGE) on $\mathbb{A}'$ with avalanche-control parameter $K$ to obtain
   a reduced system of equations $\mathbb{A}''$ of degree $d$.
3. Multiply each equation in $\mathbb{A}''$ by each monomial of degree 1 with probability $p$ (that is, with
   probability $1-p$, a multiplication is skipped) to obtain a system $\mathbb{A}'''$ of degree $d+1$. Append the
   equations of $\mathbb{A}'''$ to $\mathbb{B}$. $\mathbb{B}$ now contains equations of degrees up to $d+1$. Rename $\mathbb{A}'''$ as $\mathbb{A}'$.
4. If the degree of the system of equations $\mathbb{B}$ is $D$, end the process. Otherwise, go to Step 2 with $d$
   incremented by 1.
```

If we get a full-rank (or close-to-full-rank) system for a particular D, we solve that system. Otherwise, we increase the degree bound D, and run XL_SGE-2 again to reduce the rank deficit.

The multiplication probability p has been heuristically chosen in our experiments. We have worked with several fixed values of p in different layers (degrees d of $\mathbb{A}'$). From our experimental experiences, we recommend values of $p \geq 0.5$. A slight modification in the above algorithm for XL_SGE-2 is also studied. In this variant, monomial multiplications are randomly skipped even in Step 1 (that is, since the very beginning of the expansion process).

Another possibility is to use different probabilities in different layers of multiplication. We study two models for varying p with the degree d of $\mathbb{A}'$. In the first model, we take $p_1 = 1 - \frac{1}{d+1}$. For this choice, we initially restrict the expansion of the system. If the initial restriction leads to large rank deficits, we progressively remove the restriction on the growth of the system. In the second model, we take the gradually decreasing sequence of probabilities $p_2 = \frac{D-d}{D-d+1}$. Initially, the system size is small, so we can afford the system to grow at this stage. As d increases, $\mathbb{A}'$ becomes increasingly large, and restricting the growth of the system gradually controls the eventual growth of the system. Note also that higher-degree monomials appear in the linearized system from a larger number of sources. Hence, more restriction in the growth is required to generate more variables with column weight one as d increases.

Column-weight Two Reduction

The original SGE procedure (Algorithm 2) can be modified so as to remove columns of weights two or more. In order that the rank profile of the expanded system does not deteriorate too much, we have experimented with deletion of columns of weight two only.

Algorithm 5: Structured Gaussian Elimination with Column-weight Two Reduction (SGE$'$)

```
1. Delete columns of weight 0 and 1.
2. Delete columns of weight 2: If a column has weight 2, delete one equation corresponding to that
   variable. Substitute that equation in the other equation, and delete the column.
```

3. Delete rows of weight 0 and 1.
4. Delete rows of weight 1 in the light part. After Steps 2–4, update column weights.

Although this heuristic modification of SGE seems to be effective, in the current form it does not work very well. One must not use Algorithm 5 to reduce the initial quadratic system (available after Step 1 of XL_SGE or XL_SGE-2), since random systems at this stage exhibit the tendency of losing all quadratic variables. Using the modified SGE for all $d \geq 3$ sometimes shows good performance. But the general observation is that the system suffers from drastic reduction in size (a form of avalanche effect) resulting in degraded rank profile and demanding a large number of iterations (that is, large values of D). It appears that the modified SGE procedure of Algorithm 5 should be skipped for certain small values of d (in addition to $d = 2$). However, the exact range of applicability of Algorithm 5 (that is, the minimum d from which it is safe to use this algorithm) has not yet been experimentally or theoretically determined. Such a study would require initial systems larger than what we have experimented with.

XL_SGE with Row Deletion (XL_SGE-3)

XL_SGE-2 demonstrates the benefits of using partial monomial multiplication. Instead of blindly skipping certain multiplications, we can adopt a more intelligent strategy. We first carry out all monomial multiplications. Subsequently, by analyzing the column statistics of the expanded system, we mark some equations as less important than the others. We delete the less important equations from the system and then perform SGE before the next stage of multiplication. This variant, henceforth referred to as XL_SGE-3, has one potential advantage over XL_SGE-2. Now, we have a better control over the initial reduction in the system size in the sense that the degradation of the rank profile can be carefully handled.

Algorithm 6: XL_SGE with Row Deletion (XL_SGE-3)

```
1. Expand the initial system $\mathbb{A}$ up to degree $d = 2$ using XL to obtain a linearized system $\mathbb{A}'$. Make
   a copy of the linearized system $\mathbb{A}'$ as $\mathbb{B}$.
2. Apply structured Gaussian elimination (SGE) with avalanche-control parameter $K$ on $\mathbb{A}'$ to obtain
   a reduced system of equations $\mathbb{A}''$ of degree $d$.
3. Multiply the reduced system $\mathbb{A}''$ with monomials of degree 1 and linearize the system to obtain a
   system $\mathbb{A}'''$ of degree $d+1$.
4. Identify and delete some rows of $\mathbb{A}'''$. Append the equations of $\mathbb{A}'''$ to $\mathbb{B}$. $\mathbb{B}$ now contains equa-
   tions of degrees up to $d+1$. Rename the system $\mathbb{A}'''$ as $\mathbb{A}'$.
5. If the degree of the system of equations $\mathbb{B}$ is $D$, end the process. Otherwise, go to step 2 after
   incrementing $d$ by 1
```

Depending upon how we identify the redundant rows for deletion in Step 4, we have different variants of XL_SGE-3, some of which are elaborated below. The deletion of redundant equations can also be employed after Step 1 of Algorithm 6.

XL_SGE-3 with Deterministic Deletion Strategy (XL_SGE-3d)

We have considered only the variables of column weight two. Among the two equations containing a variable with column weight two, we delete (at most) one equation as follows.

Strategy 1

- If any of these two equations contains a variable with column weight one, then skip the deletion of both the equations. (In this case, the equation with the variable with column weight one is anyway deleted by SGE, thereby reducing the weight of the variable with column weight two.)

- Otherwise, delete the equation with the larger row weight. If both the equations have the same row weight, delete any one of these arbitrarily.

Strategy 2

- If any of these two equations contains a variable with column weight one, then skip the deletion of both the equations.
- If both the equations have the same right side (0 or 1), delete the equation with the larger row weight. Make arbitrary choices to break ties.
- If exactly one of the two equations has right side 1, then keep that equation, and delete the other.

Strategy 3

- If any one of the equations contains a variable with column weight one, determine whether that variable can reappear in the system in a future monomial-multiplication stage. If not, none of the equations is deleted. Otherwise, delete the equation containing the variable with column weight one.
- If both the equations contain variables of column weight one that can reappear from a future monomial-multiplication stage, then delete one of them depending on their row weights (as in Strategy 1).
- If both the equations contain no variables of column weight one, then take decision as in Strategy 1.

Let $z = x_1x_2x_3$ be a monomial with column weight one, and let the equation containing z also contain a variable with column weight two. In Strategy 3, we check whether z can reappear in the next multiplication layer (like multiplication of x_1x_3 by x_2). If that is the case, the current rank degradation incurred by the deletion of the equation containing z will be repaired later.

XL_SGE-3 with Random Deletion Strategy (XL_SGE-3r)

Let z be a variable (monomial) with weight t. We delete m of the t equations in which z appears. If the system is overdefined, this deletion is not expected to have a bad effect on the rank profile. The details of this strategy are given below. In our experiments, we have worked with $t = 2$ and 3, and $m = 1$.

- Find an equation with a variable of column weight t.
- If the equation contains a variable of column weight one, skip the deletion.
- Otherwise, delete the equation with probability p_d.
- Repeat this process until there are no removable equations with variables of column weight t.

Experimental Results

We have experimented with several variants of XL_SGE on small random systems (Table 5), and also on the initial system of size 890 × 208 obtained from four-round baby-Rijndael (Table 6). XL_SGE-2 and XL_SGE-3 significantly improves the performance of XL and XL_SGE.

Conclusion

The chief technical contribution of this paper is our efforts to improve upon the XL family of algebraic attacks. We suggest variants of XL_SGE. Our experiments establish the effectiveness of using our modifications in tandem with XL_SGE. Our proposals address some of the open problems of XL_SGE, but some other issues continue to remain unattended. Most importantly, a theoretical analysis of the XL_SGE family is needed. Here, we state some new avenues for research, that this paper opens up.

Table 5: Performances of XL and variants of XL_SGE for random systems

	Size of $\mathbb{B}$				
Size of $\mathbb{A}$	XL	XL_SGE	XL_SGE-2	XL_SGE-3d	XL_SGE-3r
15×10	2712×637	2528×619	1447×631	1939×637	1360×637
16×11	2846×561	2119×561	943×561	1322×560	934×561
17×12	749×298	748×298	460×298	714×298	394×298
18×14	5347×1470	4796×1469	2199×1461	4356×1469	2462×1469
19×14	4831×1470	3620×1470	2333×1468	3447×1470	2414×1470
20×15	3783×1940	3963×1940	2907×1940	3149×1940	3073×1940
20×16	6402×2516	6094×2516	3700×2514	5407×2516	3994×2516
23×18	117996×31179	122701×31175	86200×31175	112307×31172	85227×31179

Table 6: Performances of XL and variants of XL_SGE for four-round baby-Rijndael ($D = 3$).

Algorithm	K	p	p_d	Size of $\mathbb{B}$	Rank Deficit δ
XL	0	1	0	2594060×1498713	96936
XL_SGE	3	1	0	2571848×1476481	93172
XL_SGE-2	0	0.75	0	2276971×1442363	89387
XL_SGE$'$	0	1	0	2556116×1449153	81576
XL_SGE-3d	0	1	0	1934149×1163740	79630
XL_SGE-3r	0	1	0.20	2355165×1449152	85470
XL_SGE-3r	0	1	0.25	2283125×1449152	89640

- The domains of applicability of XL_SGE$'$ need to be experimentally or theoretically determined.
- The dependence of the system size and rank profile on the seed (multiplication/deletion decisions) for XL_SGE-2 and XL_SGE-3r should be studied.
- An optimal choice for p (in XL_SGE-2) and p_d (in XL_SGE-3r) requires more experimentation and theoretical analysis.

References

[1] J. C. Faugére, A new efficient algorithm for computing Gröbner basis (F4), 2000.

[2] J. C. Faugére, A new efficient algorithm for computing Gröbner basis without reduction to zero (F5), ISSAC ’02, pp. 75–83, 2002.

[3] A. Kipnis and A. Shamir, Cryptanalysis of the HFE public key cryptosystem by relinearization, in CRYPTO, pp. 19–30, 1999.

[4] N. Courtois , A. Klimov, J. Patarin and A. Shamir, Efficient algorithms for solving overdefined systems of multivariate polynomial equations, in EUROCRYPT, pp. 392–407, 2000.

[5] N. Courtois and J. Pieprzyk, Cryptanalysis of block ciphers with overdefined systems of equations, in ASIACRYPT, pp. 267–287, 2002.

[6] J. Ding, J. Buchmann, M. Mohamed, W. Mohamed and R. Weinmann, MutantXL, in SCC, pp. 16–22, 2008.

[7] G. Bard, N. Courtois and C. Jefferson, Solution of sparse polynomial systems over GF(2) via sat-solvers, in ECRYPT workshop Tools for Cryptanalysis, 2007.

[8] S. Ghosh and A. Das, An improvement of linearization-based algebraic attacks, in InfoSecHiComNet, pp. 157–167, 2011.

[9] B. LaMacchia and A. Odlyzko, Solving large sparse linear systems over finite fields, in CRYPTO, pp. 109–133, 1991.

Satrajit Ghosh Indian Institute of Technology Kharagpur
satrajit@cse.iitkgp.ernet.in
Abhijit Das Indian Institute of Technology Kharagpur
abhij@cse.iitkgp.ernet.in

The multivariate probabilistic encryption scheme MQQ-ENC

Danilo Gligoroski and Simona Samardjiska

We propose a new multivariate probabilistic encryption scheme with decryption errors MQQ-ENC that belongs to the family of MQQ-based public key schemes. Similarly to MQQ-SIG, the trapdoor is constructed using quasigroup string transformations with multivariate quadratic quasigroups, and a minus modifier with relatively small and fixed number of removed equations. To make the decryption possible and also efficient, we use a universal hash function to eliminate possibly wrong plaintext candidates. We show that, in this way, the probability of erroneous decryption becomes negligible.

MQQ-ENC is defined over the fields $\mathbb{F}_{2^k}$ for any $k \geq 1$, and can easily be extended to any $\mathbb{F}_{p^k}$, for prime p. One important difference from MQQ-SIG is that in MQQ-ENC we use left MQQs (LMQQs) instead of bilinear MQQs. Our choice can be justified by our extensive experimental analysis that showed the superiority of the LMQQs over the bilinear MQQs for the design of MQQ-ENC.

We apply the standard cryptanalytic techniques on MQQ-ENC, and from the results, we pose a plausible conjecture that the instances of the MQQ-ENC trapdoor are hard instances with respect to the MQ problem. Under this assumption, we adapt the Kobara-Imai conversion of the McEliece scheme for MQQ-ENC and prove that it provides $\mathsf{IND}\text{--}\mathsf{CCA}$ security despite the negligible probability of decryption errors.

We also recommend concrete parameters for MQQ-ENC for encryption of blocks of 128 bits for a security level of $\mathcal{O}(2^{128})$.

D. Gligoroski Norwegian University of Science and Technology
danilog@item.ntnu.no
S. Samardjiska Norwegian University of Science and Technology
simonas@item.ntnu.no

Edwards curves with large torsion subgroups over number fields

Dawu Gu, Haihua Gu, and Wenlu Xie

Abstract

Edwards curves allow faster scalar multiplication than all other known curve shapes. This implies speed improvement for many applications in cryptography and number theory. Bernstein et al. suggested to use Edwards curves instead of Montgomery curves or Weierstrass curves in the elliptic curve method (ECM).

In this paper, we gave infinitely Edwards curves with a large torsion subgroup over number fields. These curves are more efficient for ECM when factoring numbers from the Cunningham project.

Introduction

Integer factorization is one of the well-studied problems in algorithmic number theory and cryptology. Elliptic curve method (ECM) is an integer factorization algorithm, which is invented by H.W. Lenstra [6] in 1987. It is a generalization of Pollard's $p-1$ algorithm. The idea is to estimate scalar multiplication $d \cdot P$ on elliptic curves over the ring $\mathbb{Z}/n\mathbb{Z}$. Although n is not prime, computations are done as if we were working on a field. If something fails, a non-trivial factor of n can be found. ECM is one of the fastest algorithms for integers with 10-80 digits. And it is often used in the number field sieve which is the most efficient factorization algorithm for integers used in cryptography. ECM can also be used to factor Cunningham numbers. They are of the form $a^m \pm 1$, where a and m are integers and a is not already a power of some other number.

Traditionally, Weierstrass curves or Montgomery curves is used in ECM. In 2008, Bernstein et al. [2] adapted ECM using Edwards curves. To improve the efficiency, Bernstein et al. generated Edwards curves with a large torsion subgroup over $\mathbb{Q}$. Recently, Brier and Clavier [3] shows that for Cunningham integers, curves with a large torsion subgroup over small extension of $\mathbb{Q}$ is better.

The aim of this paper is to generate Edwards curves with a large torsion subgroup over small extension of $\mathbb{Q}$.

Background

A number field is a finite algebraic extension of $\mathbb{Q}$. An elliptic curve E defined over a number field K turns out to be a commutative group. The Mordell-Weil theorem states that this group is finitely generated and can be written as

$$E(K) \cong \mathcal{T} \otimes \mathbb{Z},$$

where the integer r is called rank and $\mathcal{T}$ is called torsion group, which consists in elements of finite order. Furthermore, $\mathcal{T}$ is isomorphic to $\mathbb{Z}/m\mathbb{Z} \times Z/n\mathbb{Z}$ with the constraints that m divides n and the $n-$th roots of unity all lie in the field K. If K is the rational number field $\mathbb{Q}$, the order of $\mathcal{T}$ is less than or equal to 16. If K is the quadratic extension of $\mathbb{Q}$, the order of $\mathcal{T}$ is less than or equal to 24, and it is not more than 36 when K is the quartic extension of $\mathbb{Q}$.

Lemma 1. [4, P. 308] Let $E(K)$ be an elliptic curve

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

This work was supported by National Natural Science Foundation of China (No. 61073150).

The maps $x \mapsto u^2x' + r$ and $y \mapsto u^3y' + u^2sx' + t$ with $u, r, s, t \in K$ and $u \neq 0$ are invertible and transform the curve $E(K)$ into

$$E'(K) : y'^2 + a'_1x'y' + a'_3y' = x'^3 + a'_2x'^2 + a'_4x' + a'_6,$$

where the a'_i belongs to K and can be expressed in terms of a_i, u, r, s, t.

Lemma 2. [7] A Weierstrass-form elliptic curve $E : y^2 = x^3 + ax + b$ is transformable to the Montgomery-form if and only if it satisfies two conditions as follows:
1. The equation $x^3 + ax + b = 0$ has at least one root in $\mathbb{F}_p$
2. The number $3\alpha^2 + a$ is the quadratic residue in $\mathbb{F}_p$, where α is a root of the equation $x^3 + ax + b = 0$ in $\mathbb{F}_p$.

Note that this lemma considers finite fields $\mathbb{F}_p$, and it is also true for number fields. Assume an elliptic curve E satisfies such conditions. Let $s = 1/\sqrt{3\alpha^2 + a}$, then E can be mapped to the Montgomery-form curve $E_{M,A,B} : By^2 = x^3 + Ax^2 + x$ by $(x, y) \mapsto (s(x - \alpha), sy)$, where $B = s$ and $A = 3\alpha s$.

Lemma 3. [1] Fix a field K with $\mathrm{char}(K) \neq 2$.
1. Fix $A \in K \setminus \{-2, 2\}$ and $B \in K \setminus \{0\}$. The Montgomery curve $E_{M,A,B}$ is birationally equivalent to the twisted Edwards curve $E_{E,a,d}$, where $a = (A + 2)/B$ and $d = (A - 2)/B$. The map $(u, v) \mapsto (x, y) = (u/v, (u - 1)/(u + 1))$ is birational equivalence from $E_{M,A,B}$ to $E_{E,a,d}$;
2. Fix distinct nonzero elements $a, d \in K$. The twisted Edwards curve $E_{E,a,d}$ is birationally equivalent to the Montgomery curve $E_{M,A,B}$ where $A = 2(a + d)/(a - d)$ and $B = 4/(a - d)$. The map $(x, y) \mapsto (u, v) = ((1 + y)/(1 - y), (1 + y)/(1 - y)x)$ is birational equivalence from $E_{E,a,d}$ to $E_{M,A,B}$.

Computations in extended Edwards coordinate would benefit from using twisted Edwards curves with $a = -1$. If a is a square in K , the twisted Edwards curve $E_{E,a,d}$ is isomorphic to $E_{E,1,d/a}$: $x^2 + y^2 = 1 + (d/a)x^2y^2$ over K. The isomorphism is $(x, y) \mapsto (\sqrt{a}x, y)$.

Twisted Edwards curves

In this section, we prove the following four results.

Theorem 4. *The twisted Edwards curves $-x^2 + y^2 = 1 + dx^2y^2$ with $d = -\frac{9}{144v^2}$ have a torsion group which is isomorphic to $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}(\sqrt{-1}, \sqrt{t^4 - 6t^2 + 1})$, where $v = \frac{t^4-6t^2+1}{4(t^2+1)^2}$, $t \in \mathbb{Q}$ and $t \neq 0, \pm 1$.*

Proof. Jeon et.al. [5] constructed infinitely Weierstrass curves

$$y^2 + xy - (v^2 - \frac{1}{16})y = x^3 - (v^2 - \frac{1}{16})x^2, \tag{1}$$

have a torsion group which is isomorphic to $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}(\sqrt{-1}, \sqrt{t^4 - 6t^2 + 1})$, where $v = \frac{t^4-6t^2+1}{4(t^2+1)^2}$ and $t \neq 0, \pm 1$.

Since we don't find maps which can transform these Weierstrass curves directly to Edwards curves, the curves are first transformed to $y^2 = x^3 + ax + b$, then mapped to Montgomery curves, and converted to Edwards curves at last. Thanks to the software named Sage [8], we can do these symbol computations easily. By Lemma 1 and Lemma 2, curves in Eq. (1) can be transformed to the following Montgomery curves $E_{M,A,B} : By^2 = x^3 + Ax^2 + x$, where $A = \frac{2(16v^2+1)}{16v^2-1}$, $B = \frac{4}{9(16v^2-1)}$. Lemma 3 says that the Montgomery curves are birational to the twist Edwards curves $ax^2 + y^2 = 1 + dx^2y^2$ with

$$\begin{cases} \frac{2(a+d)}{(a-d)} = A \\ \frac{4}{(a-d)} = B. \end{cases}$$

One can get that $d=9$ and $a=144v^2$. Fortunately, a is a square and -1 is also a square in $\mathbb{Q}(\sqrt{-1})$, so the twist Edwards curves $E_{E,a,d}: ax^2+y^2=1+dx^2y^2$ is isomorphic to $E_{E,-1,-d/a}: -x^2+y^2=1-\frac{d}{a}x^2y^2$ over $\mathbb{Q}(\sqrt{-1},\sqrt{t^4-6t^2+1})$. The theorem follows. □

This theorem implies that we have generated infinitely Edwards curves with a large torsion subgroup over the quartic extension of $\mathbb{Q}$.

Theorem 5. *The twisted Edwards curves $-x^2+y^2=1+dx^2y^2$ with $d=-\frac{(v^2+2v+3)^4}{(v^2-2v+3)^4}$ where $v\in\mathbb{Q}$ and $v\neq 1,3$ have a torsion group which is isomorphic to $\mathbb{Z}/4\mathbb{Z}\times\mathbb{Z}/4\mathbb{Z}$ over $\mathbb{Q}(i)$.*

Proof. Brier and Clavier [3] constructed infinitely Weierstrass curves

$$y^2=x^3+ax+b \tag{2}$$

with positive rank and torsion subgroup $\mathbb{Z}/4\mathbb{Z}\times\mathbb{Z}/4\mathbb{Z}$ over $\mathbb{Q}(i)$, where
$a=-432v^4(v^{16}+24v^{14}+476v^{12}+4200v^{10}+18022v^8+37800v^6+38556v^4+17496v^2+6561)$,
$b=3456v^6(v^{24}+36v^{22}+66v^{20}-6732v^{18}-101409v^{16}-707256v^{14}-2772260v^{12}-6365304v^{10}-8214129v^8-4907628v^6+433026v^4+2125764v^2+531441)$ and $v\in\mathbb{Z}$.

One can transform the above curves to the Montgomery curves

$$By^2=x^3+Ax^2+x, \tag{3}$$

where $B=1/[2^4\cdot 3^2\cdot(v^2+1)\cdot(v^2+3)\cdot(v^2+9)\cdot v^3]$; $A=36\cdot(v^{10}+36\cdot v^8+214\cdot v^6+324\cdot v^4+81\cdot v^2)\cdot B$. The Montgomery curves are birational to the twist Edwards curves $ax^2+y^2=1+dx^2y^2$. So we let

$$\begin{cases}\frac{2(a+d)}{(a-d)}=A\\ \frac{4}{(a-d)}=B.\end{cases}$$

It follows that

$$\begin{aligned} a &= 36\cdot(v^2+2\cdot v+3)^4\cdot v^2;\\ d &= 36\cdot(v^2-2\cdot v+3)^4\cdot v^2.\end{aligned}$$

Fortunately, a is a square and -1 is also a square in $Q(i)$, so the twist Edwards curves $E_{E,a,d}$: $ax^2+y^2=1+dx^2y^2$ is isomorphic to $E_{E,-1,-d/a}: -x^2+y^2=1-\frac{d}{a}x^2y^2$ over $Q(i)$.

□

Bernstein et.al. [2] showed that twisted Edwards curves with $a=-1$ can't have a torsion group which is isomorphic to $\mathbb{Z}/12\mathbb{Z}$ or $\mathbb{Z}/2\mathbb{Z}\times\mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}$. Now we will show that they have the torsion group which is isomorphic to $\mathbb{Z}/12\mathbb{Z}$ or $\mathbb{Z}/2\mathbb{Z}\times\mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}(i)$.

Theorem 6. *There exist twisted Edwards curves of the form $ax^2+y^2=1+dx^2y^2$ with $a=-1$ and have a torsion group which is isomorphic to $\mathbb{Z}/2\mathbb{Z}\times\mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}(i)$.*

Proof. Theorem 6.9 of [2] shows that if $u\in\mathbb{Q}\setminus\{0,-1,-2\}$, $x_8=\frac{u^2+2u+2}{u^2-2}$ and $d=\frac{2x_8^2-1}{x_8^4}$ then the Edwards curve $x^2+y^2=1+dx^2y^2$ has a torsion group which is isomorphic to $\mathbb{Z}/2\mathbb{Z}\times\mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}$. Since $a=-1$ is a square in $\mathbb{Q}(i)$, it follows that $x^2+y^2=1+dx^2y^2$ is isomorphic to $-x^2+y^2=1-dx^2y^2$ over $\mathbb{Q}(i)$. This implies $-x^2+y^2=1-\frac{2x_8^2-1}{x_8^4}x^2y^2$ has a torsion group which is isomorphic to $\mathbb{Z}/2\mathbb{Z}\times\mathbb{Z}/8\mathbb{Z}$ over $\mathbb{Q}(i)$. □

Using the same method, we can prove the following result.

Corollary 7. *There exist twisted Edwards curves of the form $ax^2+y^2=1+dx^2y^2$ with $a=-1$ and have a torsion group which is isomorphic to $\mathbb{Z}/12\mathbb{Z}$ over $\mathbb{Q}(i)$.*

Conclusion

In this paper, we formed infinitely Edwards curves with a large torsion subgroup over number fields.

References

[1] D.J.Bernstein, P.Birkner, M.Joye, et al. Twisted Edwards Curves. In: *AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Comput. Sci.*, pages 389-405,Springer, 2008.

[2] D.J.Bernstein, P.Birkner, T.Lange, et al. ECM using Edwards curves. Accepted by *Math. Comp.*, Cryptology ePrint Archive, Report 2008/016, 2008.

[3] É.Brierl, C.Clavier. New Families of ECM Curves for Cunningham Numbers. In: *Proceedings of ANTS IX*, volume 6197 of *Lecture Notes in Comput. Sci.*, pages 96-109,Springer, 2010.

[4] H.Cohen, G. Frey, R. Avanzi, et al. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Discrete Mathematics and its applications, Chapman & Hall/CRC, Boca Raton, FL, 2006.

[5] D.Jeon, C.H.Kim and Y.Lee. Families of elliptic curves over quartic number fields with prescribed torsion subgroups. *Math. Comp.* 80: 579-591,2011.

[6] H.W.Lenstra. Factoring integers with elliptic curves. *Ann. Math.* 126: 649-673, 1987.

[7] K.Okeya, H.Kurumatani and K.Sakurai. Elliptic Curves with the Montgomery-Form and Their Cryptographic Applications. In: *PKC 2000*, volume 1751 of *Lecture Notes in Comput. Sci.*, pages 238-257,Springer, 2000.

[8] W.A.Stein, et al. Sage Mathematics Software (Version 4.7), The Sage Development Team, 2011. http://www.sagemath.org.

Dawu Gu Shanghai Jiao Tong University
dwgu@sjtu.edu.cn
Haihua Gu Shanghai Jiao Tong University
guhaihua@shhic.com
Wenlu Xie Shanghai Huahong Integrated Circuit Co.,Ltd.

An algebraic fault attack on the LED Block Cipher

P. Jovanovic, M. Kreuzer, and I. Polian

Introduction

Immunity to conventional cryptanalysis has been formally proven for a number of ciphers. Newly developed ciphers are expected to be resistant against known cryptanalytic methods. For this reason, *fault-based cryptanalysis* [5] is receiving increasing attention [9, 10, 13, 16, 19]. In fault-based cryptanalysis, the attacker targets the hardware implementation of a cryptographic algorithm rather than the algorithm itself. The attacker performs a *fault injection* into the electronic circuit and manipulates the logical values being processed by the circuit. A variety of fault-injection techniques has been discussed [2]. For instance, the attacker may reduce the power-supply voltage of the circuit, causing the logic gates within the circuit to switch slower; as a consequence, wrong values will be calculated. A different technique is irradiating a desired location in the circuit (a logic gate performing some calculation or a register holding an intermediate value) using a laser. The laser pulse will induce parasitic currents and ultimately flip the logical value of the targeted location from logic-0 to logic-1 or vice versa.

Typically, the attacker will run the cryptographic algorithms multiple times, with and without fault injection, and will perform differential cryptanalysis on the outcomes (see [3]). Obviously, fault-based attacks are easier if the attacker can accurately control which logic structure is manipulated and what new value it assumes. In reality, the effectiveness of a fault-based attack may suffer if the attacker has only limited control over the location and/or the exact time (calculation step) of the fault injection. For example, the laser may have a precision that is sufficient to target a register but not sufficient to target individual memory cells within the register. In this case, the register's value will be modified, but to an unknown value. Therefore, a fault-based attack is always defined with respect to an assumption on the attacker's technical capabilities.

We recently introduced a fault-based attack [12] on the new LED block cipher [7]. The LED encryption scheme is conceptually similar to AES [17] but belongs to the family of lightweight block ciphers [4, 8], which are developed for usage in low-cost, power-constrained systems, and are typically employed in mobile, embedded and ubiquitous contexts. Those ciphers carefully balance cryptographic strength against resource requirements, most importantly power consumption. We were able to break LED using one fault injection under weak assumptions on the resolution of the equipment. Our attack yielded a reduced set of key candidates which was feasible for brute force enumeration.

Recently, a new idea originated in [18], namely to enhance algebraic attacks by information obtained through side-channel cryptanalysis. This idea was further developed in [6] and used in [15] to attack the stream cipher Trivium. In this paper, we exploit this idea by combining the previously mentioned fault-based attack on the LED block cipher with a more traditional algebraic attack. The paper is organized as follows.

In the next section we describe the 64-bit and 128-bit versions of the LED cipher and provide a complete algebraic description of the encryption map. After that we recall in Section the fault attack from [12] and discuss the transformation of the fault equations to fault polynomials. Finally, Section containing the actual attack and experimental results showing its practical feasibility finishes the paper.

Unless specifically stated otherwise, we will use the terminology and notation introduced in [14].

Algebraic Representation of the `LED` Block Cipher

In this section we show how to construct the polynomial representation of the `LED` cipher [7]. It will be contained in $\mathbb{F}_2[p_i, k_i, x_i^{(r)}, y_i^{(r)} z_i^{(r)}, c_i \mid i = 1, \ldots, 64; r = 1, \ldots, 32]$, a polynomial ring having no less than 6336 indeterminates.

AddConstants (AC).

To represent this operation by polynomials, we distinguish two cases: round number $r = 1$ and round numbers $r > 1$. In the first case we model the input whitening and the first application of `AC` in one step. Since the first round constants vector is $(b_5, b_4, b_3, b_2, b_1, b_0) = (0,0,0,0,0,1)$, we get

$$\begin{aligned} x_i^{(1)} &= p_i + k_i + 1 \quad && \text{for } i \in \{20, 24, 35, 51, 52, 56\}, \\ x_i^{(1)} &= p_i + k_i && \text{otherwise.} \end{aligned}$$

Here the indeterminates $x_i^{(1)}$ describe the state after the first application of `AC`. Similarly, let $x_i^{(r)}$ describe the state after the r-th application of `AC`, for $r = 2, \ldots, 32$, and let $z_i^{(r)}$ denote the state of the cipher after the application of `MSC` in round r.

For the case $r > 1$, let $(b_5^{(r)}, b_4^{(r)}, b_3^{(r)}, b_2^{(r)}, b_1^{(r)}, b_0^{(r)})$ be the r-th round constants vector, then we get

$$\begin{aligned} x_i^{(r)} &= z_i^{(r-1)} + b_5^{(r)} \ \text{ for } i \in \{6, 38\} \qquad & x_i^{(r)} &= z_i^{(r-1)} + b_4^{(r)} \ \text{ for } i \in \{7, 39\} \\ x_i^{(r)} &= z_i^{(r-1)} + b_3^{(r)} \ \text{ for } i \in \{8, 40\} & x_i^{(r)} &= z_i^{(r-1)} + b_2^{(r)} \ \text{ for } i \in \{22, 54\} \\ x_i^{(r)} &= z_i^{(r-1)} + b_1^{(r)} \ \text{ for } i \in \{23, 55\} & x_i^{(r)} &= z_i^{(r-1)} + b_0^{(r)} \ \text{ for } i \in \{24, 56\} \\ x_i^{(r)} &= z_i^{(r-1)} + 1 \quad \text{ for } i \in \{20, 35, 51, 52\} & x_i^{(r)} &= z_i^{(r-1)} \qquad\quad \text{otherwise} \end{aligned}$$

in rounds whose round number r is not divisible by four, and the same equations plus a keybit addition every fourth round.

SubCells (SC) and ShiftRows (SR). The `ShiftRows` permutation can be described by

$$\begin{aligned} \sigma = {} & (17\ 29\ 25\ 21)(18\ 30\ 26\ 22)(19\ 31\ 27\ 23)(20\ 32\ 28\ 24) \\ & (33\ 41)(34\ 42)(35\ 43)(36\ 44)(37\ 45)(38\ 46)(39\ 47)(40\ 48) \\ & (49\ 53\ 57\ 61)(50\ 54\ 58\ 62)(51\ 55\ 59\ 63)(52\ 56\ 60\ 64) \end{aligned}$$

Now we model the combined effect of `SubCells` and `ShiftRows`. Let $i_1 = 4i - 3$, $i_2 = 4i - 2$, $i_3 = 4i - 1$ and $i_4 = 4i$ for $i = 1, \ldots, 16$. Then, in round r, we get the following four equations.

$$\begin{aligned} y_{\sigma(i_1)}^{(r)} = {} & x_{i_1}^{(r)} x_{i_2}^{(r)} x_{i_4}^{(r)} + x_{i_1}^{(r)} x_{i_3}^{(r)} x_{i_4}^{(r)} + x_{i_2}^{(r)} x_{i_3}^{(r)} x_{i_4}^{(r)} + \\ & x_{i_2}^{(r)} x_{i_3}^{(r)} + x_{i_1}^{(r)} + x_{i_3}^{(r)} + x_{i_4}^{(r)} + 1 \\ y_{\sigma(i_2)}^{(r)} = {} & x_{i_1}^{(r)} x_{i_2}^{(r)} x_{i_4}^{(r)} + x_{i_1}^{(r)} x_{i_3}^{(r)} x_{i_4}^{(r)} + x_{i_1}^{(r)} x_{i_3}^{(r)} + \\ & x_{i_1}^{(r)} x_{i_4}^{(r)} + x_{i_3}^{(r)} x_{i_4}^{(r)} + x_{i_1}^{(r)} + x_{i_2}^{(r)} + 1 \\ y_{\sigma(i_3)}^{(r)} = {} & x_{i_1}^{(r)} x_{i_2}^{(r)} x_{i_4}^{(r)} + x_{i_1}^{(r)} x_{i_3}^{(r)} x_{i_4}^{(r)} + x_{i_2}^{(r)} x_{i_3}^{(r)} x_{i_4}^{(r)} + \\ & x_{i_1}^{(r)} x_{i_2}^{(r)} + x_{i_1}^{(r)} x_{i_3}^{(r)} + x_{i_1}^{(r)} + x_{i_3}^{(r)} \\ y_{\sigma(i_4)}^{(r)} = {} & x_{i_2}^{(r)} x_{i_3}^{(r)} + x_{i_1}^{(r)} + x_{i_2}^{(r)} + x_{i_4}^{(r)} \end{aligned}$$

MixColumnsSerial (MCS).

Let $y_1^{(r)} \parallel \cdots \parallel y_{64}^{(r)}$ be the state of the cipher after `ShiftRows` has been executed in round r, and let $z_1^{(r)} \parallel \cdots \parallel z_{64}^{(r)}$ be its state after `MCS`. The entries of the state matrix are the field elements

$y_{4i-3}^{(r)}x^3 + y_{4i-2}^{(r)}x^2 + y_{4i-1}^{(r)}x + y_{4i}^{(r)}$ of $\mathbb{F}_{16}$. Then the `MCS` operation can be described by the following equations

$$
\begin{aligned}
z_{j_1}^{(r)} &= y_{j_3}^{(r)} + y_{j_5}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{14}}^{(r)}\\
z_{j_2}^{(r)} &= y_{j_1}^{(r)} + y_{j_4}^{(r)} + y_{j_6}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{15}}^{(r)}\\
z_{j_3}^{(r)} &= y_{j_1}^{(r)} + y_{j_2}^{(r)} + y_{j_7}^{(r)} + y_{j_9}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{13}}^{(r)} + y_{j_{16}}^{(r)}\\
z_{j_4}^{(r)} &= y_{j_2}^{(r)} + y_{j_8}^{(r)} + y_{j_9}^{(r)} + y_{j_{13}}^{(r)}\\
z_{j_5}^{(r)} &= y_{j_1}^{(r)} + y_{j_4}^{(r)} + y_{j_6}^{(r)} + y_{j_7}^{(r)} + y_{j_9}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{14}}^{(r)} + y_{j_{15}}^{(r)}\\
z_{j_6}^{(r)} &= y_{j_1}^{(r)} + y_{j_2}^{(r)} + y_{j_5}^{(r)} + y_{j_7}^{(r)} + y_{j_8}^{(r)} + y_{j_9}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{13}}^{(r)} + y_{j_{15}}^{(r)} + y_{j_{16}}^{(r)}\\
z_{j_7}^{(r)} &= y_{j_2}^{(r)} + y_{j_3}^{(r)} + y_{j_6}^{(r)} + y_{j_8}^{(r)} + y_{j_9}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{14}}^{(r)} + y_{j_{16}}^{(r)}\\
z_{j_8}^{(r)} &= y_{j_3}^{(r)} + y_{j_5}^{(r)} + y_{j_6}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{13}}^{(r)} + y_{j_{14}}^{(r)}
\end{aligned}
$$

$$
\begin{aligned}
z_{j_9}^{(r)} &= y_{j_2}^{(r)} + y_{j_4}^{(r)} + y_{j_5}^{(r)} + y_{j_6}^{(r)} + y_{j_7}^{(r)} + y_{j_8}^{(r)} + y_{j_9}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{16}}^{(r)}\\
z_{j_{10}}^{(r)} &= y_{j_1}^{(r)} + y_{j_3}^{(r)} + y_{j_6}^{(r)} + y_{j_7}^{(r)} + y_{j_8}^{(r)} + y_{j_9}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{13}}^{(r)}\\
z_{j_{11}}^{(r)} &= y_{j_1}^{(r)} + y_{j_2}^{(r)} + y_{j_4}^{(r)} + y_{j_7}^{(r)} + y_{j_8}^{(r)} + y_{j_9}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{14}}^{(r)}\\
z_{j_{12}}^{(r)} &= y_{j_1}^{(r)} + y_{j_3}^{(r)} + y_{j_4}^{(r)} + y_{j_5}^{(r)} + y_{j_6}^{(r)} + y_{j_7}^{(r)} + y_{j_9}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{15}}^{(r)} + Y_{j_{16}}^{(r)}\\
z_{j_{13}}^{(r)} &= y_{j_2}^{(r)} + y_{j_6}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{14}}^{(r)} + y_{j_{16}}^{(r)}\\
z_{j_{14}}^{(r)} &= y_{j_3}^{(r)} + y_{j_7}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{13}}^{(r)} + y_{j_{15}}^{(r)}\\
z_{j_{15}}^{(r)} &= y_{j_1}^{(r)} + y_{j_4}^{(r)} + y_{j_5}^{(r)} + y_{j_8}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{13}}^{(r)} + y_{j_{14}}^{(r)} + y_{j_{16}}^{(r)}\\
z_{j_{16}}^{(r)} &= y_{j_1}^{(r)} + y_{j_5}^{(r)} + y_{j_9}^{(r)} + y_{j_{10}}^{(r)} + y_{j_{11}}^{(r)} + y_{j_{12}}^{(r)} + y_{j_{13}}^{(r)} + y_{j_{15}}^{(r)} + y_{j_{16}}^{(r)}
\end{aligned}
$$

where $i \in \{1,2,3,4\}$ and $j_k = 4i - 4 + k$, $j_{4+k} = 4i + 12 + k$, $j_{8+k} = 4i + 28 + k$, and $j_{12+k} = 4i + 44 + k$ for $k = 1,2,3,4$.

Final Key Addition. For $i = 1,\dots,64$, the equations $c_i = z_i^{(32)} + k_i$ describe the final key addition and finish the algebraic representation of the `LED-64` block cipher. It is clear that `LED-128` has a similar description, using additional indeterminates for the second key and the extra rounds.

Algebraic Representation of the Fault Equations

The algebraic representation of `LED-64` constructed above is not suitable to launch a successful algebraic attack. It involves too many non-linear equations in too many indeterminates. To reconstruct the secret key from given (correct or faulty) plaintext – ciphertext pairs requires additional information. This information will be furnished by a fault attack. In [12] we discussed a method for injecting fault and using it to break `LED-64` by exhaustive search. In the following, we construct a polynomial version of the fault equations which were generated there.

Let us recall the description of the attack. We assume the following fault model. The attacker is supposed to be able to encrypt the same plain text unit twice using the same secret key k. The first encryption takes place correctly, and during the second encryption a fault is introduced. The fault is a random change in the value of the first (4-bit sized) entry of the state matrix at the beginning of round 30. As a consequence, we obtain a correct ciphertext c and a faulty ciphertext c'. The propagation of the fault is observed. It leads to an incorrect first column of the state matrix after the SBox has been applied in round 31 whose 4-bit entries we denote by a,b,c,d. In [12] we derived 16 fault equations containing, besides a,b,c,d, the indeterminates $k_1,\dots,k_{16}$, which represent the 4-bit parts of the secret key, the indeterminates $\bar{c}_1,\dots,\bar{c}_{16}$, which represent the parts of the correct ciphertext, and $\bar{c}'_1,\dots,\bar{c}'_{16}$ the parts of the faulty ciphertext. Since these equations involve the map $S^{-1} : \mathbb{F}_{16} \longrightarrow \mathbb{F}_{16}$ (the inverse SBox), we need to find a polynomial representation of this map.

Using univariate interpolation, we construct the following polynomial representation of S^{-1}.

$$\begin{aligned} S^{-1}(y) = {} & (x^2+1)+(x^2+1)y+(x^3+x)y^2+(x^3+x^2+1)y^3+xy^4+ \\ & (x^3+1)y^5+(x^3+1)y^7+(x+1)y^9+(x^2+1)y^{10}+(x^3+1)y^{11}+ \\ & (x^3+x)y^{12}+(x+1)y^{13}+(x^3+x^2+1)y^{14} \end{aligned}$$

Next, we plug the right-hand sides of the fault equations into this polynomial. We get 16 polynomial fault equations which are defined over the polynomial ring $\mathbb{F}_{16}[a,b,c,d,\bar{k}_1,\dots,\bar{k}_{16},\bar{c}_1,\dots,\bar{c}_{16},\bar{c}'_1,\dots,\bar{c}'_{16}]$. For every group of equations $E_{t,0},E_{t,1},E_{t,2},E_{t,3}$ having the same left-hand side $t \in \{a,b,c,d\}$, we can form three differences $E_{t,0}-E_{t,i}=0$ with $i=1,2,3$. Now, comparing coefficients for $\{1,x,x^2,x^3\}$ yields 48 equations in the bits $k_1,\dots,k_{64}$ of the secret key, the bits $c_1,\dots,c_{64}$ of the correct ciphertext, and the bits $c'_1,\dots,c'_{64}$ of the faulty ciphertext. Notice that we can use the field equations $k_i^2+k_i=0$, $c_i^2+c_i=0$, and $(c'_i)^2+c'_i=0$ for simplification here.

Altogether, we find 48 polynomials in $\mathbb{F}_2[k_1,\dots,k_{64},c_1,\dots,c_{64},c'_1,\dots,c'_{64}]$. They all have degree 3 and consist of approximately 3400-8800 terms. These polynomials will be called the **fault polynomials**.

An Algebraic Fault Attack on `LED-64`

Description of the Attack

In the preceding two sections we derived polynomials describing the encryption map of `LED-64` and additional information gained from a fault attack. All in all, we found 6208 polynomials in 6336 indeterminates describing the encryption map, 6336 field equations, and 48 fault polynomials in 192 indeterminates.

As mentioned previously, we assume that we are able to mount a known-plaintext-attack and a repeat encryption involving the same key and the fault injection described previously. For every concrete instance of this attack, we can therefore substitute the plaintext bits, correct ciphertext bits, and faulty ciphertext bits into our polynomials. After this substitution, we have 6208 polynomials in 6208 indeterminates for the encryption map, 6208 field equations, and 48 fault polynomials in the 64 indeterminates of the secret key.

The resulting fault polynomials consist typically of 40-150 terms. Some of them (usually no more than 5) drop their degree and become linear. Of course, these linear polynomials are particularly valuable, since they decrease the complexity of the problem by one dimension. In the experiments reported below it turned out to be beneficial to interreduce the fault polynomials after substitution in order to generate more linear ones.

The polynomial systems can be solved using various techniques. For our experiments, we applied the algorithms for conversion to a SAT-solving problem explained in [11].

Experimental Results

All experiments were performed on a workstation having eight 3.5 GHz Xeon cores and 50 GB of RAM. We used the SAT-solvers **Minisat 2.2** (MS) and **CryptoMiniSat 2.9.4** (CMS). All timings are averages over ten `LED-64` instances with random plaintext, key and fault values. The first two lines of Table 7 show the timings for the straightforward application of the SAT-solving technique to the given polynomial systems.

For the second set of experiments, we first interreduced the fault polynomials using the computer algebra system **ApCoCoA** (see [1]) and then appended the linear polynomials to the system. In this way we were sometimes able to find more linear dependencies between the key indeterminates, thereby reducing the dimension even further. Moreover, the SAT-solvers appear to benefit from this simplification, because it is typically the number of terms in a polynomial that complicates its logical representation. This seemingly minor modification results in a meaningful speed-up, as we can see in line 3 and 4 of Table 7.

SAT solver	MS (1 thread)	CMS (1 thread)	CMS (4 threads)
time (in sec)	90852	71656	22639
time (in h)	25.23	19.90	6.28
time (in sec)	36665	52835	11829
time (in h)	10.18	14.67	3.28

Table 7: Average SAT Solver Timings (Lines 1 & 2) and with Additional Linear Equations (Lines 3 & 4).

In summary, it is clear that the proposed attack is able to break the `LED-64` encryption scheme. While it is slower than the direct fault attack presented in [12], it does not rely on the specific properties underlying the key filtering steps there, and it offers numerous possibilities for optimization.

References

[1] ApCoCoA: Applied Computations in Commutative Algebra, available for download at `http://www.apcocoa.org`.

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, C. Whelan, The Sorcerer's Apprentice Guide to Fault Attacks, Proceedings of the IEEE, vol. **94**, IEEE Computer Society, 2006, pp. 370–382.

[3] E. Biham ans O. Dunkelman, Techniques for cryptanalysis of block ciphers, Springer, Heidelberg 2011.

[4] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin and C. Vikkelsoe, PRESENT: An Ultra-Lightweight Block Cipher, In: P. Paillier and I. Verbauwhede (eds.) *CHES* 2007, LNCS, vol. **4727**, Springer, Heidelberg 2007, pp. 450–466.

[5] D. Boneh, R.A. DeMillo and R.J. Lipton, On the Importance of Elimination Errors in Cryptographic Computations, J. Cryptology **14** (2001), 101–119.

[6] C. Carlet, J-C. Faugere, C. Goyet, G. Renault, Analysis of the algebraic side channel attack, Journal of Cryptographic Engineering, vol. **2** nr. 1, Springer Heidelberg 2012, pp. 45–62.

[7] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, The LED Block Cipher, In: B. Preneel and T. Takagi (eds.) *CHES* 2011, LNCS, vol. **6917**, Springer, Heidelberg 2011, pp. 326–341.

[8] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, S. Chee, HIGHT: A New Block Cipher Suitable for Low-Resource Device, In: L. Goubin and M. Matsui (eds.) *CHES* 2006, LNCS, vol. **4249**, Springer, Heidelberg 2006, pp. 46–59.

[9] M. Hojsík and B. Rudolf, Differential Fault Analysis of Trivium, In: K. Nyberg (ed.) *FSE* 2008, LNCS, vol. **5086**, Springer, Heidelberg 2008, pp. 158–172.

[10] M. Hojsík and B. Rudolf, Floating Fault Analysis of Trivium, In: D.R. Chowdhury, V. Rijmen and A. Das (eds.) *INDOCRYPT* 2008, LNCS, vol. **5365**, Springer, Heidelberg 2008, pp. 239–250.

[11] P. Jovanovic and M. Kreuzer, Algebraic Attacks using SAT-Solvers, Groups – Complexity – Cryptology **2** (2010), pp. 247–259.

[12] P. Jovanovic, M. Kreuzer, I. Polian, A Fault Attack on the LED Block Cipher, In: W. Schindler and S. Huss (eds.) *COSADE* 2012, LNCS, vol. **7275**, Springer Heidelberg 2012, pp. 120–134.

[13] C.H. Kim and J-J. Quisquater, Fault Attacks for CRT Based RSA: New Attacks, New Results, and New Countermeasures, In: D. Sauveron, C. Markantonakis, A. Bilas and J-J. Quisquater (eds.) *WISTP* 2007, LNCS, vol. **4462**, Springer, Heidelberg 2007, pp. 215–228.

[14] M. Kreuzer and L. Robbiano, Computational Commutative Algebra 1, Springer Verlag, Heidelberg 2000.

[15] M.S.E. Mohamed, S. Bulygin and J. Buchmann, Using SAT Solving to Improve Differential Fault Analysis of Trivium, In: T-H. Kim, H. Adeli, R.J. Robles and M.O. Balitanas (eds.) *ISA* 2011, CCIS, vol. **200**, Springer, Heidelberg 2011, pp. 62–71.

[16] D. Mukhopadhyay, An Improved Fault Based Attack of the Advanced Encryption Standard, In: B. Preneel (ed.) *AFRICACRYPT* 2009, LNCS, vol. **5580**, Springer, Heidelberg 2009, pp. 421–434.

[17] National Institute of Standards and Technology (NIST). Advanced Encryption Standard (AES). FIPS Publication 197, available for download at `http://www.itl.nist.gov/fipsbups/`, 2001.

[18] M. Renauld, F-X. Standaert and N. Veyrat-Charvillon, Algebraic Side-Channel Attacks on the AES: Why Time also Matters in DPA, In: C. Clavier and K. Gaj (eds.) *CHES* 2009, LNCS, vol. **5747**, Springer Heidelberg 2009, pp. 97–111.

[19] M. Tunstall, D. Mukhopadhyay and S. Ali, Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault, In: C.A. Ardagna and J. Zhou (eds.) *WISTP* 2011, LNCS, vol. **6633**, Springer Heidelberg 2011, pp. 224–233.

P. Jovanovic Universität Passau
jovanovi@fim.uni-passau.de
M. Kreuzer Universität Passau
martin.kreuzer@uni-passau.de
I. Polian Universität Passau
ilia.polian@uni-passau.de

On the immunity of Boolean functions against fast algebraic attacks using bivariate polynomial representation

Meicheng Liu, Yin Zhang, and Dongdai Lin

Abstract

In the last decade, algebraic and fast algebraic attacks are regarded as the most successful attacks on LFSR-based stream ciphers. Since the notion of algebraic immunity was introduced, the properties and constructions of Boolean functions with maximum algebraic immunity have been researched in a large number of papers. However, it is unclear whether these functions behavior well against fast algebraic attacks. In this paper, we study the immunity of Boolean functions against fast algebraic attacks using bivariate polynomial representation. We present a sufficient and necessary condition for a Boolean function to achieve good immunity against fast algebraic attacks, and prove that the class of Tang-Carlet-Tang's functions achieve (almost) optimal immunity against fast algebraic attacks.

Introduction

Boolean functions are frequently used in the design of stream ciphers, block ciphers and hash functions. One of the most vital roles in cryptography of Boolean functions is to be used as filter and combination generators of stream ciphers based on linear feedback shift registers (LFSR). The study of the cryptographic criteria of Boolean functions is important because of the connections between known cryptanalytic attacks and these criteria.

In recent years, algebraic and fast algebraic attacks [1, 5, 6] have been regarded as the most successful attacks on LFSR-based stream ciphers. These attacks cleverly use overdefined systems of multivariable nonlinear equations to recover the secret key. Algebraic attacks lower the degree of the equations by multiplying a nonzero function; fast algebraic attacks obtain equations of small degree by linear combination.

Thus the algebraic immunity ($\mathcal{AI}$), the minimum algebraic degree of annihilators of f or $f+1$, was introduced by W. Meier et al. [13] to measure the ability of Boolean functions to resist algebraic attacks. It was shown by N. Courtois and W. Meier [5] that maximum $\mathcal{AI}$ of n-variable Boolean functions is $\lceil \frac{n}{2} \rceil$. Constructions of Boolean functions with maximum $\mathcal{AI}$ are researched in a large number of papers, e.g., [10, 11, 4, 16, 17]. However, there are few results referring to constructions of Boolean functions with good immunity against fast algebraic attacks.

The resistance against fast algebraic attacks is not covered by algebraic immunity [7, 2, 12]. At Eurocrypt 2006, F. Armknecht et al. [2] introduced an effective algorithm for determining the immunity against fast algebraic attacks, and showed that a class of symmetric Boolean functions (the majority functions) have poor resistance against fast algebraic attacks despite their resistance against algebraic attacks. Later M. Liu et al. [12] stated that almost all the symmetric functions including these functions with good algebraic immunity behavior badly against fast algebraic attacks. In [14] P. Rizomiliotis introduced a method to evaluate the behavior of Boolean functions against fast algebraic attacks using univariate polynomial representation.

A preprocessing of fast algebraic attacks on LFSR-based stream ciphers, which use a Boolean function $f : GF(2)^n \rightarrow GF(2)$ as the filter or combination generator, is to find a function g of small

Supported by the National 973 Program of China under Grant 2011CB302400, the National Natural Science Foundation of China under Grants 10971246, 60970152, and 61173134, the Grand Project of Institute of Software of CAS under Grant YOCX285056 and the CAS Special Grant for Postgraduate Research, Innovation and Practice.

degree such that the multiple gf has degree not too large. In [6] N. Courtois proved that for any pair of positive integers (e,d) such that $e+d \geq n$, there is a non-zero function g of degree at most e such that gf has degree at most d. This result reveals an upper bound on maximum immunity to fast algebraic attacks. It implies that the function f has maximum possible resistance against fast algebraic attacks, if for any pair of positive integers (e,d) such that $e+d < n$ and $e < n/2$, there is no non-zero function g of degree at most e such that gf has degree at most d.

In this paper, we study the immunity of Boolean functions against fast algebraic attacks using bivariate polynomial representation. Based on this representation, we prove that a Boolean function admits no non-zero function g of degree at most e such that the product gf has degree at most d if and only if the matrix $B(f;e,d)$ has full column rank. Then we prove that the functions of D. Tang et al. [15] achieve (almost) optimal immunity against fast algebraic attacks.

Immunity of Boolean functions against fast algebraic attacks using bivariate polynomial representation

In this section we focus on the immunity of Boolean functions against fast algebraic attacks using bivariate polynomial representation.

Bivariate polynomial representation

Let $\mathbb{F}_{2^n}$ denote the finite field $GF(2^n)$ and α a primitive element of $\mathbb{F}_{2^n}$. An n-variable Boolean function is a mapping from $\mathbb{F}_{2^n}$ into $\mathbb{F}_2$. Denote by $\mathbf{B}_n$ the set of all n-variable Boolean functions. An n-variable Boolean function f can be uniquely represented as its truth table, i.e., a binary string of length 2^n,

$$f = [f(0), f(1), f(\alpha), \cdots, f(\alpha^{2^n-2})].$$

The support of f is given by $\mathrm{supp}(f) = \{x \in \mathbb{F}_{2^n} \mid f(x) = 1\}$. The Hamming weight of f, denoted by $\mathrm{wt}(f)$, is the number of ones in the truth table of f. An n-variable function f is said to be balanced if its truth table contains equal number of zeros and ones, that is, $\mathrm{wt}(f) = 2^{n-1}$.

Let $n = n_1 + n_2$ $(n_1 \leq n_2)$ and denote by $m = \mathrm{lcm}(n_1, n_2)$ the least common multiple of positive integers n_1 and n_2. The Boolean function f considered as a mapping from $\mathbb{F}_{2^{n_1}} \times \mathbb{F}_{2^{n_2}}$ into $\mathbb{F}_2$ can be uniquely represented as

$$f(x,y) = \sum_{i=0}^{2^{n_1}-1} \sum_{i=0}^{2^{n_2}-1} a_{ij} x^i y^j, \; a_{ij} \in \mathbb{F}_{2^m}, \tag{1}$$

where $f^2(x,y) \equiv f(x,y)(\mathrm{mod}(x^{2^{n_1}} - x, y^{2^{n_2}} - y))$. Expression (1) is called the bivariate polynomial representation of the function f. $f^2(x,y) \equiv f(x,y)(\mathrm{mod}(x^{2^{n_1}} - x, y^{2^{n_2}} - y))$ if and only if $a_{0,0}, a_{0,2^{n_2}-1}, a_{2^{n_1}-1,0}, a_{2^{n_1}-1,2^{n_2}-1} \in \mathbb{F}_2$ and for $1 \leq i \leq 2^{n_1}-2$ and $1 \leq j \leq 2^{n_2}-2$, $a_{0,2j} = a_{0j}^2, a_{2^{n_1}-1,2j} = a_{2^{n_1}-1,j}^2, a_{2i,0} = a_{i0}^2, a_{2i,2^{n_2}-1} = a_{i,2^{n_2}-1}^2, a_{2i,2j} = a_{ij}^2$, where $2i$ and $2j$ are considered as $2i \,\mathrm{mod}(2^{n_1}-1)$ and $2j \,\mathrm{mod}(2^{n_2}-1)$ respectively. The algebraic degree of the function f equals $\max\limits_{a_{ij}\neq 0}\{\mathrm{wt}(i)+\mathrm{wt}(j)\}$.

In particular, when $n = 2k$, the Boolean function f considered as a mapping from $\mathbb{F}_{2^k} \times \mathbb{F}_{2^k}$ into $\mathbb{F}_2$ can be uniquely represented as

$$f(x,y) = \sum_{i=0}^{2^k-1} \sum_{i=0}^{2^k-1} a_{ij} x^i y^j, \; a_{ij} \in \mathbb{F}_{2^k}, \tag{2}$$

where $f^2(x,y) \equiv f(x,y)(\mathrm{mod}(x^{2^k} - x, y^{2^k} - y))$.

Immunity against fast algebraic attacks

Let

$$\mathcal{W}_e = \{(a,b) \mid \mathrm{wt}(a)+\mathrm{wt}(b) \le e, 0 \le a \le 2^{n_1}-1, 0 \le b \le 2^{n_2}-1\}$$

and

$$\overline{\mathcal{W}}_d = \{(a,b) \mid \mathrm{wt}(a)+\mathrm{wt}(b) \ge d+1, 0 \le a \le 2^{n_1}-1, 0 \le b \le 2^{n_2}-1\}.$$

Hereinafter, for $(a,b) \in \mathcal{W}_e$ or $(a,b) \in \overline{\mathcal{W}}_d$: if $a-a' < 0$ or $a+a' > 2^{n_1}-1$ $(0 \le a' \le 2^{n_1}-1)$ then the operations "$+$" and "$-$" are considered as addition and subtraction operations modulo $2^{n_1}-1$ respectively; if $b-b' < 0$ or $b+b' > 2^{n_2}-1$ $(0 \le b' \le 2^{n_2}-1)$ then the operations "$+$" and "$-$" are considered as addition and subtraction operations modulo $2^{n_2}-1$ respectively.

Let f,g,h be (n_1+n_2)-variable functions and g be a function of algebraic degree at most e satisfying that $h = gf$ has algebraic degree at most d, where $n_1 \le n_2$, $e < \frac{n_1+n_2}{2}$ and $e \le d$. Let

$$f(x,y) = \sum_{i=0}^{2^{n_1}-1}\sum_{i=0}^{2^{n_2}-1} f_{i,j}x^iy^j,\ f_{i,j} \in \mathbb{F}_{2^{\mathrm{lcm}(n_1,n_2)}},$$

$$g(x,y) = \sum_{(i,j)\in\mathcal{W}_e} g_{i,j}x^iy^j,\ g_{i,j} \in \mathbb{F}_{2^{\mathrm{lcm}(n_1,n_2)}},$$

and

$$h(x,y) = \sum_{(i,j)\in\mathcal{W}_d} h_{i,j}x^iy^j,\ h_{i,j} \in \mathbb{F}_{2^{\mathrm{lcm}(n_1,n_2)}}$$

be the bivariate polynomial representations of f, g and h respectively. For $(a,b) \in \overline{\mathcal{W}}_d$, we have $h_{a,b} = 0$ and thus

$$0 = h_{a,b} = \sum_{(u,v)\in\mathcal{W}_e} \hat{b}_{(a,b),(u,v)} g_{u,v}, \tag{3}$$

where $(a,b) \neq (u,v)$ (since $\mathcal{W}_e \cap \overline{\mathcal{W}}_d = \emptyset$ for $e \le d$) and

$$\hat{b}_{(a,b),(u,v)} = \begin{cases} 0, & \text{if } a=0, u\neq 0 \text{ or } b=0, v\neq 0, \\ f_{0,b-v} + f_{2^{n_1}-1,b-v}, & \text{if } a=u\neq 0, b\neq 0, b\neq v, \\ f_{a-u,0} + f_{a-u,2^{n_2}-1}, & \text{if } a\neq 0, a\neq u, b=v\neq 0, \\ f_{a-u,b-v}, & \text{otherwise.} \end{cases} \tag{4}$$

The above equations on $g_{u,v}$'s are homogeneous linear. Denote by $B(f;e,d)$ the coefficient matrix of the equations, which is a $\sum_{i=d+1}^{n}\binom{n}{i} \times \sum_{i=0}^{e}\binom{n}{i}$ matrix.

Theorem 1. *Let $f \in \mathbf{B}_{n_1+n_2}$, $n_1 \le n_2$, $e < \frac{n_1+n_2}{2}$ and $e \le d$. Let $\sum_{i=0}^{2^{n_1}-1}\sum_{i=0}^{2^{n_2}-1} f_{i,j}x^iy^j$ ($f_{i,j} \in \mathbb{F}_{2^{\mathrm{lcm}(n_1,n_2)}}$) be the bivariate polynomial representation of f. Then there exists no non-zero function g of degree at most e such that the product gf has degree at most d if and only if the matrix $B(f;e,d)$ has full column rank.*

A special case

Next we study the $2k$-variable Boolean functions $f(x,y) = \varphi(xL(y)) + (x^{2^k-1}+1)\psi(y)$, where φ and ψ are k-variable Boolean functions and L is a linear transformation from $\mathbb{F}_{2^k}$ into $\mathbb{F}_{2^k}$. Note that the algebraic degree of $\varphi(xL(y))$ is $2\deg(\varphi)$. We know that the algebraic degree of f is the maximum between $2\deg(\varphi)$ and $k+\deg(\psi)$. Thus f has degree $2k-1$ if and only if $\deg(\psi) = k-1$.

Theorem 2. *Let $k \neq 2^s+1$ and $f : \mathbb{F}_{2^k} \times \mathbb{F}_{2^k} \to \mathbb{F}_2, (x,y) \mapsto \varphi(xy) + (x^{2^k-1}+1)\psi(y)$, $\varphi,\psi \in \mathbf{B}_k$. If $\deg(\varphi) < n$, then there exist an integer $e < k$ and a non-zero function g with degree at most e such that the product gf has degree at most d, where $d = \max\{2k-e-2, k+\deg(\psi)\}$.*

Let α be a primitive element of $\mathbb{F}_{2^k}$. Let $\varphi_{CF} \in \mathbf{B}_k$ and

$$\mathrm{supp}(\varphi_{CF}) = \{\alpha^l, \alpha^{l+1}, \alpha^{l+2}, \cdots, \alpha^{l+2^{k-1}-1}\}, 0 \le l \le 2^k - 2. \tag{5}$$

The function φ_{CF} was first presented in [8] and further studied by C. Carlet and K. Feng [4]. The functions constructed by D. Tang et al. in [15] have the form $f(x,y) = \varphi_{CF}(xy) + (x^{2^k-1}+1)\psi(y)$. Such functions have maximum algebraic immunity and good nonlinearity. It was observed through computer experiments by Armknecht's algorithm [2] that some of D. Tang et al.'s functions have good behavior against fast algebraic attacks. Theorem 2 show the upper bounds on the immunity of these functions against fast algebraic attacks, while the following results show their lower bounds.

Theorem 3. *Let $\psi \in \mathbf{B}_k$ and $f(x,y) = \varphi_{CF}(xy) + (x^{2^k-1}+1)\psi(y) \in \mathbf{B}_{2k}$.*

Then for any positive integer e with $e < k$, the function f admits no non-zero function g with algebraic degree at most e such that gf has degree at most $2k-e-3$.

If $\deg(\psi) = k-1$, then for any positive integer e with $e < k$, the function f admits no non-zero function g with algebraic degree at most e such that gf has degree at most $2k-e-2$.

Theorem 3 state that the function $f(x,y) = \varphi_{CF}(xy) + (x^{2^k-1}+1)\psi(y)$ with $\deg(\psi) = k-1$ achieves (almost) optimal immunity against fast algebraic attacks. The function $\varphi_{CF}(xL(y)) + (x^{2^k-1}+1)\psi(y)$ has the same immunity when L is a linear permutation.

References

[1] F. Armknecht. Improving fast algebraic attacks. In *FSE 2004*, volume 3017 of *Lecture Notes in Comput. Sci.*, pages 65–82. Springer, 2004.

[2] F. Armknecht, C. Carlet, P. Gaborit, et al. Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. In *EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Comput. Sci.*, pages 147–164. Springer, 2006.

[3] C. Carlet. Boolean functions for cryptography and error correcting codes. In *Boolean Methods and Models in Mathematics, Computer Science, and Engineering*, pages 257–397. Cambridge, at Cambridge University Press, 2010.

[4] C. Carlet and K. Feng. An infinite class of balanced functions with optimal algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity. In *ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Comput. Sci.*, pages 425–440. Springer, 2008.

[5] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology-EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Comput. Sci.*, pages 345–359. Springer, 2003.

[6] N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology-CRYPTO 2003*, volume 2729 of *Lecture Notes in Comput. Sci.*, pages 176–194. Springer, 2003.

[7] N. Courtois. Cryptanalysis of Sfinks. In *ICISC 2005*, volume 3935 of *Lecture Notes in Comput. Sci.*, pages 261–269. Springer, 2006.

[8] K. Feng, Q. Liao, and J. Yang. Maximal values of generalized algebraic immunity. *Designs, Codes and Cryptography*, 50(2):243–252, 2009.

[9] S. Fischer and W. Meier. Algebraic immunity of S-boxes and augmented functions. In *FSE 2007*, volume 4593 of *Lecture Notes in Comput. Sci.*, pages 366–381. Springer, 2007.

[10] N. Li, L. Qu, W. Qi, et al. On the construction of Boolean Functions with optimal algebraic immunity. *IEEE Transaction on Information Theory*, 54(3):1330–1334, 2008.

[11] N. Li and W. Qi. Construction and analysis of Boolean functions of $2t+1$ variables with maximum algebraic immunity. In *ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Comput. Sci.*, pages 84–98. Springer, 2006.

[12] M. Liu, D. Lin, and D. Pei. Fast algebraic attacks and decomposition of symmetric Boolean functions. *IEEE Transaction on Information Theory*, 57(7):4817–4821, 2011.

[13] W. Meier, E. Pasalic, and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology-EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 474–491. Springer, 2004.

[14] P. Rizomiliotis. On the resistance of Boolean functions against algebraic attacks using univariate polynomial representation. *IEEE Transaction on Information Theory*, 56(8):4014–4024, 2010.

[15] D. Tang, C. Carlet, and X. Tang. Highly nonlinear Boolean functions with optimal algebraic immunity and good behavior against fast algebraic attacks. *Cryptology ePrint Archive,* Report 2011/366, http://eprint.iacr.org/

[16] Z. Tu and Y. Deng. A conjecture about binary strings and its applications on constructing Boolean functions with optimal algebraic immunity. *Designs, Codes and Cryptography,* 60(1):1–14, 2011.

[17] X. Zeng, C. Carlet, J. Shan, and L. Hu. More balanced Boolean functions with optimal algebraic immunity and good nonlinearity and resistance to fast algebraic attacks. *IEEE Transaction on Information Theory*, 57(9):6310–6320, 2011.

M. Liu SKLOIS, Chinese Academy of Sciences
meicheng.liu@gmail.com
Y. Zhang SKLOIS, Chinese Academy of Sciences
zhangy@is.iscas.ac.cn
D. Lin SKLOIS, Chinese Academy of Sciences
ddlin@iie.ac.cn

The MOR cryptosystem and extra-special p-groups

Ayan Mahalanobis

Abstract

This paper studies the MOR cryptosystem, using the automorphism group of the extra-special p-group of exponent p, for an odd prime p. Similar results can be obtained for extra-special p-groups of exponent p^2 and for the even prime.

Introduction

In this paper, we study the MOR cryptosystem with extra-special p groups. Similar studies were done, using the group of unitriangular matrices [2] and the group of unimodular matrices [3]. The group of unitriangular matrices and the group of unimodular matrices are both matrix groups. There are many ways to represent a group – natural representations, like a matrix representation or permutation representation, or a more abstract representation in the form of generators and relations, commonly known as a *finite presentation*. In this paper, we shift our study of the MOR cryptosystem, from the matrix representation of a group to a finite presentation. We show that using finite presentation, in the form of generators and relations, one can build a **secure** MOR cryptosystem.

In a MOR cryptosystem, one works with the *discrete logarithm problem* in the automorphism group. On one hand, this is not a major change; because the discrete logarithm problem works in a group and the automorphisms form a group. On the other hand, an automorphism group arises from any algebraic structure, like a graph, vector space, etc. So the MOR cryptosystem can be seen, as the one, that liberates the discrete logarithm problem from groups to other algebraic structures.

The principal contribution of this paper is to show that, one can build a MOR cryptosystem using a finite p-group, such that the MOR cryptosystem is **as hard as the discrete logarithm problem** in $\mathbb{F}_{q^d}$, see Theorems 1 & 2. Here d is the cardinality of a minimal generating set for that p-group.

The MOR cryptosystem

In this section we describe the MOR cryptosystem [5] as automorphisms of a finite group G, however it can be generalized to other finitely generated algebraic structures easily. A description and a critical analysis of the MOR cryptosystem is in [2] and the references there.

Description of the MOR cryptosystem

Let $G = \langle g_1, g_2, \ldots, g_\tau \rangle$, $\tau \in \mathbb{N}$ be a finite group and ϕ a non-trivial automorphism of G. Alice's keys are as follows:

Private Key m, $m \in \mathbb{N}$.

Public Key $\{\phi(g_i)\}_{i=1}^{\tau}$ and $\{\phi^m(g_i)\}_{i=1}^{\tau}$.

Encryption

a To send a message (plaintext) $a \in G$ Bob computes ϕ^r and ϕ^{mr} for a random $r \in \mathbb{N}$.

b The ciphertext is $\left(\{\phi^r(g_i)\}_{i=1}^{\tau}, \phi^{mr}(a)\right)$.

This research is supported by a NBHM research grant

Decryption

a Alice knows m, so if she receives the ciphertext $(\phi^r, \phi^{mr}(a))$, she computes ϕ^{mr} from ϕ^r and then ϕ^{-mr} and then computes a from $\phi^{mr}(a)$.

Alice knows the order of the automorphism ϕ, she can use the identity $\phi^{t-1} = \phi^{-1}$ whenever $\phi^t = 1$ to compute ϕ^{-mr}.

Notations and definitions

All definitions are standard and so are the notations.

The exponent of a finite group G is the least common multiple of all possible orders of elements in G. For a finite p-group, it is the largest order of an element in G.

The center of a group G, denoted by $\mathrm{Z}(G)$, is the set of all elements in G that commute with every element of G. It is known that $\mathrm{Z}(G)$ is *characteristic*.

For a group G, G' is the commutator of G and $\Phi(G)$ is the Frattini subgroup of G, see [1, Page 2] for details.

The description and analysis of extra-special p-groups for the MOR cryptosystem

For a given prime p, all groups of order p^2 are abelian. So the first non-abelian group G is of order p^3. There is a complete classification of groups of order p^3. For $p = 2$, there are two groups of of order 8, the dihedral group D_8, and the quaternion group Q_8.

Groups of order p^3, for an odd prime p

For a odd prime p, there are two non-isomorphic classes [6, Section 4.13] of non-abelian groups of order p^3:

$$M := \langle x,y \mid x^p = 1 = y^p; [x,y] = z \in \mathrm{Z}(M); z^p = 1\rangle \tag{1}$$

$$N := \langle x,y \mid y^p = 1; [x,y] = x^p = z \in \mathrm{Z}(N); z^p = 1\rangle \tag{2}$$

Both of these groups are 2-generator p-groups, the first one has exponent p and the second one has exponent p^2. In this paper we study the MOR cryptosystem using M, similar study can be done with N and with the D_8 and Q_8, with similar conclusions. Let ϕ be an automorphism of M, then ϕ can be written as

$$\phi(x) = x^{m_1}y^{n_1}z^{l_1} \tag{3}$$

$$\phi(y) = x^{m_2}y^{n_2}z^{l_2}. \tag{4}$$

Then $[\phi(x), \phi(y)] = z^{\det(T)}$, where $T = \begin{pmatrix} m_1 & n_1 \\ m_2 & n_2 \end{pmatrix}$. This shows that $\det(T) \neq 0 \bmod p$. Notice that $\dfrac{M}{\Phi(M)} \cong \mathbb{Z}_p \times \mathbb{Z}_p$, and M is *extra-special*, hence the group of inner automorphisms of M, denoted by I, is isomorphic to $\mathbb{Z}_p \times \mathbb{Z}_p$. This gives the following exact sequence:

$$0 \longrightarrow \mathbb{Z}_p \times \mathbb{Z}_p \longrightarrow \mathrm{Aut}(M) \longrightarrow \mathrm{GL}(2,p) \longrightarrow 1$$

There are two kinds of automorphisms of M, one that is trivial on $\mathrm{Z}(M)$ and the other that is not. Since any automorphism of the center of M can be extended to an automorphism of M, the automorphism that act non-trivially on the center are generated by

$$x \mapsto x, \quad y \mapsto y^{\theta} \tag{5}$$

where θ is primitive mod p. If we denote the automorphisms that are trivial on the center by H, then there is an exact sequence of the form

$$0 \longrightarrow \mathbb{Z}_p \times \mathbb{Z}_p \longrightarrow H \longrightarrow \mathrm{SL}(2,p) \longrightarrow 1$$

Since for M, the central and the inner automorphisms are identical, the inner automorphisms are of the form $x \mapsto xz^{d_1}$, $y \mapsto yz^{d_2}$, where $0 \leq d_1, d_2 < p$.

Hence we have shown that any automorphism ϕ of M is a composition of automorphisms, (5), inner automorphism and an element from $\mathrm{SL}(2,p)$.

It is not hard to see that if ϕ is given by

$$\begin{aligned} \phi(x) &= x^{m_1} y^{n_1} z^{l_1} \\ \phi(y) &= x^{m_2} y^{n_2} z^{l_2} \end{aligned}$$

and ϕ^m is given by

$$\begin{aligned} \phi^m(x) &= x^{m_1{}'} y^{n_1{}'} z^{l_1{}'} \\ \phi^m(y) &= x^{m_2{}'} y^{n_2{}'} z^{l_2{}'} \end{aligned}$$

then

$$\begin{pmatrix} m_1 & n_1 \\ m_2 & n_2 \end{pmatrix}^m = \begin{pmatrix} m_1' & n_1' \\ m_2' & n_2' \end{pmatrix}.$$

So the discrete logarithm problem in the automorphism $\langle\phi\rangle$ is converted to the discrete logarithm problem in $\mathrm{GL}(2,p)$. One can use m_i and n_i, $i = 1,2$ in ϕ, such that, the matrix T is in $\mathrm{SL}(2,p)$.

Conversely, assume that one can solve the discrete logarithm problem in 2×2 matrices. Then it is clear from the above argument that one can determine m from ϕ and ϕ^m. Hence we have proved the following theorem.

Theorem 1. *The hardness to solve for m from ϕ and ϕ^m is equivalent to solving a discrete logarithm problem in $GL(2,p)$.*

The best algorithm to solve the discrete logarithm problem in matrices is the Menezes-Wu algorithm [4]. That algorithm finds the eigenvalues of the matrix and the eigenvalues of the power of that matrix, and then try to solve the discrete logarithm problem in those eigenvalues. So if the characteristic polynomial corresponding to the matrix of ϕ is irreducible then the complexity to solve the discrete logarithm problem in ϕ and ϕ^m is identical to solving the discrete logarithm problem in $\mathbb{F}_{p^2}$.

Note that the determinant is a multiplicative map from the group of non-singular matrices to the field, in this case $\mathbb{F}_p$. So the determinant can reduce the discrete logarithm problem in matrices to the underlying field. However, this can easily be avoided by choosing the automorphism ϕ in such a way that the corresponding matrix is unimodular.

Extra-special p-groups of exponent p

An extra-special group P is a p-group, in which the center $\mathrm{Z}(P)$, the commutator P', and the Frattini subgroup $\Phi(P)$ are equal and cyclic of order p [6, Definition 4.14]. The two most important extra-special p-groups are M and N above. Extra-special p-groups are well studied and their automorphism groups was described by Winter [7]. We don't want to redo all the work done by Winter but refer an interested reader to his paper [7].

Let P be the *iterative central product* [1, Section 2.2] of M with itself r times. As we know M is a group of order p^3 and exponent p. This makes P an extra-special p-group of exponent p. The finite presentation for the group P is the following [1, Page 33]:

$$P = \langle x_1, \ldots, x_r, y_1, \ldots, y_r \mid [x_i, y_j] = 1, i \neq j;\ [x_i, y_i] = z \in \mathrm{Z}(P) \rangle$$

each of x_i, y_i and z is of order p.

One can define a non-degenerate, bilinear alternating form, B, on $\dfrac{P}{\Phi(P)}$ as a vector space over $\mathbb{Z}_p$ [1, Page 33]. Let $x, y \in P$, and $\overline{x}, \overline{y}$ be their image in $\dfrac{P}{\Phi(P)}$. Then $B(\overline{x}, \overline{y}) = c$, where $[x, y] = z^c$.

Description of the automorphisms of P involves three steps.

A Find all automorphisms that are non-trivial on the center.

B Prove that an automorphism preserves the bilinear form if and only if it acts trivially on the center. Let H be the subgroup of the automorphism group that acts trivially on the center.

C Prove that $H/I \cong \mathrm{Sp}(2r, p)$. Where I is the subgroup of inner automorphisms of P and $\mathrm{Sp}(2r, p)$ is the *symplectic group* on the vector space $\dfrac{P}{\Phi(P)}$ over $\mathbb{Z}_p$, defined by the bilinear form B.

We briefly sketch the proof of the above three assertions, for details, see [7]. It is known that for an extra-special p-group the inner automorphisms are identical to the central automorphisms. Hence the inner automorphisms are given by

$$x_i \mapsto x_i z^{d_i}, \;\; y_i \mapsto y_i z^{d_i'}$$

where $0 \leq d_i, d_i' < p$. Clearly there are p^{2n} inner automorphisms of P.

(A) The automorphisms that doesn't act trivially on $\mathrm{Z}(P)$ are given by powers of $z \mapsto z^{\theta}$, where θ is a primitive element mod p. Notice that $\mathrm{Z}(P)$ is a cyclic group of order p. Hence these automorphisms can be defined by:

$$\theta : \; x_i \mapsto x_i, \;\; y_i \mapsto y_i^{\theta} \tag{6}$$

where θ is primitive mod p. Clearly, θ is of order $p-1$.

(B-C) Corresponding to an automorphism ϕ of P, one can trivially define an automorphism $\overline{\phi}$ on $\dfrac{P}{\Phi(P)}$. Then the automorphism $\overline{\phi}$ preserves the bilinear form B if and only if ϕ acts trivially on $Z(P)$. This follows from the equation

$$[\phi(x), \phi(y)] = B\left(\overline{\phi(x)}, \overline{\phi(y)}\right) = B(\overline{x}, \overline{y}) = [x, y].$$

Hence there is a epimomorphism $\tau : H \to \mathrm{Sp}(2r, p)$. It is easy to see that the kernel is the set of inner automorphisms I. This proves that $H/I \cong \mathrm{Sp}(2r, p)$.

By an argument identical to the MOR cryptosystem in M, one can reduce the discrete logarithm problem in the automorphism group of the extra-special p-group P to that of a discrete logarithm problem in $\mathrm{Sp}(2r, p)$ and conversely. This proves the following:

Theorem 2. *The hardness to solve for m from ϕ and ϕ^m is equivalent to solving a discrete logarithm problem in $Sp(2r, p)$.*

The discrete logarithm problem in $\mathrm{Sp}(2r, p)$, in the best case scenario (irreducible characteristic polynomial), embeds into a discrete logarithm problem in $\mathbb{F}_{p^{2r}}$. This is the best known attack against the discrete logarithm problem in $\mathrm{Sp}(2r, p)$.

Conclusion

The discrete logarithm problem is the backbone of many modern day public key cryptosystems and key exchanges. A MOR cryptosystem generalizes the central idea of the discrete logarithm problem from a group to any finitely generated algebraic structure.

It was an open question, if one can build a secure MOR cryptosystem using the finite presentation of a group. We have shown that the answer is yes.

The situation with other extra-special p-groups is almost identical.

References

[1] C. Leedham-Green and S. McKay. *The structure of groups of prime power order*. Oxford University Press, 2002.

[2] A. Mahalanobis. A simple generalization of the ElGamal cryptosystem to non-abelian groups. *Communications in Algebra*, 36(10):3880–3891, 2008.

[3] A. Mahalanobis. A simple generalization of the ElGamal cryptosystem to non-abelian groups II. *Communications in Algebra*, 2012. to appear.

[4] A. Menezes and Y.-H. Wu. The discrete logarithm problem in $\mathrm{GL}(n,q)$. *Ars Combinatorica*, 47:23–32, 1997.

[5] S.-H. Paeng, K.-C. Ha, J. H. Kim, S. Chee, and C. Park. New public key cryptosystem using finite non-abelian groups. In J. Kilian, editor, *Crypto 2001*, volume 2139 of *LNCS*, pages 470–485. Springer-Verlag, 2001.

[6] M. Suzuki. *Group Theory II*. Springer-Verlag, 1986.

[7] D. L. Winter. The automorphism group of an extraspecial p-group. *Rocky Mountain Journal of Mathematics*, 2(2):159–168, 1972.

A Mahalanobis Indian Institute of Science Education and Research Pune
ayan.mahalanobis@gmail.com

Computational aspects of retrieving a representation of an algebraic geometry code

I. Márquez-Corbella, E. Martínez-Moro, G.R. Pellikaan, and D. Ruano

Abstract

Code-based cryptography is an interesting alternative to classic number-theory PKC since it is conjectured to be secure against quantum computer attacks. Many families of codes have been proposed for these cryptosystems such as algebraic geometry codes. In a previous paper [9] we showed that for so called very strong algebraic geometry codes $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ where $\mathcal{X}$ is an algebraic curve over $\mathbb{F}_q$ and $\mathcal{P} = (P_1, \ldots, P_n)$ is an n-tuple of mutually distinct $\mathbb{F}_q$-rational points of $\mathcal{X}$ and E is a divisor of $\mathcal{X}$ with disjoint support from $\mathcal{P}$ it was shown that an equivalent representation $\mathcal{C} = \mathcal{C}_L(\mathcal{Y}, \mathcal{Q}, F)$ can be found. The n-tuple of points are obtained directly from a generator matrix of C, where the columns are viewed as homogeneous coordinates of these points. The curve $\mathcal{Y}$ is given by $I_2(\mathcal{Y})$, the homogeneous elements of degree 2 of the vanishing ideal $I(\mathcal{Y})$. Furthermore it was shown that $I_2(\mathcal{Y})$ can be computed in an efficient as the kernel of certain linear map. What was not shown was how to get the divisor F and a decoding algorithm in an efficient way. In this talk show some work in progress on the topics needed to be dealt towards an efficient computational approach to this problem.

Introduction

In 1978, McEliece [11] introduced the first public key cryptosystem (PKC) based on the theory of error-correcting codes in particular he proposed to use a classical binary Goppa code. The security of this scheme is based on the hardness of the decoding problem for general linear codes and the hardness of distinguishing a code with the prescribed structure from a random one. Moreover, McEliece scheme an interesting candidate for post-quantum cryptography. An overview of the state of the art of cryptosystems that are secure against attacks by quantum computers is provided in [3]. Another advantage of this scheme is its fast encryption and decryption functions.

Many attempts to replace Goppa codes with different families of codes have been proven to be insecure as for example using GRS codes such as the original Niederreiter system [12] which was broken by Sidelnikov and Shestakov [13] in 1992.

let $\mathcal{X}$ be an algebraic curve of genus g over the finite field $\mathbb{F}_q$, $\mathcal{P} = (P_1, \ldots, P_n)$ be an n-tuple of mutually distinct $\mathbb{F}_q$-rational points of $\mathcal{X}$ and E a divisor of $\mathcal{X}$ with disjoint support from $\mathcal{P}$ of degree m. We define the *vector space of rational functions associated to E* as the set

$$\mathcal{L}(E) = \left\{ f \in \mathbb{F}_q(\mathcal{X}) \mid f = 0 \text{ or } (f) \geq -E \right\},$$

and the *linear series* of E as the collection $|E| = \{ F \mid F \equiv E, F \geq 0 \}$. Then the following evaluation map

$$\mathrm{ev}_{\mathcal{P}} : \quad \mathcal{L}(E) \quad \longrightarrow \quad \mathbb{F}_q^n$$

is well defined by $\mathrm{ev}_{\mathcal{P}}(f) = (f(P_1), \ldots, f(P_n))$. The *algebraic geometry code* $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ is the image of $\mathcal{L}(E)$ under the evaluation map $\mathrm{ev}_{\mathcal{P}}$, i.e.

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \{(f(P_1), \ldots, f(P_n) \mid f \in \mathcal{L}(E))\} \subseteq \mathbb{F}_q^n.$$

As consequence of the Riemann-Roch theorem, if $n > m > 2g - 2$ then $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ has dimension $m + 1 - g$ and minimum distance at least $n - m$.

Recall that GRS codes can be seen as the special class of algebraic geometry codes on the projective line, that is the algebraic curve of genus zero. This result was generalized to curves of genus 1 and 2 by Faure and Minder [5] in 2008. These attacks can be viewed as retrieving the curve, n points on this curve and the divisor E.

Since the initial Niederreiter scheme is completely broken, Berger and Loidreau [2] proposed in 2005 another version which was designed to resist precisely the Sidelnikov-Shestakov attack. The main idea of this variant is to work with subcodes of the original GRS code rather than using the complete GRS code. However Wieschebrink [14] in 2006 presents the first feasible attack to the Berger-Loidreau cryptosystem that allows us to recover the secret key if the chosen subcode is large enough but which was impractical for small subcodes. Furthermore in 2010 Wieschebrink [15] noted that it seems that with high probability the square code of a subcode of a GRS code of parameters $[n,k]$ is itself a GRS code of dimension $2k-1$.

Therefore we can apply the Sidelnikov-Shestakov attack and thus reconstruct the secret key in polynomial time. Continuing this line of work, in [10], we characterized those subcodes which are weak keys for the Berger-Loidreau cryptosystem. That is, firstly those subcodes which are themselves GRS codes, we have seen that the probability of occurrence of this fact is very small, and secondly those subcodes whose square code is a GRS code of maximal dimension which has high probability of occurrence.

In 1996 Janwa and Moreno [7] proposed to use the collection of AG codes on curves for the McEliece cryptosystem. As we have already explained this system was broken for codes on curves of genus $g \leq 2$ by Faure and Minder [5]. But the security status of this proposal for higher genus was not known.

Definition 1. *A code $\mathcal{C}$ over $\mathbb{F}_q$ is called* very strong algebraic-geometric *(VSAG) if $\mathcal{C}$ is equal to $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)$ where the curve $\mathcal{X}$ over $\mathbb{F}_q$ has genus g, $\mathcal{P}$ consists of n points and E has degree m such that*

$$2g+2 \leq m < \tfrac{1}{2}n \ \text{ or } \ \tfrac{1}{2}n+2g-2 < m \leq n-4.$$

In [9] we proved the following result

Theorem 2. *Let $\mathcal{C}$ be a VSAG code then a VSAG representation can be obtained from its generator matrix. Moreover all VSAG representations of $\mathcal{C}$ are strict isomorphic.*

Theorem 2 implies, **provided we have an efficient procedure for decoding the VSAG representation obtained in the theorem**, that one should not use VSAG codes for the McEliece PKC system in the range

$$\gamma \leq R \leq \tfrac{1}{2}-\gamma \ \text{ or } \ \tfrac{1}{2}+\gamma \leq R \leq 1-\gamma,$$

for $n \to \infty$, since there is an efficient attack by our result. In the same paper, by a shortening argument, we proved that also codes in the range

$$\tfrac{1}{2}-\gamma \leq R \leq 1-3\gamma \ \text{ or } \ 3\gamma \leq R \leq \tfrac{1}{2}+\gamma,$$

for $n \to \infty$, should be excluded. The above mentioned intervals $[\gamma, \frac{1}{2}-\gamma]$, $[\frac{1}{2}+\gamma, 1-\gamma]$, $[\frac{1}{2}-\gamma, 1-3\gamma]$ and $[3\gamma, \frac{1}{2}+\gamma]$ are nonempty if and only if $\gamma \leq \frac{1}{4}$, and the union of these intervals cover the whole interval $[\gamma, 1-\gamma]$ if and only if $\gamma \leq \frac{1}{6}$.

Work in progress

As it was mention before, a VSAG representation isomorphic to the original code can be built from the public key of the PKC (the scrambled generator matrix of the original code). Indeed, decoding the VSAG representation implies decoding the original code, i.e. breaking the cryptosystem. The purpose of this research is twofold

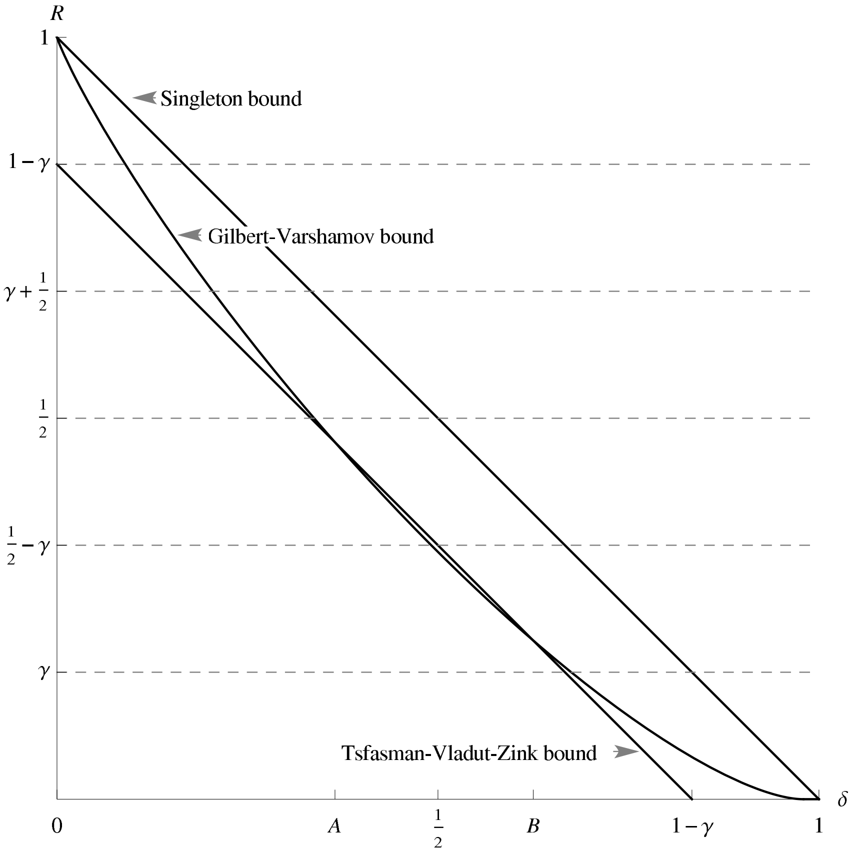


Figure 1: Bounds on R as a function of the relative minimum distance δ for $q = 49$ and $\gamma = \frac{1}{6}$.

1. Compute efficiently the VSAG representation, i.e. retrieving the triple given by the curve, a set of points and the divisor defining the functions to be evaluated.

2. Decode the code given by VSAG representation.

Up to now we have made some advances in direction 1. Indeed, if the VSAG representation lies in some of the families of AG codes that are provided with an efficient error correcting procedure this will imply tht the PKC based on the original code would be broken.

Computing the VSAG representation

Let $r = l(E) - 1$ and $\{f_0, \dots, f_r\}$ be a basis of $\mathcal{L}(E)$. Consider the following map:

$$\varphi_E : \mathcal{X} \longrightarrow \mathbb{P}^r(\mathbb{F}_q)$$

defined by $\varphi_E(P) = (f_0(P), \dots, f_r(P))$.

If $m > 2g$ then $r = m - g$, so φ_E defines an embedding of the curve $\mathcal{X}$ of degree m in $\mathbb{P}^r$. More precisely, let $\mathcal{Y} = \varphi_E(\mathcal{X})$, $Q_j = \varphi_E(P_j)$ and $\mathcal{Q} = (Q_1, \dots, Q_n)$. Then $\mathcal{Y}$ is a curve in $\mathbb{P}^{m-g}$ of degree m, φ_E is an isomorphism from $\mathcal{X}$ to $\mathcal{Y}$ and $\varphi_E(E) = \mathcal{Y} \cdot H$ for some hyperplane H of $\mathbb{P}^{m-g}$ that is disjoint from $\mathcal{Q}$. See [6, Theorems 7.33 and 7.40]. Let $F = \varphi_E(E) = \mathcal{Y} \cdot H$. Then $\mathcal{C} = \mathcal{C}_L(\mathcal{Y}, \mathcal{Q}, F)$, that is $(\mathcal{Y}, \mathcal{Q}, F)$ is also a representation of the code $\mathcal{C}$ which is strict isomorphic with $(\mathcal{X}, \mathcal{P}, E)$.

Computing $\mathcal{Y}$. Let $\mathcal{C}$ be a k dimensional subspace of $\mathbb{F}_q^n$ with basis $\{\mathbf{g}_1, \dots, \mathbf{g}_k\}$. We denote by $S^2(\mathcal{C})$ the second symmetric power of $\mathcal{C}$. If $x_i = \mathbf{g}_i$, then $S^2(\mathcal{C})$ has basis $\left\{x_i x_j \mid 1 \le i \le j \le n\right\}$

and dimension $\binom{k+1}{2}$. Furthermore we denote by $\langle \mathcal{C} * \mathcal{C} \rangle$ or $\mathcal{C}^{(2)}$ the square of $\mathcal{C}$, that is the linear subspace in $\mathbb{F}_q^n$ generated by $\{\mathbf{a} * \mathbf{b} | \mathbf{a}, \mathbf{b} \in C\}$. See [4, §4 Definition 6] and [10, 15]. Now we consider the linear map

$$\sigma : \quad S^2(\mathcal{C}) \quad \longrightarrow \quad \mathcal{C}^{(2)},$$

where the element $x_i x_j$ is mapped to $\mathbf{g}_i * \mathbf{g}_j$. The kernel of this map will be denoted by $K^2(\mathcal{C})$.

Proposition 3 (Proposition 15 in [9])**.** *Let $\mathcal{Q}$ be an n-tuple of points in $\mathbb{P}^r(\mathbb{F}_q)$ not in a hyperplane, $k = r+1$, $G_{\mathcal{Q}}$ be the $k \times n$ matrix associated to $\mathcal{Q}$ and $\mathcal{C}$ be the subspace of $\mathbb{F}_q^n$ generated by the rows of $G_{\mathcal{Q}}$. Then*

$$I_2(\mathcal{Q}) = \{ \textstyle\sum_{1 \le i \le j \le k} a_{ij} X_i X_j \mid \sum_{1 \le i \le j \le k} a_{ij} x_i x_j \in K^2(\mathcal{C}) \}.$$

Let $\mathcal{Q}$ be an n-tuple of points in $\mathbb{P}^r(\mathbb{F}_q)$ not in a hyperplane. Then $\mathcal{O}(n^2\binom{r}{2})$ is an upper bound on the complexity of the computation of $I_2(\mathcal{Q})$ and a Gröbner basis of this ideal can be computed by straight-forward adaptation of the *Projective version of the classical Buchberger-Möller Algorithm* presented in [1] for the special case where we know that the elements of the reduced Gröbner basis have degree two.

Computing $E = \mathcal{Y} \cdot H$**.**

Let $\mathbf{g}_1, \ldots, \mathbf{g}_k$ be the rows of the chosen generator matrix G of $\mathcal{C}$. By the star product $*$ he vector space $\mathbb{F}_q^n$ is an $\mathbb{F}_q$-algebra. Consider the map of $\mathbb{F}_q$-algebras

$$\varepsilon : \mathbb{F}_q[X_1, \ldots, X_k] \longrightarrow \mathbb{F}_q^n$$

given by $X_i \mapsto \mathbf{g}_i$ for $i = 1, \ldots, k$ and extended by the universal property of $\mathbb{F}_q[X_1, \ldots, X_k]$ as an $\mathbb{F}_q$-algebra.

Let R be the factor ring $R = \mathbb{F}_q[X_1, \ldots, X_k]/I(\mathcal{Y})$. The ideal $I(\mathcal{Y})$ is in the kernel of ε. Hence ε induces a map

$$\varepsilon : R \longrightarrow \mathbb{F}_q^n,$$

that we also denote by ε. Let R_d be the subspace of R given by cosets of homogeneous polynomials of degree d. Then $\varepsilon(R_1) = C$ by construction of ε, and more generally $\varepsilon(R_d) = C^{(d)}$.
Let $f(X)$ be a nonzero linear function in R_1. Then $\varepsilon(f(X)) = \mathbf{g}$ is a nonzero codeword of $\mathcal{C}$ and $\varepsilon(f(X)R_1) = \mathbf{g} * C$.

Let H be the hyperplane given by the linear equation $f(X) = 0$. We may assume without loss of generality after possibly extending the field of constants that $E = \mathcal{Y} \cdot H$ that there is a nonzero function $f \in \mathcal{L}(E)$ such that $(f)_\infty = E$, that means that the divisor of poles of f is equal to E. Let $\mathbf{g} = \mathrm{ev}_{\mathcal{P}}(f) \in \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \mathcal{C}$. Then $\mathbf{g} * C$ is a subspace of $\mathcal{C}^{(2)}$ and the coset $\mathcal{C}^{(2)}/\mathbf{g} * C$ has dimension $(2m+1-g) - (m+1-g) = m$. Therefore we have an explicitly given $\mathbb{F}_q$-linear map:

$$\mathbb{F}_q[X_1, \ldots, X_k] \longrightarrow \mathcal{C}^{(2)}/\mathbf{g} * C$$

with kernel the ideal $I_2(\mathcal{Y}) + (f)$, that is the vanishing ideal of $\mathcal{Y} \cap H$ with multiplicities counted. In this situation there is an efficient (polynomial) algorithm that computes a Gröbner basis of $I_2(\mathcal{Y}) + (f)$, see [8].

References

[1] J. Abbott, A. Bigatti, M. Kreuzer, and L. Robbiano. Computing ideals of points. *J. Symbolic Comput.*, 30(4):341–356, 2000.

[2] T. Berger and P. Loidreau. How to mask the structure of codes for a cryptographic use. *Designs, Codes and Cryptography*, 35:63–79, 2005.

[3] D. Bernstein. Introduction to post-quantum cryptography. In J. B. D.J. Bernstein and E. Dahmen, editors, *Post-quantum cryptography*, pages 1–14. Springer-Verlag, Berlin, 2009.

[4] I. Cascudo, H. Chen, R. Cramer, and X. Xing. Asymptotically good ideal linear secret sharing with strong multiplication overy any fixed finite field. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009, Lecture Notes in Computer Science*, volume 5677, pages 466–486, Berlin, 2009. Springer.

[5] C. Faure and L. Minder. Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes. In *Proceedings 11th Int. Workshop on Algebraic and Combinatorial Coding Theory, ACCT 2008*, pages 99–107, 2008.

[6] J. W. P. Hirschfeld, G. Kochmáros, and F. Torres. *Algebraic curves over a finite field*. Princeton Univ. Press, Princeton, 2008.

[7] H. Janwa and O. Moreno. McEliece public crypto system using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8:293–307, 1996.

[8] M. Marinari, H. Möller, and T. Mora. Gröbner basis of ideals defined by functionals with an application to ideals of projective points. *AAECC*, 4(2):103–145, 1993.

[9] I. Márquez-Corbella, E. Martínez-Moro, and G. Pellikaan. Cryptanalysis of public-key cryptosystems based on algebraic geometry codes. *To appear in Designs, Codes and Cryptography*, pages 20, MFO–Preprint OWP 2012 – 01, http://www.mfo.de/scientific–programme/publications/owp, 2012.

[10] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. The non-gap sequence of a subcode of a generalized Reed-Solomon code. In *To appear in Designs, Codes and Cryptography*, 2012.

[11] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report*, 42–44:114–116, 1978.

[12] H. Niederreiter. Knapsack-type crypto systems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[13] V. M. Sidelnikov and S. O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Math. Appl.*, 2:439–444, 1992.

[14] C. Wieschebrink. An attack on the modified Niederreiter encryption scheme. In *PKC 2006, Lecture Notes in Computer Science*, volume 3958, pages 14–26, Berlin, 2006. Springer.

[15] C. Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In *Post-Quantum Cryptography, Lecture Notes in Computer Science*, volume 6061, pages 61–72, Berlin, 2010. Springer.

I. Márquez-Corbella Universidad de Valladolid
imarquez@agt.uva.es

E. Martínez-Moro Universidad de Valladolid
edgar@maf.uva.es

G.R. Pellikaan Eindhoven University of Technology
g.r.pellikaan@tue.nl

D. Ruano Aalborg University
diego@math.aau.dk

SCAE: A code based authenticated encryption scheme

Mohammed Meziani and Rachid El Bansarkhani

Abstract

An authenticated encryption (AE) scheme is a better way to simultaneously provide privacy and authenticity. This paper presents a new and efficient two-pass AE scheme, called SCAE, which is different from previously proposed ones based on number theoretic problems such as factoring and discrete logarithm problem or block ciphers. The proposed scheme is based on coding theory and is the first AE scheme of this type. Its security is related to the hardness of the regular syndrome decoding problem. The security requirement of privacy and that of authenticity are also proved. Additionally, the performance of SCAE is comparable to the other efficient schemes from the theoretical point of view. A software or hardware implementation of the proposed scheme is left open as future work to show its speed in practice.

Introduction

Authenticated encryption (AE) schemes are symmetric cryptographic primitives that provide simultaneous privacy and authenticity (integrity) protection for transmitted data.

There exist many methods to construct AE schemes. As far as we know, the most provably secure authenticated encryption schemes proposed come with a rigorous proof of security via a reduction the underlying cryptographic primitive, and there exists no reduction to the well-known problems. Therefore, it is desirable to have provably secure AE constructions, whose security is grounded on hard problems. One of such problem is the decoding of random linear codes, called also the syndrome decoding (SD) problem. Unlike the number-theoretic problems such as factoring and discrete logarithm problem [12], this problem is NP-complete [6] and is believed to resist quantum algorithms (certainly for properly chosen parameters). The fastest algorithm [3] for solving this problem has an exponential running time. In addition to that, SD-based systems enjoy the benefits of having fast encryption and decryption algorithms; they only use simple operations like shifts and XORs making them one of the promising candidates for post-quantum cryptography [7].

The present work presents a two-pass efficient and provably secure authenticated encryption scheme, called SCAE, based on coding theory. To the best of our knowledge it is the first proposal of this type. Its design is inspired from the sponge approach [8] and its security depends on the hardness of the regular syndrome decoding problem. Furthermore, its security proofs are simple and straightforward. Additionally, its performance is comparable to that of the other efficient schemes. Different parameters are also proposed for SCAE allowing a trade-off between performance and security.

Preliminaries

Notations:

$|x|$: the length in bits of a string x.

$\mathtt{wt}(x)$: the Hamming weight of a string$x$, defined as the number of its non-null coordinates.

$x^\top$: the transpose of a string x.

$x \,||\, y$: the concatenation of two strings of x and y.

$\mathsf{X} \,||\, \mathsf{Y}$: the concatenation of two matrices $\mathsf{X}$and Y

$x \oplus y$: the bitwise XOR of two strings x and y, having the same size.

$x \xleftarrow{\$} S$: choosing an element x from a finite set S at random and assigning it to x

$\mathcal{M}_{\ell,\eta}$: the set of all binary random matrices of size $\ell \times \eta$.

$\mathcal{W}_{\eta,\omega}$: the set of all strings of length η and weight ω.

Linear Codes: In general, an $[n,w,k]$ linear code $\mathcal{C}$ is a k-dimensional subspace of an n-dimensional vector space over a finite field $\mathbb{F}_q$, where k and n are positive integers with $k < n$ and q a prime power. The integer $b = n - k$ is called the co-dimension of $\mathcal{C}$. The weight of a word x, denoted by $w = \mathtt{wt}(x)$, is the number of non-zero entries in x. If the quotient n/w is a power of two, then a word x of length n and weight w is called regular if it consists of w blocks of length n/w, each with a single non-zero entry. The sum of two regular words is called a 2-regular word. A generator matrix G of $\mathcal{C}$ is a matrix whose rows form a basis of $\mathcal{C}$, .i.e., $\mathcal{C} = \{x \cdot G : x \in \mathbb{F}_q^k\}$. A parity check matrix H of $\mathcal{C}$ is defined by $\mathcal{C} = \{x \in \mathbb{F}_q^n : H \cdot x^\top = 0\}$ and generates the code's dual space. In this work we set $q = 2$.

Hard Problems: The security of some code-based cryptographic primitives is related to the hardness of the following problems.

Problem 1 (Regular Syndrome Decoding (RSD)):
Given a $b \times n$ random binary matrix H, a binary vector $y \in \mathbb{F}_2^b$,and an integer $w > 0$, find a regular word $x \in \mathbb{F}_2^n$ of weight $wt(x) = w$, such that $H \cdot x^T = y$.

Problem 2 (2-Regular Null Syndrome Decoding (2-NRSD)):
Given a $b \times n$ random binary matrix H, a binary vector $y \in \mathbb{F}_2^b$,and an integer $w > 0$, find a 2-regular word $x \in \mathbb{F}_2^n$ of weight $wt(x) \leq 2w$, such that $H \cdot x^T = 0$.

These two problems have also been proven to be NP-Complete in [2].

Code-based Authenticated Encryption

The Proposed Protocol: SCAE

In what follows, we describe a new construction for an authenticated scheme based on coding theory, called SCAE, which stands for Sponge-like Code-based Authenticated Encryption scheme.

The key idea behind our construction is to use the randomize-then-combine paradigm, introduced by Bellare and Micciancio [4], inside the sponge-like construction in order to obtain a code-based authenticated encryption scheme. Unlike sponge construction, a counter is used to modify the c-bit part using XOR operation during the encryption/decryption process.

Parameters. Consider five positive integers n, w, c and r satisfying $\frac{n}{w} = 2^{\alpha}$ for some $\alpha > 0$, and $b = w \cdot \alpha = r + c$. To use our scheme one has to specify a random binary matrix $\mathbf{A}$ of size $b \times n$. Let $\mathcal{K} = \{0,1\}^{\frac{b}{2}}$ be the set of possible keys. Given these parameters, one defines an encryption function $E : \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$, where each $E(K,\cdot) = E_K(\cdot)$ is a one-to-one transformation over $\{0,1\}^b$. Formally, for a random secret key $K \in \mathcal{K}$, we first define

$$f(y) = \bigoplus_{i=1}^{w} A_i[\langle y_i \rangle],\ y = (y_1, \cdots, y_i, \cdots, y_w) \in \{0,1\}^b \text{ s.t. } |y_i| = \alpha,\ \langle y_i \rangle \in \{0,1,\ldots,2^{\alpha}-1\}, \tag{1}$$

where $A_i[j] \in \mathbb{F}^b$ for $j \in \{0,1,\ldots,2^{\alpha}-1\}$, are the columns of a random binary matrix A of size $b \times n$, and the mapping $y_i \to \langle y_i \rangle$ is the big-endian encoding algorithm converting each α-bit input block into a decimal value from $\{0,1,\cdots,2^{\alpha}-1\}$ that indicate which columns of $\mathbf{A}$ have to be combined using the bitwise XOR-operator.

Now we define our encryption functions as

$$E_K(z) = f((K|z_1) \oplus f(z_2|K)),\ z = (z_1, z_2) \in \{0,1\}^{\frac{b}{2}} \times \{0,1\}^{\frac{b}{2}}. \tag{2}$$

Description of SCAE. Before giving its detailed description, we mention the properties and techniques that our proposal uses.

• **Nonces:** Like other authenticated encryption schemes, our proposal uses a nonces N of length r bits, which is required for the encryption and decryption process. Each nonce should be non-repeating and selected by the party who want to encrypt. Every new message is associated with a single nonce.

• **Tags**. The tag length has length c bits and consists of a number of unknown ”local” tags having the same length. By trivial means, it implies that the probability to forge a valid ciphertext has to be 2^{-c}.

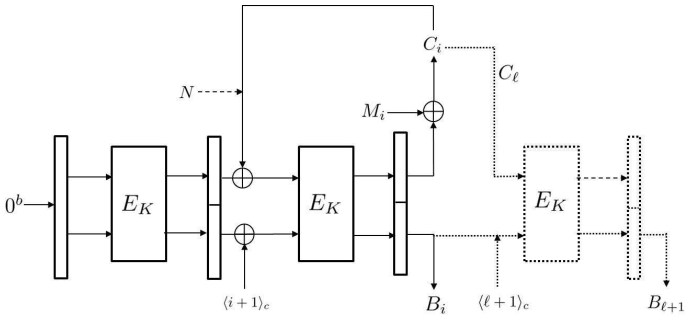


Figure 1: A schematic diagram of the proposed authenticated encryption scheme.

Provided that these properties are satisfied, our construction consists of the following steps:

- Key Generation: Select randomly a secret key K of length $b/2$ bits from $\mathcal{K}$, and binary random matrix A of size $b \times n$ to construct the encryption function $E_K(\cdot)$ defined by equation (2). The key K is then secretly transmitted to two parties who want to encrypt and decrypt in order to authenticate their messages, while the matrix A is made public.

- Encryption: To encrypt a plaintext $M \in \{0,1\}^*$ using key $K \in \{0,1\}^{\frac{b}{2}}$ and nonce $N \in \{0,1\}^{\frac{b}{2}}$, obtaining a ciphertext C and a tag T, do the following. Let $\ell = \lceil \frac{|M|}{r} \rceil$, and denote $M = (M_1, \cdots, M_\ell)$ the message to be encrypted. If $|M_\ell| < r$ then prepend one "1" followed by $r - |M_\ell|$ zeros to M_ℓ to obtain an r-bit block. As in the sponge construction, initialize the system with 0^b at the beginning. Compute $(I,J) = E_K(0^b)$ with $|I| = r$ and $|J| = c$. For $i = 1, \cdots, \ell$, produce ciphertexts C_i as follows: $C_0 = N$, $(L, B_i) = E_K(C_{i-1} \oplus I \parallel \langle i \rangle_c \oplus J)$, and $C_i = L \oplus M_i$, where $|L| = r$ and $|B_i| = c$. Then compute $(L, B_{\ell+1}) = E_K(C_\ell \oplus I \parallel \langle \ell+1 \rangle_c \oplus J)$. Finally, compute a tag $T = B_1 \oplus B_2 \oplus \cdots \oplus B_{\ell+1}$. The "local" tags $B_1, \cdots, B_{\ell+1}$ are never made directly visible to the attacker, but only their XOR-sum is returned.

- Decryption and verification: Given $C = (C_1, \cdots, C_\ell)$, T and N, the receiver knowing the secret key K executes the following in order to recover plaintext $M = (M_1, \cdots, M_\ell)$. First compute $(I,J) = E_K(0^b)$ with $|I| = r$ and $|J| = c$. Then for $i = 1$ to ℓ, do the following: $C_0 = N$, $(L, B_i) = E_K(C_{i-1} \oplus I \parallel \langle i \rangle_c \oplus J)$, and $M_i = L \oplus C_i$, where $|L| = r$ and $|B_i| = c$. Then compute $(L, B_{\ell+1}) = E_K(C_\ell \oplus I \parallel \langle \ell+1 \rangle_c \oplus J)$. To verify whether the received tag T is valid, compute $T' = B_1 \oplus B_2 \oplus \cdots \oplus B_{\ell+1}$. If T and T' match, then accept the plaintext $M = (M_1, \cdots, M_\ell)$, otherwise output a fail symbol $\perp$ indicating that the message is not authentic.

Security of SCAE

Security Notions

An authenticated encryption is designed to provide two security goals: privacy and authenticity. Following the security model in [10], these notions are formally defined as follows. An adversary $\mathcal{A}$ as a probabilistic algorithm having access to an encryption oracle $\mathcal{E}_K(\cdot,\cdot)$ selects nonce-message pairs $(N^1, M^1), \cdots, (N^q, M^q)$ and obtains the corresponding ciphertexts $\mathcal{C}^i = (C^i, T^i) = \mathcal{E}_K(N^i, M^i)$, $i = 1, \cdots, q$. The adversary must be nonce-respecting meaning that it is not allowed to repeat a nonce in its queries to the encryption oracle, i.e., $N^i \neq N^j$, for all $i \neq j$. In order to attack the privacy notion, $\mathcal{A}$ is either given access to the real encryption $\mathcal{E}_K(\cdot,\cdot)$, or to a fake oracle $\mathcal{O}(\cdot,\cdot)$, that take as input (N^i, M^i) and output random ciphertexts $\mathcal{O}(N^i, M^i)$ having the same length as the real ciphertexts $(C^i, T^i) = \mathcal{E}_K(N^i, M^i)$. The attacker has to make a distinction between both oracles. Formally, this can be defined as follows. An authenticated encryption Π is said to be ε-*privacy secure*, if for all nonce-respecting adversaries $\mathcal{A}$, it holds

$$\mathbf{Adv}_{\Pi}^{priv} = \Pr[K \xleftarrow{\$} \mathcal{K} | \mathcal{A}^{\mathcal{E}_K^{(\cdot,\cdot)}(\cdot)} = 1] - \Pr[\mathcal{A}^{\mathcal{O}(\cdot,\cdot)} = 1] \leq \varepsilon \tag{3}$$

In an authenticity attack, the adversary $\mathcal{A}$ first asks queries $(N^1, M^1), \cdots, (N^q, M^q)$, obtains the corresponding ciphertexts $\mathcal{C}^i = (C^i, T^i) = \mathcal{E}_K(N^i, M^i)$, and finally constructs a ciphertext $\mathcal{C}$ and a nonce N. It is said to successfully *forge* if $\mathcal{C} \notin \{\mathcal{C}^1, \cdots, \mathcal{C}^q\}$ and $\mathcal{D}_K(\mathcal{C})$ is valid. This can be formulated as follows. An authenticated encryption Π is said to be ε-*authenticity secure*, if for all nonce-respecting adversaries $\mathcal{A}$, it holds

$$\mathbf{Adv}_{\Pi}^{auth} = \Pr[K \xleftarrow{\$} \mathcal{K} | \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} \text{ outputs a forgery}] \leq \varepsilon \tag{4}$$

Cryptographic Assumptions

We state our complexity assumptions below.

Assumption 1: *For properly chosen parameters (n,w,b) there is no polynomial time algorithm which can distinguish the underlying $b \times n$ binary matrix from a random matrix of the same size with non-negligible probability.*

The second assumption states that it is hard to solve an instance of the (regular) syndrome decoding problem when the parameters (n,w,b) are chosen properly.

Assumption 2: *The Syndrome Decoding Problem problem with parameters (n,w,b) is hard for every polynomial time algorithm.*

Some Properties of E_K

The underlying encryption function enjoys two interesting features:

1. **Security reduction.** It is easy to prove that the encryption function E_K is reducible to the syndrome decoding problem, meaning that it can be rewritten as $E_K(x) = A \cdot y^\top$, where y is an (unknown) regular, which is related to x and K.

2. **Pseudorandomness.** Here, we show that E_K is pseud-random, meaning that its its outputs are indistinguishable from random string. This result comes from that the indistinguishability property of the randomized Niederreiter's system [9] based on the assumptions stated above.

Security Arguments

The main theorems regarding the security of SCAE scheme are stated as follows. Their proofs are given in the full version of this paper.

Assuming an nonce-respecting adversary making q queries of nonce-message pairs $(N^1, M^1), \cdots,$ (N^q, M^q), where $M^i = (M_1^i, \cdots, M_{\ell_i}^i)$, $N^i = C_0^i$ for $i = 1, \cdots, q$ and $t = \sum_{i=1}^{q} \ell_i$, and gets the corresponding ciphertexts $(C^1, T^1), \cdots, (C^q, T^q)$, with $C^i = (C_1^i, \cdots, C_{\ell_i}^i)$.

Theorem 1 (Privacy property)**.** *The SCAE scheme based on the function $E_K(\cdot)$ is* ε-privacy secure, *against all nonce-respecting adversaries, where* $\varepsilon = \frac{(q-1)t}{2^{\frac{r}{3.3}}}$.

Theorem 2 (Authenticity property)**.** *The SCAE scheme based on the function $E_K(\cdot)$ is* ε-authenticity secure *with respect to all nonce-respecting adversaries, where* $\varepsilon = \frac{(q-1)t}{2^{\frac{r}{3.3}}} + \frac{1}{2^c}$.

Performance and Comparison

Table 8 gives a brief overview on basic features of SCAE compared to some other proposals. As we can see, in particular, the theoretical cost (measured by the number of the underlying function calls) required to handel a $|M|$-bit plaintext approximately amounts to $\lceil \frac{|M|}{b} \rceil + 2$. As a result, SCAE runs at the same speed as OCB mode, and only is a bit slower than remaining schemes. Furthermore, SCAE possesses smaller and correlated tags and nonces, allowing a trade-off between the security and the performance in contrast to OCB, EAX, and GCM. Table 9 presents different parameters for our proposal including the tag size, the nonce/block, and the upper bounds for privacy and authenticity as a function of the number of queries and blocks. Note that the upper bound on the plaintext length for SCAE is $r(2^c - 3)$ bits, which approximately gives 2^c blocks .

References

[1] N. S. P. 800-38A. Recommendation for block cipher modes of operation-methods and techniques. 2001. `http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf/`.

[2] D. Augot, M. Finiasz, and N. Sendrier. A Family of Fast Syndrome Based Cryptographic Hash Functions. In E. Dawson and S. Vaudenay, editors, *Mycrypt 2005*, volume 3715, pages 64–83. Springer, 2005.

[3] A. Becker, A. M. A. Joux, and A. Meurer. Decoding random binary linear codes in $2^{(n/20)}$: How 1+1=0 improves information set decoding. Eurocrypt 2012. Lecture Notes in Computer Science, Springer-Verlag, 2012.

[4] M. Bellare and D. Micciancio. A new paradigm for collision-free hashing: incrementality at reduced cost. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 163–192. Springer, 1997.

[5] M. Bellare, P. Rogaway, and D. Wagner. Eax: A conventional authenticated-encryption mode. 2003. `http://eprint.iacr.org/`.

[6] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(2):384–386, May 1978.

[7] D. J. Bernstein, J. Buchmann, and E. Dahmen. *Post Quantum Cryptography*. Springer-Verlag, 2008.

[8] M. P. G. Bertoni, J. Daemen and G. V. Assche. Sponge Functions. In *ECRYPT Hash Workshop 2007*, 2007.

[9] R. Nojima, H. Imai, K. Kobara, and K. Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Des. Codes Cryptography*, 49:289–305, December 2008.

[10] P. Rogaway. Authenticated-encryption with associated-data. In *ACM Conference on Computer and Communications Security*, pages 98–107, 2002.

[11] P. Rogaway, M. Bellare, and J. Black. Ocb: A block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.

[12] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *SFCS '94: Proc. of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society, 1994.

M. Meziani CASED – Center for Advanced Security Research Darmstadt, Germany
mohammed.meziani@cased.de
R. El Bansarkhani Darmstadt University of Technology, Germany
elbansarkhani@cdc.informatik.tu-darmstadt.de

	With associated data?	Tag length (bits)	Nonce size (bits)	Non-repeating nonce?	Verify-Then-Decrypt?	Cost	Parallelizable?
EAX [5]	yes	$\leq b$	any	yes	yes	$\lceil \frac{M}{b} \rceil + 1$	no
OCB [11]	yes	$\leq b$	b	yes	no	$\lceil \frac{M}{b} \rceil + 2$	yes
GCM [1]	yes	$\leq b$	any	yes	yes	$\lceil \frac{M}{b} \rceil + 1$	yes
SCAE	no	$\leq c$	$\leq b - c$	yes	no	$\lceil \frac{M}{b} \rceil + 2$	no

Table 8: A comparison of basic characteristics of SCAE with some other schemes. The input size of the underlying block cipher or pseudo-random function (PRF) is equal to b bits while the tag length is c bits with $c < b$. The cost is given in terms of the number of the underlying block cipher or PRF calls. In order to get a reasonable comparison, the costs given here for EAX, OCB, and GCM modes do not include the cost to process the associated data (header).

parameters (n,w,b) $b=r+c$	tag size c (bits)	nonce/block size r (bits)	# queries q	# blocks t	upper bound for privacy $\approx \frac{qt}{2^{\frac{r}{3.3}}}$	upper bound for authenticity $\approx \frac{qt}{2^{\frac{r}{3.3}}}+\frac{1}{2^c}$	Expected security level for RSD(n,w,b)
$(8192,32,256)$	16	240	2^{10}	2^{16}	$\approx 2^{-46}$	$\approx 2^{-16}$	90
	32	224	2^{10}	2^{32}	$\approx 2^{-25}$	$\approx 2^{-25}$	
	64	192	2^{10}	2^{30}	$\approx 2^{-18}$	$\approx 2^{-18}$	
$(8192,48,384)$	16	368	2^{10}	2^{16}	$\approx 2^{-85}$	$\approx 2^{-16}$	120
	32	352	2^{20}	2^{32}	$\approx 2^{-64}$	$\approx 2^{-32}$	
	64	320	2^{20}	2^{64}	$\approx 2^{-12}$	$\approx 2^{-12}$	
$(8192,64,512)$	16	496	2^{40}	2^{16}	$\approx 2^{-94}$	$\approx 2^{-16}$	200
	32	480	2^{40}	2^{32}	$\approx 2^{-73}$	$\approx 2^{-32}$	
	64	448	2^{40}	2^{64}	$\approx 2^{-31}$	$\approx 2^{-31}$	

Table 9: Some concrete parameters for SCAE. The security levels are estimated according to the best known attack [3].

On multivariate cryptosystems based on edge transitive graphs

M. K. Polak, V. Ustimenko, and A. Wróblewska

We understand multivariable cryptography as studies of cryptosystems based on special regular automorphism f of algebraic variety $M_n(K)$ of dimension n in a sense of Zarisski topology over finite commutative ring K. An example of algebraic variety is a free module K^n which is simply a Cartesian product of n copies of K^n into itself. Regular automorphism is a bijective polynomial map of $M_n(K)$ onto itself such that f^{-1} is also a polynomial map. Elements of K^n can be identified with strings $(x_1, x_2, \dots, x_n)$ in alphabet K, nonlinear map f of restricted degree d can be used as a public rule if the key holder (Alice) knows the secret decomposition of f into composition of special maps $f_1, f_2, \dots, f_{2s}$ with known inverse maps $f_i{}^{-1}$. So she can decrypt by consecutive application of $f_{2s}{}^{-1}, f_{s-1}^{-1}, \dots, f_1{}^{-1}$. Of course K^n can be changed for the family of varieties $M_n(K)$, $n = 1, 2, \dots,$ the commutative ring can be treated as an alphabet, element $v \in M_n(k)$ as a ”potentially infinite” plaintext, parameter n (dimension) as a measurement of size of v.

Multivariate cryptosystem based on graphs $D(n,q)$ was introduced in [1], some implementations and generalizations the reader can find in [5], [6].

Bipartite graphs $D(n,q)$ have partition sets P (collection of points) and L (collection of lines) isomorphic to vector space $\mathbb{F}_q^n$. Point $(x_1, x_2, \dots x_n)$ and line $[y_1, y_2, \dots, y_n]$ are incident if and only if $y_i - x_i = x_{k(i)} y_{s(i)}$ where $k(i) < i$ and $s(i) < i$ (for the description of functions s(i) and k(i) see [1] or [2]). The parenthesis and brackets will allow us to distinguish points and lines.

As it follows from results [1] the well defined projective limit of graphs D(n, q) is an infinite q-regular forest. There is an automorphism group of $G = G(n,q)$ which acts regularly on the totality of edges for $D(n,q)$. In the case of char $\mathbb{F}_q \neq 2$ there is a factorization of group G into two subgroups G_1 and G_2 such that G_2 acts regularly on the totality of connected components of the graph and G_1 acts regularly on edges of each component. Group $G(n,q)$ acts transitively on P and L . The transformation group (G,P) is a subgroup of $AGL_n(q)$. We introduce a colour of the point or the line as its first coordinate. So, the colour set is $\mathbb{F}_q$.

We convert the graph into finite automation via labeling the directed edge between vertices v_1 and v_2 by difference of colours v_2 and v_1. Let $t_1, t_2, \dots t_{2s}$ be the sequence of labels of consecutive edges forming walk which starts from the point x. We assume that $t_{i+1} \neq -t_i$, $i = 1, 2, \dots 2s-1$. Let $\mathrm{y} = N_{t_1, t_2, \dots, t_{2s}}(x)$ be the final point of the walk. The map $\mathrm{x} \to N_{t_1, t_2, \dots, t_{2s}}(x)$ is a polynomial map on the vector space $P = F_q{}^n$.. It has degree 3 (see [6]).

Theorem
Let L_1 and L_2 be invertible affine transformations of vector space $\mathbb{F}_q^n$, such that $L_1 = L_2{}^{-1} \in G(n,q)$.

(1) The order of $F = L_1 N_{t_1, t_2, \dots, t_{2s}} L_2$, where $t_{2s} \neq t_1$ goes to infinity with the growth of parameter n.

(2) The cyclic group generated by non identical composition F of L_1, $N_{t_1, t_2, \dots, t_{2s}}$ and L_2 is a cubical map.

We will use the composition $F = F_{\mathrm{t},n} = F(L_1, L_2, \mathrm{t}, \mathbb{F}_q)$, where $\mathrm{t} = (t_1, t_2, \dots, t_{2s})$, L_1 and L_2 are sparse affine transformation of the vector space $\mathbb{F}_q^n$, as a public rule

$$\begin{gathered} x_1 \to f_1(x_1, x_2, \dots, x_n), \\ x_2 \to f_2(x_1, x_2, \dots, x_n), \\ \\ \vdots \end{gathered}$$

$$x_n \to f_n(x_1, x_2, \ldots, x_n).$$

We assume that polynomials f_i are written in a standard form and the triple (L_1, L_2, t) is hidden.

Notice that generation of cubical multivariate public rules in terms of Computer Algebra with n-variables over $\mathbb{F}_q$ will have a complexity $O(n^4)$, the complexity to break it is $3^{O(n)}$ with general algorithm based on ideology of Grőbner base or alternative methods.

The key holder (Alice) will use her knowledge about triple (L_1, L_2, t) to develope a private key algorithm $e(L_1, L_2)$, for the decryption on numerical level. Its complexity is $O(n)$.

The numerical private key algorithms with fixed affine transformation can be used alone as a tool for symmetric encryption. If q is odd, then for arbitrarily chosen plaintexts p_1 and p_2 there is a string t, such that corresponding encryption converts p_1 into p_2. So, encryption can translate text from English into Spanish under assumption that both files are of the same length.

The public rule $F_{\mathrm{t},n}$ can be used as tool for the key exchange. In fact r-th power $F_{\mathrm{t},n}^r$ of $F_{\mathrm{t},n}$, i. e. the composition of r copies of $F_{\mathrm{t},n}$, is also a cubical map. $F_{\mathrm{t},n}^r = F(L_1, L_2, \mathrm{t}', \mathbb{F}_q)$, where $\mathrm{t}' = (t_1, t_2, \ldots, t_{2s}, t_1, t_2, \ldots, t_{2s}, \ldots, t_1, t_2, \ldots, t_{2s})$ of length rs. Public users Bob and Alice can choose positive integers k_B and k_A, send to each other $F_{\mathrm{t},n}^{k_B}$ and $F_{\mathrm{t},n}^{k_A}$. After Alice computes k_A-th power of $F_{\mathrm{t},n}^{k_B}$ and Bob computes k_B-th power of $F_{\mathrm{t},n}^{k_A}$. They may use the ordered lexicographically array of coefficients of collision map $F_{\mathrm{t},n}^{k_B k_A}$ or various sparse functions on this set.

In the above mentioned construction we can change P for L or totality of edges of $D(n,q)$. More general graphs $D(n,K)$ (see [7]) are defined over general finite commutative ring K. In the paper we present the generalization of our theorem for graphs $D(n,K)$ and their groups of symmetries. We are working also on cryptographical applications of affine parts of generalized 6-gons and octagons (see [8], [3]). They are also edge transitive graphs and mentioned above scheme for creation of public rules works.

The classical extremal graph theory studies maximal or minimal simple graphs satisfying to a certain property. Let $|V|$ denotes number of vertices in graph Γ. Let C_n denotes the cycle of length n then by $ex(|V|, C_n)$ we denote the greatest size (number of edges) of C_n-free cycles graph with V vertices.

Erdõs Even Circuits Theorem
The following property holds:

$$ex(|V|, C_{2k}) \leq \mathrm{C}|V|^{1+1/k}$$

where C is positive constant.

The length of the shortest cycle in graph is called *girth*. It is clear that graph with size
$ex(|V|, C_3, C_4, \ldots, C_{2k})$ have girth $> 2k$.

In 2008 J. Tits was awarded by prestigious Abel Prize. In 1959 he started classification of geometries related to finite groups. He used the concept of D. Hilbert, shortly: geometry is a special simple graph. The minimal geometry according to Tits is a finite generalized m-gons i.e. bipartite, biregular graph of girth $2m$ and diameter m. From the existance of families of regular generalized m-gons for $m = 3, 4, 6$ it follows that the Erdõs bound is sharp for $k = 2, 3, 5$:

$$ex(v, C_4) = c_1 v^{1+1/2}$$
$$ex(v, C_6) = c_2 v^{1+1/3}$$
$$ex(v, C_{10}) = c_3 v^{1+1/5}$$

For other k ($k \neq 2, 3, 5$) we have open question, whether or not the Erdõs bound is sharp.

The distance between vertices v_1 and v_2 of the graph is the length of minimal pass from v_1 and v_2. The graph is connected if for arbitrary pair of vertices v_1, v_2 there is a pass from v_1 to v_2. The diameter of connected simple graph is the maximum od distances between vertices of the graph. *Bipartite graph* we call graph $\Gamma(V, E)$, in which a set of nodes V can be divided into two subsets $V = V_1 \cup V_2$ in such a way that no two vertices from each set V_i, $i = 1,2$ are connect by edge. We refer to bipartite graph $\Gamma(V,E)$ with partition sets V_i, $i = 1,2$, $V = V_1 \cup V_2$ as biregular one if the number of neighbors for representatives of each partition sets are constants $a+1$ and $b+1$ (bidegrees). We call the graph regular in case $a = b$.

Recall, that *generalized m-gons* are connected biregular bipartite graphs with girth $2m$ and diameter m. As for $D(n,q)$ in case of generalised m-gon $\Gamma(V_1 \cup V_2, E)$ one partition set of $V_1 = P$ is called set of point and other $V_2 = L$ is called the set of lines.

When two vertices point (p) and line $[l]$ are connected by edge we refer to this incidence pair (p,l) as *flag*. We define the distance from flag (p,l) to vertex $v \in V$ as the sum of distances from p to v and l to v.

Affine generalized m-gon can be obtained by the following way. Let us chose flag (p,l) from generalized m-gon and remove all points and lines except these with are on maximal distance from the flag. By this method we obtain biregular graph with bidegrees a and b. It is clear that affine generalized m-gons have girth $\geq 2m$. If the generalised m-gon is edge transitive then the construction of generalised m-gon does not depend on the choice of flag.

In case $m = 6$ there is only one known family of regular generalised m-gons. Its bidegree is $a+1 = b+1$, where $a = q = \alpha^M$, p is prime, $M \geq 1$. Each representative of this family is an edge transitive graph.When $m = 6$ we denote generalized m-gon as $GH(q)$ and affine generalized m-gon as $AH(q)$, where q is a prime power. Notice that $q+1$-regular graph $GH(q)$ has $1+q+q^2+q^3+q^4+q^5$ points and the same number of lines. The order of q-regular $AH(q)$ is $2q^5$. It is easy to check that this graph is on Erdõs bound for $ex(|V|, C_{10})$. We can consider $AH(q)$ as a infinite family with parameter q.

$AH(q)$ admit the following nice description ([8]). Let $\mathbb{F}_q$ be the finite field containing q elements. Each point can be identified with $(p) = (x_1,x_2,x_3,x_4,x_5)$ and each line with $[l] = [y_1,y_2,y_3,y_4,y_5]$. Brackets and parenthesis allow us to distinguish points and lines. We say point (p) is incident to line $[l]$, and we write $(p)I[l]$, if following relations on their coordinates hold:

$$\begin{cases} x_2 - y_2 = x_1 y_1 \\ 2y_3 - x_3 = 2x_1 y_1 \\ x_4 - 3y_3 = -3x_1 y_3 \\ 2x_5 - 3y_5 = 3x_3 y_2 - 3x_2 y_3 + x_4 y_1 \end{cases} \tag{1}$$

This interpretation works for $\alpha \geq 5$.

Let $v = (v_1,v_2,v_3,v_4,v_5) \in AH(q)$ (or $v = [v_1,v_2,v_3,v_4,v_5] \in AH(q)$) and $N_t(v)$ be the operator of taking neighbor of vertex v where first coordinate is $v_1 + t$:

$$N_t(v_1,v_2,v_3,v_4,v_5) \to [v_1+t,*,*,*,*]$$
$$N_t[v_1,v_2,v_3,v_4,v_5] \to (v_1+t,*,*,*,*)$$

The remaining coordinates can be determined uniquely using relations (1).

Denote the composition of $N = N_{t_1} \circ N_{t_2} \circ N_{t_3} \ldots \circ N_{t_{2s}}$ as $N_{t_1,t_2,\ldots,t_{2s}}$. It is easy to check that if $N_{t_1,t_2,\ldots,t_{2s}}(\overline{x}) = \overline{y}$ then $N_{-t_{2s},-t_{s-1},\ldots,-t_1}(\overline{y}) = \overline{x}$. N is a polynominal transformation of F_α^{5M} into itself. Let L_1, L_2 be the affine transformation of F_q^5 into itself

$$L_1 = T_{A,b} : \overline{x} \longrightarrow \overline{x}A + b,$$

where $A = \left[a_{i,j}\right]$ is 5×5 matrix with $a_{i,j} \in F_q$. It is clear that

$$L_2 = T_{A,b}^{-1} = T_{A^{-1},-bA^{-1}}.$$

If Alice want to encode information, she chooses her private encryption key $K_e = (A,b,t_1,t_2,\ldots,t_{2s})$ where $t_{i+1} \neq -t_i$ for $i = 1,..,2s-1$, which guarantes the irreducibility of the key (all elements of the key is from F_q). To encode she uses the composition:

$$F = L_1 \circ N_{t_1,t_2,\ldots,t_{2s}} \circ L_2 = L_1 \circ N_{t_1} \circ N_{t_2} \circ N_{t_3} \ldots \circ N_{t_{2s}} \circ L_2.$$

Alice private decryption map is of the form

$$L_2 \circ N_{-t_{2s},-t_{s-1},\dots,-t_1} \circ L_1$$

If we fixed A, b then for $2s \leq 5$ different keys produce distinct ciphertext.

We assume that $q = \alpha^M$, where α is fix but M can be as large as we want so algorithm is working with ”potentialy infinite” plaintext in the alfabet F_α.

$$(v_1, v_2, v_3, v_4, v_5) = (u_{1,1}, u_{1,2}, \dots, u_{1,M} \dots, u_{2,1}, \dots, u_{3,1}, \dots, u_{5,M}),$$

where $v_i \in F_q$ and $v_{n,j} \in F_\alpha$ in the choosen base.

Alice keeps secret her public key K_e. If she wants to receive confidential information from Bob (public user), she can use symbolic computation and present f in the form

$$\begin{array}{rcl} x_1 & \longrightarrow & f_1(x_1, x_2, x_3, x_4, x_5) \\ x_2 & \longrightarrow & f_2(x_1, x_2, x_3, x_4, x_5) \\ x_3 & \longrightarrow & f_3(x_1, x_2, x_3, x_4, x_5) \\ x_4 & \longrightarrow & f_4(x_1, x_2, x_3, x_4, x_5) \\ x_5 & \longrightarrow & f_5(x_1, x_2, x_3, x_4, x_5), \end{array}$$

where $x_i = (x_{i,1} x_{i,2} x_{i,3} x_{i,4}, \dots, x_{i,M})$ and f_i are polynomials from

$$F_\alpha[x_{1,1}, x_{1,2}, x_{1,3}, \dots x_{2,1}, \dots, x_{3,1}, \dots, x_{4,1}, \dots, x_{4,M}, \dots, x_{5,M}].$$

Computations show that:

$$1 < \deg f_i(x_{1,1}, x_{1,2}, x_{1,3}, .., x_{1,M} \dots, x_{2,1}, \dots, x_{3,1}, \dots, x_{5,M}) \leq 5$$

independent from the choice of string $t_1, t_2, \dots, t_{2s}$. Alice prints polynomials $f_i(x_{1,1}, x_{1,2}, x_{1,3}, .., x_{1,M} \dots, x_{2,M}, \dots, x_{3,M}, \dots, x_{4,M}, \dots, x_{5,M})$ in the telephone book. Bob can only encode information using telephone book. It is know that general algorithm of finding f^{-1} (Gröbner basis or alternative methods) has complexity $5^{O(M^2)}$. Finding of f^{-1} is equivalent of finding the minimal d such that $f^d = e$. Because of that we get $f^{-1} = f^{d-1}$. The order d is growing fast when M is growing $d = \alpha^{cM}$ and the complexity of finding f^{-1} in this case is $5^{O(5M)^2}$.

Similar scheme can be used for the generalised octagon (girth ≥ 16) over the field F_q, $q = 2^{2\beta+1}$ (see [9]).

It is interesting that families of graphs described above can be effectively used both in Coding Theory and Cryptography. Tools of Coding theory have to be used together with cryptographic algorithms because even unique error during the transmission of ciphertext can makes the decryption impossible.

Let us consider the extension of the field F_q to the field F_{q^R}, where $F_{q^R} = F_q[x]/p(x)$, $p(x)$ is irreducible polynomial of degree R.

Then affine transformation L_1 and L_2, used in public key rules, can be defined on smaller field F_q. The operator of taking neighbor N can be defined over F_{q^R}. Public rules defined via F in the generalised algorithm will be also cubical. In the case of affine generalised m-gons the degree of public rules are constant.

References

[1] F. Lazebnik F. and V. Ustimenko, *Explicit construction of graphs with an arbitrary large girth and of large size,* Discrete Appl. Math. , 60, (1995), 275 - 284.

[2] F. Lazebnik, V. A. Ustimenko, A. J. Woldar , *A new series of dense graphs of high girth,* Bulletin of the AMS 32 (1) (1995), 73-79.

[3] M. K. Polak, *On the cryptographical applications of nite geometries of small rank*, abstract of CECC-2011, Debrecen.

[4] V. Ustimenko, *Maximality of affine group and hidden graph cryptosystems*, J. Algebra Discrete Math. **10** (2004), 51–65.

[5] V. Ustimenko, *On the graph based cryptography and symbolic computations*, Serdica Journal of Computing, Proceedings of International Conference on Application of Computer Algebra, ACA-2006, Varna, N1 (2007).

[6] V. Ustimenko, *On the extremal graph theory for directed graphs and its cryptographical applications* In: T. Shaska, W.C. Huffman, D. Joener and V. Ustimenko, Advances in Coding Theory and Cryptography, Series on Coding and Cryptology, vol. 3, 181-200 (2007).

[7] V. A. Ustimenko, *Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography*, Journal of Mathematical Sciences, Springer, vol.140, N3 (2007) pp. 412-434.

[8] V. A. Ustimenko, *Graphs with special arcs and cryptography*, Acta Applicandae Mathematicae (Kluwer) 2002, 74,117-153.

[9] V. Ustimenko, A. Woldar, *Extremal properties of regular and affine generalized polygons as tactical configurations*, European Journal of Combinatorics 24 (2003):99.

[10] A. Wroblewska *On some properties of graph based public keys*, Albanian Journal of Mathematics, Volume 2, Number 3, 2008, 229-234, Proceedings of the NATO Advanced Studies Institute: ”New challenges in digital communications”.

M. K. Polak Maria Curie-Skłodowska University in Lublin
monika.kataryzna.polak@gmail.com

V. Ustimenko Maria Curie-Skłodowska University in Lublin
ustymenko_vasyl@yahoo.com

A. Wróblewska Maria Curie-Skłodowska University in Lublin
awroblewska@hektor.umcs.lublin.pl

Construction of the Tsujii-Shamir-Kasahara (TSK) type multivariate public key cryptosystem, which relies on the difficulty of prime factorization

Shigeo Tsujii, Kohtaro Tadaki, Masahito Gotaishi, and Ryou Fujita

Problem of Polynomial Algebra, with the equivalent difficulty as the Prime Factoring

A basic problem of polynomial algebra with the equivalent difficulty as the prime factorization is proposed.

Underlying Intractable Problem

Two large prime numbers p,q are selected. M^T means the transposed matrix of a matrix M.

Two prime numbers p,q are selected. $N := pq$

The plain text vector x is an m-dimentional vector, with each element defined on the residue class ring $\mathbb{Z}_N$.

$$x = (x_1, x_2, \dots, x_m)^T, x_i \in \mathbb{Z}_N, i = 1, 2, \dots, m$$

Two m-dimentional random polynomial vector $A(x), B(x)$ are generated:

$$A(x) = (a_1(x), a_2(x), \dots, a_m(x))$$
$$B(x) = (b_1(x), b_2(x), \dots, b_m(x))$$

Subsequently, an m-dimensional quadratic polynomial vector $C(x)$ on the residue class ring $\mathbb{Z}_N$ is defined using $p, q, A(x), B(x)$

$$C(x) := (c_1(x), c_2(x), \dots, c_m(x))^T = A(x)p + B(x)q \tag{1}$$

With the above assumption, the problem of finding the prime numbers p,q from the value of $C(x)$ for a given value of x, with $A(x)$ and $B(x)$ confidential, is discussed. This problem is called “prime factorization problem with additional information.” Then the following theorem is true:

Theorem 1. *The following two conditions are equivalent.*

i. Prime factorization is difficult.

ii. Prime factorization with additional information is difficult.

Structure of the Proposed System and the Trapdoor

Considering both the progress of the quantum computer technology and the progress of the development of MPKCs as the post-quantum cryptosystem, the constraint that ‘MPKCs should be secure against quantum computers,’ is lifted in this section. Here the advantage of quick ‘encryption/decryption’ or ‘signature/verification’ is pursued. We are going to formulate an MPKC whose security relies on the difficulty of prime factoring. The key point lies in the trapdoor structure included in the central map $G(z)$.

As shown in the Figure 1, the central map of the proposed system has the structure of:

Prime		Random Quadratic		Prime		Random Quadratic
Number		Polynomial Vector		Number		Polynomial Vector
p	$\times$	$A(u)$	$+$	q	$\times$	$B(u)$

Preparation of Public Key and Private Key

1. Two prime numbers p,q are selected. $N := pq$

2. The plain text vector x is an m-dimentional vector, with each element defined on the residue class ring $\mathbb{Z}_N$
$$x = (x_1, x_2, \ldots, x_m)^T,\ x_i \in \mathbb{Z}_N,\ i = 1, 2, \ldots, m$$

3. m-dimensional affine transformation is expressed as S.

4. The variable x is transformed to the intermediate variable u by the affine transformation S: $u := S(x)$

5. The central map is $G(u)$. The intermediate variable vector w is expressed as $w := G(u)$.

6. Let T be an m-dimensional affine transformation.

7. m-dimensional polynomial vector (public key) is expressed as $E(x) = (e_1(x), e_2(x), \ldots, e_m(x))$

8. Two m-dimensional polynomial vectors $A(x),\ B(x)$, both of which have the structure of TSK central map, are expressed as:
$$A(x) = (a_1(x), a_2(x), \ldots, a_m(x))^T,\ \ B(x) = (b_1(x), b_2(x), \ldots, b_m(x))^T$$

9. The quadratic polynomial vector $G(u)$ is defined as the function of $p,\ q,\ A(u),\ B(u)$
$$G(u) = (g_1(u), g_2(u), \ldots, g_m(u))^T = pA(u) + qB(u) \tag{2}$$

   The central map is structured by Sequential Solution Method, which is explained in Figure 1.

In this way intermediate variables are computed in sequence. These two polynomial vectors, $A(x)$ and $B(x)$, both of which has the structure of Sequential Solution Method, are combined symmetrically, with $A(x)$ multiplied with p and $B(x)$ with q to complement each other. The complementary structure of the central map is illustrated in Figure 1. The original Sequential Solution Method has the weakness that the element $w_m(u_1)$ is univariate. However, all elements include all variables by combining two Sequential Solution Method Structures in the proposed system.

This MPKC system is expected to have security against typical attacks such as Gröbner Bases and Rank Attacks.

Evaluation of Security

Security against Prime Factorization

Theorem 1 implies that, even if quadratic random polynomial vectors are added, the difficulty of prime number factorization maintains. However, compared with the polynomials in the Theorem 1, the polynomials shown in the Theorem 1 are in an ideal form, the ones in the proposed MPKC is not so ideal, since a trapdoor structure is included. Here we discuss whether the security of the cryptsystem is still assured by the difficulty of prime number factorization even in this case.

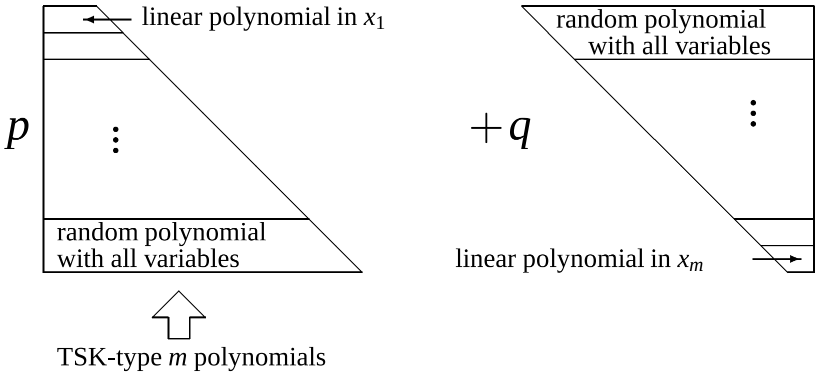

linear polynomial in x_1
p
random polynomial with all variables
$+q$
random polynomial with all variables
linear polynomial in x_m
TSK-type m polynomials

Figure 1: Structure of Central Map

Theorem 1 shows that in the equation:

$$pA_i(x) + qB_i(x) = C(x) \quad (i = 1, 2, \dots, m) \tag{3}$$

$$x = (x_1, x_2, \dots, x_m) \tag{4}$$

where $A(x),\ B(x),$ and $C_i(x)$ are quadratic random polynomials such that

$$A_i(x) = \sum_{j,k=1}^{m} a_{ijk} x_j x_k,\ B_i(x) = \sum_{j,k=1}^{m} b_{ijk} x_j x_k \tag{5}$$

In the equation (3), $C_i(x)$ is a random polynomial, without including any information of p and q, because in the equation (5), it is possible to satisfy

$$a_{ijk} x_j x_k \neq 0,\ b_{ijk} x_j x_k \neq 0 \tag{6}$$

and $C_i(x)$ becomes random polynomial.

For the public key polynomials with trapdoor construction, it is easy to satisfy (6) by properly deciding the affine transformations S and T, and the central map.

Considering that rank attacks are impossible against the proposed MPKC, as stated in the section , we can assume that attackers are unable to know the trapdoor and Theorem 1 is also applicable to the proposed MPKC.

Security against Gröbner Bases Attack

About each polynomial, as shown in the theorem 1, the security is assured by the difficulty of prime factorization. Since all coefficients of $A(u)$ and $B(u)$ are independent of each other, there is no dependency among polynomials. Hence all of the polynomials are independent of each other. Therefore theorem 1 is applicable. Consequently, it is impossible to find the plaintext of this system by computing the Gröbner Bases, as long as p and q are sufficiently large.Tranditionally the majority of the MPKC are defined on small fields such as F_2 and the number of variables is larger than 100.

Usually the first thing to do in evaluating the security of MPKCs is solving the equation system $E(x) = y$. The typical way of solvng the system is computing the Gröbner Bases of the ideal $\langle E(x) = y \rangle$. When the polynomials are defined on a finite field $GF(q)$, all variables satisfy $x_i^q = x_i$. Therefore the set of field equations $(x_1^q - x_1, \dots, x_m^q - x_m)$ is appended to the generators in computing the Gröbner Bases. Thus computed Gröbner Bases includes m or slightly fewer linear polynomials, as long as the public key $E(x)$ is determined. Without the field equations computation of Gröbner

Bases becomes too memory-consuming to proceed normally. But if the polynomials are defined on a residue class ring with large characteristics, field equations do exist, but it is impossible to find the integer d such that $x_i^d = x_i$ without factorizing the characteristic N. Or, another way of computing Göbner bases of the ideal generated by polynomial systems defined on residue class ring $\mathbb{Z}_N$ is computing the ones on the ideal defined on partial fields F_p, F_q. It is also impossible without the knowledge of $pq = N$. Therefore if the attacker attempts to attack the cryptosystem by computing Gröbner bases, they have to compute it regarding the base ring as a field F_N.

Security against Rank Attack or other Attacks analyzing the structure of the Secret key

Since all polynomials of the central map has the rank m, rank attack is fundamentally impossible in this system. Therefore, although this system is a variant of TSK type MPKC, there is no probabilistic algorithm which it is impossible to generate an element of central map $C(u)$ without knowing p or q. Moreover, it is still difficult to extract an element of $C(u)$ even if there is not the affine transformation S. Let S be identical map $(u := S(x) = x)$.The public key $P(x) := (p_1(x), \ldots, p_m(x))^T$ is expressed as $p_i(x) := \sum_{j=1}^m t_{ij}(pa_j(x) + qb_j(x))$ $(1 \leq i \leq m, t_{ij}$ is the j-th element of the i-th row of $T)$. Let $a_i(x) := \sum_{j=1}^m \alpha_{ij} x_i x_j,\ b_i(x) := \sum_{j=1}^m \beta_{ij} x_i x_j$. When 0 is assgined to variables $x_2, \ldots, x_m$, only $b_1(x_1, \ldots, x_m)$ of $B(x)$ remains and the polynomial vector $P(x)$ becomes:

$$p(\sum_{j=1}^m t_{1j} b_{j11} x_1^2, \ldots, \sum_{j=1}^m t_{mj} b_{j11} x_1^2)^T + q(t_{11} b_{111} x_1^2, \ldots, t_{m1} b_{111} x_1^2)^T \tag{7}$$

$$:= p\gamma(x_1) + q\delta(x_1)$$

It would be found that the polynomial (8) is the "Prime factorization with additional information," where the parameter m is 1. Hence it is difficult to extract $\gamma(x_1),\ \delta(x_1)$, even if there is not the affine transformation S.

Conclusion

The structure of an MPKC, with the security assured by the difficulty of prime factoring, is described. The system proposed here is an example and there are several combinations of existing cryptosystems for $A(x)$ and $B(x)$. The cryptosystems considered in this paper are the sequencial solution methods. But it is possible to choose other cryptosystems such as MI or HFE. The possibility of likely combinations of the cryptosystems and their usage should be studied further in the future. Additionally, the encryption and decryption are expected to be made faster compared with RSA or Elliptic Curve. We are going to discuss the matter further.

Acknowledgment

The authors greatly appreciate Professor Masao Kasahara of Osaka Gakuin University and Dr. Ryuichi Sakai of Osaka Electro-Communication Univerisity, both of whom have had active and fruitful discussion with authors and have given important advices and encouragement.

S. Tsujii Chuo University
tsujii@tamacc.chuo-u.ac.jp
K. Tadaki Chuo University
tadaki@tamacc.chuo-u.ac.jp
M. Gotaishi Chuo University
gotaishi@tamacc.chuo-u.ac.jp
R. Fujita Chuo University
rfujita@tamacc.chuo-u.ac.jp

On the family of cubical multivariate cryptosystems based on exceptional extremal graphs

Vasyl Ustimenko and Urszula Romańczuk

On the definiton of multivariate cryptography

Multivariate cryptography in the narrow sense (see [3]) is the generic term for asymmetric cryptographic primitives based on multivariate polynomials over finite fields. In certain cases these polynomials could be defined over both a ground and an extension field. If the polynomials have the degree two, we talk about multivariate quadratics. Algorithm of finding a solution of systems of multivariate polynomial equations is proven to be NP-Hard or NP-Complete. That is why these schemes are often considered to be good candidates for post-quantum cryptography, once quantum computers can break the current schemes. Today multivariate quadratics could be used only to build signatures. This definition rises several questions: Why a finite field but not a commutative ring is used? Why quadratics are so important?

We define multivariable cryptography as studies of cryptosystems based on special regular automorphism f of algebraic variety $M_n(\mathbb{K})$ of dimension n in a sense of Zarisski topology over finite commutative ring $\mathbb{K}$. An example of algebraic variety is a free module $\mathbb{K}^n$ which is simply a Cartesian product of n copies of $\mathbb{K}^n$ into iself. Regular automorphism is a bijective polynomial map of $M_n(\mathbb{K})$ onto itself such that f^{-1} is also a polynomial map. Elements of $\mathbb{K}^n$ can be identified with strings $(x_1, x_2, \dots, x_n)$ in alphabet $\mathbb{K}$, nonlinear map f of restricted degree d can be used as a public rule if the key holder (Alice) knows the secret decomposition of f into composition of special maps $f_1, f_2, \dots, f_k$ with known inverse maps $f_i{}^{-1}$. So she can decrypt by consecutive application of $f_k{}^{-1}, f_{k-1}^{-1}, \dots, f_1{}^{-1}$. Notice, that public user (Bob) has to use symbolic computations to work with f, but Alice may use numerical computations for the implementation of private key decryption process. Of course K^n can be changed for the family of varieties $M_n(\mathbb{K})$, $n = 1, 2, \dots$, the commutative ring can be treated as an alphabet, element $v \in M_n(\mathbb{K})$ as a ”potentially infinite” plaintext, parameter n as a measurement of size of variety.

The complexity of the best general algorithms for the solution of nonlinear system of equation of kind $f(x) = y$, $x, y \in \mathbb{K}^n$ equals $d^{0(n)}$ (see recent paper [1]). One can use Gröbner basis, Gauss elimination method or alternative options for the investigation of the system. Of course, one can write simple nonlinear equations which are easy to solve. So the system of nonlinear equations has to be tested on ”pseudorandomness” and the map f has to be of large order. Notice, that one of the first attempts to create workable multivariate cryptosystem was proposed by Imai and Matsumoto. They used finite field of characteristic 2 and its extension, f has a decomposition $f_1 f_2 f_3$, where f_1 and f_2 are affine maps (of degree 1) and f_2 is a Frobenius automorphism. Cryptanalysis for the scheme the reader can find in [3], the history of its various modifications goes on (see, for instance survey in [3]). We have to notice that the failure of this cryptosystem is not a surprise for specialists in algebra. Despite its formal quadratic appearance Frobenius automorphism is quite close to linear maps (in his famous book [2] J.Diedonne uses term 3/2 linear map for such automorphism). One of the popular directions in multivariate cryptography is the use of tools outside commutative algebra such as dynamical systems or extremal algebraic graphs (see [4], [5], [14]) and further references) for the creativity of nonlinear maps of pseudorandom nature.

The reader can find history survey of an varius the modifications of Imai and Matsumoto cryptosystem in [3].

Multivariate cryptography, Post-Quantum Information Security and pseudo-random graphs

One of the goals of Multivariable Cryptography is is development of new cryptosystems, which have some potential to be used in the era of Postquantum Cryptography. The Quantum Computer is a special random computational machine. Recall that computation in Turing machine can be formalised with the concept of finite automaton as a walk in the graph with arrows labelled by special symbols. ”Random computation” can be defined as a random walk in the random graph. So we are looking for the deterministic approximation of random graphs by extremal algebraic graphs. It is known that the explicit solutions for an optimization graphs have properties similar to random graphs.

The probability of having rather short cycle in the walking process on random graph is zero. So the special direction of Extremal Graph Theory of studies of graphs of order v (the variable) without short cycles of maximal size (number of edges) can lead to the discovery of good approximation for random graphs. On can use dual problem of finding k-regular graphs of minimal order v without cycles of given length $3,4,\dots,d$ during the search for good pseudorandom graphs. We can try to use similar idea for directed graphs, which are important for automata theory. In that case we have to prohibit commutative diagrams instead cycles. So we will look for optimal algebraic graphs. Recall that in case of algebraic graph, its vertex set and edge set (arrow set for directed graph) are algebraic varieties over special finite ring $\mathbb{K}$. Of course for the purposes of Multivariate Cryptography we need a strong additional condition that walk of the graph produce bijective polynomial nonlinear automorphism of the vertex set of restricted polynomial degree.

In the case of simple graphs we concentrate mainly on the investigation of maximal size $ex(C_3,C_4,\dots,C_{2m},v)$ of the graph on v vertices without cycles of length $3,4,\dots,2m$ i. e. graphs of girth $> 2m$. Recall that the girth is the length of minimal cycle in the simple graph. As it follows from famous Even Circuit Theorem by P. Erdős we have inequality

$$ex(C_3,C_4,\dots,C_{2m},v) \le cv^{1+1/n},$$

where c is a certain constant. The bound is known to be sharp only for $n = 4,6,10$.

The first general lower bounds of kind

$$ex(v,C_3,C_4,\dots C_n) = \Omega(v^{1+c/n}) \tag{1}$$

where c is some constant $< 1/2$ had been obtained in 50th by famous Erdős via studies of *families of graphs of large girth*, i.e. infinite families of simple regular graphs Γ_i of degree k_i and order v_i such that

$$g(\Gamma_i) \ge c\log_{k_i} v_i,$$

where c is the independent of i constant. Erdős proved the existence of such a family with arbitrary large but bounded degree $k_i = k$ with $c = 1/4$ by his famous probabilistic method.

Just two explicit families of graphs of large girth with unbounded girth and arbitrarily large k are known: the family of Cayley graphs had been defined by G. Margulis [8] and the family of algebraic graphs $CD(n,q)$ (see [7] or [9], [14] and further references).

The best known lower bound for $d \ne 2,3,5$ had been obtained in [7]:

$$ex(v,C_3,C_4,\dots,C_{2d}) = c(v^{1+2/(3d-3+e)}) \tag{2}$$

where $e = 0$ if d is odd, and $e = 1$ if d is even. This results is based on studies of graphs $CD(n,q)$.

The family of graph $D(n,q)$ and their conected components $CD(n,q)$ was known as unique family nonlinear algebraic graphs of large girth.

We generalize the concept of a family of graphs of large girth in the following way.

Let us refer to the minimal length of a cycle through the given vertex of the simple graph as cycle indicator of the vertex. We define the cycle indicator of the graph as maximal cycle indicator of its vertices. Regular graph will be called graph with irregular cycle indicator if this indicator differs from the girth (the length of minimal cycle). The solution of the optimization problem of computation of maximal size $e = e(v)$ of the graph of order v with the size greater than d, $d > 2$ has been found very recently. It turns out that

$$e(v) \Leftrightarrow O(v^{1+[2/d]})$$

and this bound is always sharp (see [1] and further references).

Let $g_x = g_x(\Gamma)$ be the length of the minimal cycle through the vertex x from the set $V(\Gamma)$ of vertices in graph Γ. We refer to

$$\mathrm{Cind}(\Gamma) = \max\{g_x,\ x \in V(\Gamma)\}$$

as *cycle indicator* of the graph Γ.

We refer to the graph Γ as *cycle irregular graph* if

$$\mathrm{Cind}(\Gamma) \neq g(\Gamma).$$

We refer to the family of regular simple graphs Γ_i of degree k_i and order v_i as *family of graphs of large cycle indicator,* if

$$\mathrm{Cind}(\Gamma_i) \geq c\log_{\mathrm{k_i}}(v_i)$$

for some independent constant c, $c > 0$. We refer to the maximal value of c satisfying the above inequality as *speed of growth* of the girth indicator for family of graphs Γ_i.

We refer to such a family as a *family of graphs of large irregular cycle indicator* if almost all graph from the family is cycle irregular graph.

The explicit construction of such family of graphs was given in [10], [14]. This is the sequence of graph the sequence of graphs $A(n,q)$, $n = 2,3,\ldots$ with the given degree of kind $q = p^s$, where p is arbitrary odd prime and s is arbitrary positive integer. If q is odd, our graphs form the *family of small world graphs*. Irregularity of cycle indicator insure that graphs are not vertex transitive. Graphs $A(n,q)$ form a *family of expanding graphs* with the second largest eigenvalue $\leq 2\sqrt{q}$ (almoust Ramanujan graphs). So, they have the largest possible spectral gap. If odd q is fixed, then well defined projective limit of graphs $A(n,q)$ is a q-regular tree.

The algebraic graphs $A(n,q)$ over a finite field $\mathbb{F}_q$

Below we consider the family of graphs $A(n,q)$ over a finite finite field of $q = p^n$ elements, where $n > 2$.

We define first an infinite family of graphs $A(q)$. Let P and L be two copies of a infinite-dimensional vector space $\mathbb{F}_q^{\mathbb{N}}$, where $\mathbb{F}_q$ is the finite field and $\mathbb{N}$ is the set of positive integer numbers. Elements of P will be called *points* and those of L *lines*. To distinguish points from lines we use parentheses and brackets. If $x \in V$, then $(x) \in P$ and $[x] \in L$. It will also be advantageous to adopt the notation for coordinates of points and lines for the case of a general finite field $\mathbb{F}_q$ we have:

$$(p) = (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,2}, p_{2,3}, \ldots, p_{i,i}, p_{i,i+1}, \ldots)$$
$$[l] = [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,2}, l_{2,3}, \ldots, l_{i,i}, l_{i,i+1}, \ldots].$$

The elements of P and L can be thought as infinite ordered tuples of elements from $\mathbb{F}_q$, such that only finite number of components are different from zero. We now define an incidence structure (P,L,I) as follows. We say the point (p) is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations between their coordinates hold:

$$\begin{aligned} l_{i,i} - p_{i,i} &= l_{1,0}p_{i-1,i} \\ l_{i,i+1} - p_{i,i+1} &= l_{i,i}p_{0,1} \qquad i = 1,2,\ldots \end{aligned}$$

For each positive integer $n \geq 2$ we obtain an incidence structure (P_n,L_n,I_n) as follows. First, P_n and L_n are obtained from P and L, respectively, by simply projecting each vector into its n initial coordinates with respect to the above order. The incidence I_n is then defined by imposing the first

$n-1$ incidence equations and ignoring all others. The incidence graph corresponding to the structure (P_n, L_n, I_n) is denoted by $A(n,q)$. It's clear, that $A(n,q)$ is a q-regular bipartite graph of order $2q^n$.

For each positive integer $n \geq 2$ we consider the *standard graph homomorphism* ϕ_n of (P_n, L_n, I_n) onto $(P_{n-1}, L_{n-1}, I_{n-1})$ defined as simple projection of each vector from P_n and L_n onto its $n-1$ initial coordinates with respect to the above order.

We define the *colour function* π for the graph $A(n,q)$ as a projection of tuples $(p) \in P_n$ and $[l] \in L_n$ onto the first coordinate (p) or $[l]$, respectively. So the set of colours is $\mathbb{F}_q$.

Let $P_{t,n} = P_A(t, n, \mathbb{F}_q)$ be the operator of taking the neighbour of point of colour $p_{0,1} + t$

$$(p) = (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,2}, p_{2,3}, \ldots, p_{i,i}, p_{i,i+1}, \ldots)$$

of kind

$$[l] = [p_{0,1} + t, l_{1,1}, l_{1,2}, l_{2,2}, l_{2,3}, \ldots, l_{i,i}, l_{i,i+1}, \ldots],$$

where parameters $l_{1,1}$, $l_{1,2}$, $l_{2,2}$,$l_{2,3}$, $\ldots$, $l_{i,i}$, $l_{i,i+1}$, $\ldots$ are computed consequently from the equations in definition of $A(n,q)$. Similarly, $L_{t,n} = L_A(t, n, \mathbb{F}_q)$ is the operator of taking the neighbour of line of colour $l_{1,0} + t$

$$[l] = [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,2}, l_{2,3}, \ldots, l_{i,i}, l_{i,i+1}, \ldots]$$

of kind

$$(p) = (l_{1,0} + x, p_{1,1}, p_{1,2}, p_{2,2}, p_{2,3}, \ldots, p_{i,i}, p_{i,i+1}, \ldots),$$

where parameters $p_{1,1}$, $p_{1,2}$, $p_{2,2}$, $p_{2,3}$,$\ldots$, $p_{i,i}$, $p_{i,i+1}$, $\ldots$ are computed consequently from the written above equations.

Notice, that $P_n = L_n = \mathbb{F}_q^n$. So we can think that $P_{A,t,n}$ and $L_{A,t,n}$ are bijective operators on the n-dimensional vector space $\mathbb{F}_q^n$. The following statement is presented in [14].

Theorem 1. [14] *Let* $\operatorname{char} \mathbb{F}_q \neq 2$, $(t_1, t_2, \ldots, t_k) \in \mathbb{F}_q^k$. *Then*

(i) each nonidentical transformation $F_{A_P, t_1, t_2, \ldots, t_k, n}$*, which is composition of maps* $P_{A,t_1,n}$*,* $L_{A,t_2,n}$*,* $\ldots$*,* $P_{A,t_{k-1},n}$*,* $L_{A,t_k,n}$ *for even number* k *or* $P_{A,t_1,n}$*,* $L_{A,t_2,n}$*,* $\ldots$*,* $L_{A,t_{k-1},n}$*,* $P_{A,t_k,n}$ *for odd number* k *is a cubical map,*

(ii) each nonidentical transformation $F_{A_L, t_1, t_2, \ldots, t_k, n}$*, which is composition of maps* $L_{A,t_1,n}$*,* $P_{A,t_2,n}$*,* $\ldots$*,* $L_{A,t_{k-1},n}$*,* $P_{A,t_k,n}$ *for even number* k *or* $L_{A,t_1,n}$*,* $P_{A,t_2,n}$*,* $\ldots$*,* $P_{A,t_{k-1},n}$*,* $L_{A,t_k,n}$*, for odd number* k *is a cubical map,*

(iii) for nonidentical transforations $F_{A_P, t_1, t_2, \ldots, t_k, n}$ *and* $F_{A_L, t_1, t_2, \ldots, t_k, n}$*, with* $t_i + t_{i+1} \neq 0$*,* $t_1 + t_k \neq 0$ *the order goes to infinity.*

We say, g is *cubical map* if it has a form

$$g = (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)),$$

where $f_i(x_1, \ldots, x_n)$ are polynomials of n variables written as the sums of monomials of kind $x_{i_1}^{n_1} x_{i_2}^{n_2} x_{i_3}^{n_3}$, where $i_1, i_2, i_3 \in \{1, 2, \ldots, n\}$; $n_1,\ n_2,\ n_3 \in \{0, 1, 2, 3\}$, $n_1 + n_2 + n_3 \leq 3$, with the coefficients from $\mathbb{K} = \mathbb{F}_q$. As we mention before the polynomial equations $y_i = f_i(x_1, x_2, \ldots, x_n)$, which are made public, have the degree 3.

Application of algebraic graphs in Cryptography

In this section we present our multivariate public key cryptosystem using results from the previous section. Our cryptosystem will work in any arbitrary finite field $\mathbb{F}_q$. The plainspace of the algorithm is $\mathbb{F}_q^n$, where $\mathbb{F}_q$ is the chosen finite field. Graph theoretical encryption corresponds to walk on the bipartite graph with partition sets which are isomorphic to $\mathbb{F}_q^n$. We conjugate chosen graph based encryption map, which is a composition of several elementary cubical polynomial automorphisms of a n-dimentional vector space $\mathbb{F}_q^n$ with special invertible affine transformation of $\mathbb{F}_q^n$. Finally we compute symbolically the corresponding cubic public map g of $\mathbb{F}_q^n$ onto $\mathbb{F}_q^n$.

Private-key algorithms We assume that two users Alice and Bob, share a common password consisting graph the sequence of color $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $\alpha_{i+1} - \alpha_i \neq 0$, $i = 1, \ldots, k-1$ and two affine transformations τ_1, τ_2 form affine group $AGL(n,q)$. Then, they encrypt the plaintext m to ciphertext c as follows: $c = \tau_1 F_{A_P, t_1, t_2, \ldots, t_k, n} \tau_2(m)$
Decryption process is as follows: $m = \tau_2^{-1} F^{-1}_{A_P, t_1, t_2, \ldots, t_k, n} \tau_1^{-1}(c)$.
If $k < \frac{g(A(n,q))}{2}$ then different keys produce distinct ciphertext.

Public-key algorithm We assume that $\alpha_i + \alpha_{i+1} \in M(\mathbb{K})$ for $i = 1, 2, \ldots$. Alice takes τ_1, τ_2, sequence $\alpha_1, \alpha_2, \ldots, \alpha_s$, authomorphism of graph $A(n,q)$, $\psi, \zeta \in G$ and creators map

$$f_A = \tau_1 F_{A_P, t_1, t_2, \ldots, t_k, n} \tau_2$$

in symbolic way (She can use with "Maple" or "Mathematica"). She is getting a public key via cubical public rule:

$$\begin{gathered} x_1 \to f_1(x_1, x_2, \ldots, x_n), \\ x_2 \to f_2(x_1, x_2, \ldots, x_n), \\ \ldots, \\ x_n \to f_n(x_1, x_2, \ldots, x_n), \end{gathered}$$

where f_i are multivariable polynomials from $\mathbb{K}[x_1, x_2, \ldots, x_n]$.

Symbolic Diffie-Hellman algorithm Suppose Alice and Bob want to agree on a key K_{AB}.
1. The first step Alice computes $f = \tau_1 F_{A_P, t_1, t_2, \ldots, t_k, n} \tau_1^{-1}$ $(\alpha_{i+1} - \alpha_i \neq 0, i = 1, \ldots, k-1, \alpha_k - \alpha_1 \neq 0)$ of large order with usage of graph $A(n, \mathbb{K})$ and she sends f to Bob. The next step is for Alice to pick a secret integer n_A that she does not reveal to anyone, while at the same time Bob picks an integer n_B that he keeps secret.
2. Alice and Bob use their secret integers (n_A and n_B, respectively) to compute $A = f^{n_A}$ and $B = f^{n_B}$, respectively. They use composition of multivariable map f with itself. They next exchange these computed values.
3. Finally, Alice and Bob again use their secret integers to compute $K_{AB} \equiv B^{n_A} \equiv (f^{n_B})^{n_A} = f^{n_A n_B}$, and $K_{AB} \equiv A^{n_B} \equiv (f^{n_A})^{n_B} = f^{n_A n_B}$, respectively. Notice that, the collision transformation $f^{n_A n_B}$ is a cubical.

Security of the cryptographic algorithms using based on the complexity of hard discrete logarithm problem for the group generated by cubical transformations defined by graphs $A(n,q)$ (see Theorem 1). This algorithms also have a good mixing properties because families of graphs $A(n,q)$ has a good expansion properties.

In [9], [10] the reader can find the generalisation of the mentioned above algorithms for general commutative rings. The implementation of private key algorithm is described in [4], the evaluation of density properties of public rules via computer simulation the reader can find in [5]. Some previous cryptosystems based on algebraic graphs the reader can find in books [12], [14], [15].

APPENDIX: To complete the description of algorithms we define more general graphs $A(n, \mathbb{K})$ and primitive functions $F_{A_P, t_1, t_2, \ldots, t_k, n}$, where $\mathbb{K}$ is a general commutative ring. We have $A(n, \mathbb{F}_q) = A(n,q)$.

We define a bipartite graph $A(n, \mathbb{K})$ with the set of points $P_n = \mathbb{K}^n$ and set of lines $L_n = \mathbb{K}^n$, where $\mathbb{K}^n$ is a free module, via incidence relation I: xIy for $x = (x_1, x_2, \ldots, x_n) \in P$ and $y = [y_1, y_2, \ldots, y_n] \in L$ if and only if, when conditions $x_1 - y_1 = y_1 x_1$, $x_2 - y_2 = x_1 y_2$, $x_3 - y_3 = y_1 x_2$, $x_4 - y_4 = x_1 y_3$, $\ldots$, $x_n - y_n = x_1 y_{n-1}$ (for even n) and $x_n - y_n = y_1 x_{n-1}$ (for odd n). Brackets and parenthesis will allow us to distinguish points and lines.

Let us assume that the colour of the vertex v is the first coordinate of this vector (point or line). So colours are elements of $\mathbb{K}$. Each vertex v of graph $A(n, \mathbb{K})$ has unique neighbour of given colour.

Let $P_{A,t,n}$ and $L_{A,t,n}$ be the maps on the vertex set of graph $A(n,\mathbb{K})$, which transforms point $x = (x_1, x_2, \ldots, x_n)$ to its neighbour of colour $x_1 + t$, $t \in \mathbb{K}$ and transforms line $y = [y_1, y_2, \ldots, y_n]$ into its neighbour of colour $y_1 + t$, respectivly.

Some examples of this graphs over small rings is presented in the Figures 1-4 .

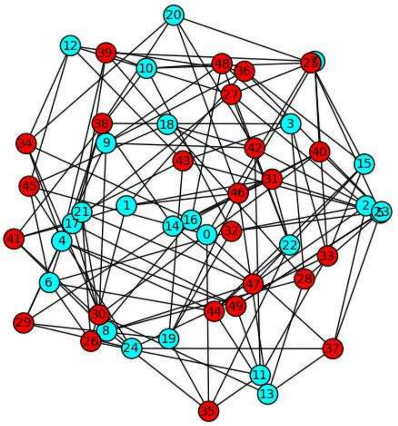


Figure 1: Graph $A(2,\mathbb{Z}_5)$

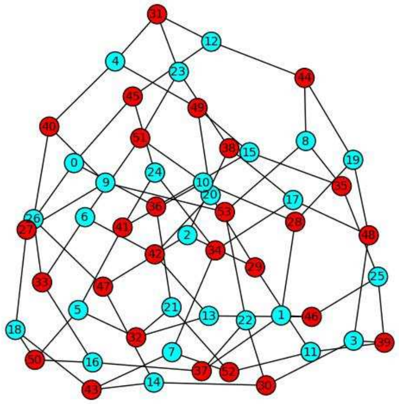


Figure 2: Graph $A(3,\mathbb{Z}_3)$

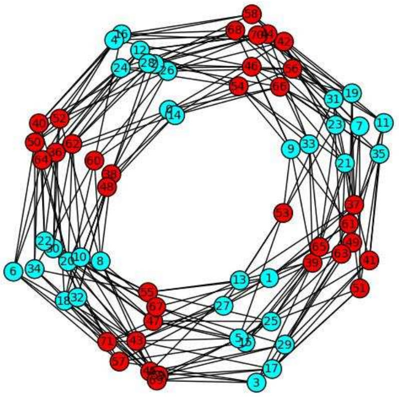


Figure 3: Graph $A(2,\mathbb{Z}_6)$

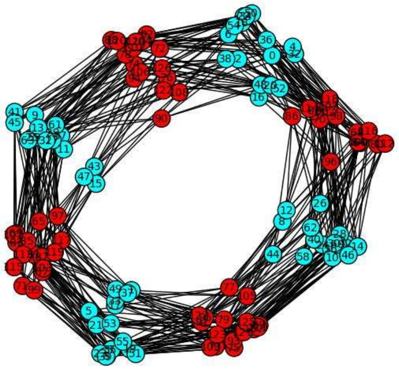


Figure 4: Graph $A(2,\mathbb{Z}_8)$

References

[1] A. L. Chistov. *An improvement of the complexity bound for solving systems of polynomial equations*, Zapisky nauchnych seminarov POMI, vol. 390, 2011, 299-306.

[2] J. Dieudonné. *La géométrie des groupes classiques*, Ergebnisse der Mathematik und ihrer Grenzgebiete (N.F.), Heft 5, Berlin, New York: Springer-Verlag, 1970.

[3] J. Ding, J. E. Gower, D. S. Schmidt. *Multivariate Public Key Cryptosystems*,Springer, Advances in Information Security, Vol. 25, 2006, XVIII, 260 p. 20 illus.

[4] J. S. Kotorowicz, V. Ustimenko, U. Romaúczk. *On the implementation of stream ciphers based on a new family of algebraic graphs*, IEEE Computer Society Press, Proceedings of the Conference CANA, FedSCIS, 2011 , pp. 485-490.

[5] M. Klisowski, U. Romańczuk, V. Ustimenko. *On the implementation of cubic public keys based on new family of algebraic graphs*, Annales UMCS Informatica AI XI, 2 (2011) p. 127-141;

[6] Z. Kotulski, J. Szczepaski. *Discrete chaotic cryptography*, Annalen der Physik 6 (1997), 381-394.

[7] Lazebnik F., Ustimenko V., Woldar A.J.: *A new series of dense graphs of high girth*, Bulletin of the AMS 32 (1) (1995), 73-79.

[8] G. A. Margulis. *Explicit construction of graphs without short cycles and low density codes*, Combinatorica, 2, (1982), 71-78.

[9] U. Romańczuk, V. Ustimenko. *On dynamical systems of large girth or cycle indicator and their applications to multivariate cryptography*, in "Artificial Intelligence, Evolutionary Computing and Metaheuristics ", In the footsteps of Alan Turing Series: Studies in Computational Intelligence, Vol. 427, Springer, June, 2012.

[10] U. Romańczuk, V. Ustimenko. *On Extremal Graph Theory, explicit algebraic constructions of extremal graphs and corresponding Turing encryption machines*, in "Artificial Intelligence, Evolutionary Computing and Metaheuristics ", In the footsteps of Alan Turing Series: Studies in Computational Intelligence, Vol. 427, Springer, June, 2012.

[11] U. Romańuczk, V. Ustimenko. *On the key exchange with new cubical maps based on graphs*, Annales UMCS Informatica AI XI 4, 11–19 (2011)

[12] T. Shaska, W.C. Huffman, D. Joener, V. Ustimenko (editors). *Advances in Coding Theory and Cryptography*, Series on Coding and Cryptology, World Scientific, 2007, 398 p.

[13] V. Ustimenko. *Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography*, Journal of Mathematical Sciences, Springer, vol.140, N3 (2007) pp. 412-434.

[14] V. Ustimenko. *On the cryptographical properties of extremal algebraic graphs*, Algebraic Aspects of Digital Communications.- NATO Science for Peace and Security Series. - D: Information and Communication Security, vol. 24.- IOS Press.- July 2009.- P. 256–281.

[15] V. Ustimenko. *Algebraic graphs and security of digital communications*, Institute of Computer Science, University of Maria Curie Sklodowska in Lublin, 2011, 151 p (oppen access book supported by European Social Foundation), available at the UMCS web.$http://informatyka.umcs.lublin.pl/files/ustimenko.pdf$.

V. Ustimenko Maria Curie-Skłodowska University in Lublin
ustymenko_vasyl@yahoo.com
U. Romańczuk Maria Curie-Skłodowska University in Lublin
urszula_romanczuk@yahoo.pl

Optimizing guessing strategies for algebraic cryptanalysis of EPCBC

Michael Walter and Stanislav Bulygin

Abstract

In this work we demonstrate how to use Mixed Integer Linear Programming to optimize guessing strategies for algebraic cryptanalysis of EPCBC-96. We are able to obtain practical attacks for the cipher with up to 3 rounds. Furthermore, we are able to demonstrate attacks that are faster than brute force for up to 5 rounds. Finally, we are able to identify a class of weak keys for which the attack is faster than brute force for up to 6 rounds.

Keywords: Algebraic Cryptanalysis, Lightweight Cryptography, Guessing Strategies, Mixed Integer Linear Programming

Introduction

The idea of algebraic cryptanalysis is to relate the inputs and outputs of a cryptographic primitive by a set of polynomial equations. In the past decade it has emerged as specific cryptanalytic method. Since analyzing primitives that yield a fairly large polynomial system is often practically infeasible, guessing strategies can be employed to estimate the complexity of attacks and thus obtain cryptanalytic results anyway, as demonstrated for example in [2] for the PRINTCipher [3]. Since cryptographic primitives often compute the output by applying a number of rounds comprising several operations to a state, the information inferred by the guesses can propagate through the corresponding polynomial system. This is especially true for ciphers inspired by PRESENT [1], since the permutation layer is realized as a plain bit permutation, where known information can pass through without restrictions. As guesses of different variables yield different information popagation, the question arises which variables to guess to achieve optimal results. We believe and, in fact, show in this work that a reasonable optimization goal for this problem is the maximization of information flow, since this minimizes the size of the resulting polynomial system. We demonstrate how to use Mixed Integer Linear Programming (MILP) to achieve this goal for EPCBC-96 [6].

For most of this work that does not involve the optimization of guessing strategies we largely follow the methodology of [2], where PRINTCipher was analyzed with algebraic techniques. As EPCBC-96, PRINTCipher is also inspired by PRESENT and comprises similar operations, so applying the methods to EPCBC-96 is very straight-forward. We also employ SAT solving to solve the polynomials systems.

Description of EPCBC-96

EPCBC-96 is a lightweight block cipher proposed by Yap et al. in 2011 [6]. The cipher's block size and key length is $b = 96$. It is heavily inspired by PRESENT [1]. Accordingly, the key schedule and the encryption itself exhibit very strong structural similarities to PRESENT and to each other. Both, the key schedule and the encryption, consist of $r = 32$ rounds, each round consisting of a substitution layer, a permutation layer and a key or constant addition layer. The substitution and permutation layer are identical in both the key schedule and the encryption. Furthermore, the substitution layer employs the same S-Box as PRESENT and the bit permutation P defining the permutation layer also strongly resembles the one of PRESENT. While the key schedule simply adds the round counter to the state during the constant addition layer, the key addition layer of the encryption adds the subkeys

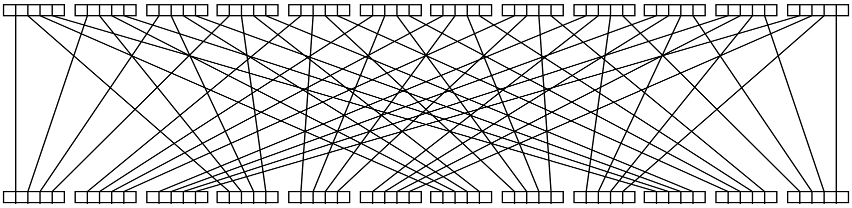


Figure 1: One round of the key schedule of EPCBC-48

produced by the key schedule. Additionally, the encryption includes an addition of the master key before the first round. For further details we refer to [6].

Optimizing Information Flow using MILP

We introduce two MILPs that choose guessing strategies in order to maximize the information flow for EPCBC-96. Note that for the purpose of information flow maximization we can neglect the constant addition of the key schedule, since the constants are publicly known. Furthermore, we can circumvent the key addition as well, due to the strong symmetry of the key schedule and the encryption. When guessing bits only in the first state of the key schedule, i.e. in the master key, all of these bits correspond to known bits in the plaintext and the knowledge is thus propagated through the first key addition layer of the encryption. As this is true for all guessed key bits, every propagated bit in the key schedule corresponds to a known bit in the encryption. So, if bits are only guessed in the input of the key schedule, the information flow in key schedule and encryption are identical. It follows that we only have to model the key schedule without the constant addition, i.e. a network of interleaved substitution and permutation layers, and maximize the information. A simple example with one round is illustrated in Figure 1 (due to space constraints for the 48 bit version of EPCBC).

Simple Propagation Model

In this section we introduce a simple model in the sense that we assume that the output bits of a certain S-Box can only be learned if all of its input bits are known. For this, let us assume a network consisting of r rounds of the EPCBC-96 key schedule (without constant addition). The state width is denoted by b. For the model we introduce a boolean decision variable $x_{i,j}$ for every bit of the state in each round with the semantics that $x_{i,j} = 1$ iff the j-th bit is known after round i. The objective function is now straight-forward:

$$\max \sum_{i=0}^{r} \sum_{j=0}^{b-1} x_{i,j} \tag{1}$$

Similarily, we can easily limit the number of bits we want to guess to an arbitrary integer k:

$$\sum_{j=0}^{b-1} x_{0,j} \leq k \tag{2}$$

Finally, we have to translate the semantics of the decision variables into our model. For this consider an arbitrary S-Box in round i and let x_{i,j_0}, x_{i,j_1}, x_{i,j_2}, x_{i,j_3} be the variables corresponding to the input bits of this S-Box. Note, that the variables corresponding to the output bits of the S-Box are $x_{i+1,P(j_0)}$, $x_{i+1,P(j_1)}$, $x_{i+1,P(j_2)}$, $x_{i+1,P(j_3)}$. To model the propagation of information through this S-Box, we include the following set of constraints:

$$x_{i+1,P(j_t)} \leq x_{i,j_s} \quad \text{for all } t,s \in \{0,\cdots,3\} \tag{3}$$

This set of constraints ensures that an output bit of the S-Box is only known, i.e. $x_{i+1,P(j_t)} = 1$, if all corresponding input bits are known. Including this set of constraints for every S-Box in every round models the information flow for the whole network.

We solved this MILP using a MILP solver[1] for $r = 4$ rounds and $k = 64$ guesses. A drawback of our method is that the model is increasingly hard to solve for more rounds. However, our result showed that information propagation does not extend beyond round three, so this solution is optimal for all rounds larger than three (but not necessarily the only optimal solution). By guessing according to this strategy we were able to infer at least $z = 160$ additional bits. Accounting for the bits propagated in the encryption, this sums up to reducing the polynomial system by at least 384 variables.

S-Box adjusted Propagation Model

In the previous subsection only known and unknown bits were distinguished, but their specific values were disregarded. In this section, we want to take them into account by adjusting the constraints in (3). For many S-Boxes some information about the output can be inferred even if the input is only partially known. For example, if the second, third, and fourth bit of the input of the S-Box used in EPCBC are known or assumed to have the value 0, then the second and third bit of the output must have the values 0 and 1, respectively. We will denote such relations as *masks*.

Again, consider an arbitrary S-Box in round i with the input variables x_{i,j_0}, x_{i,j_1}, x_{i,j_2}, x_{i,j_3} and output variables $x_{i+1,P(j_0)}$, $x_{i+1,P(j_1)}$, $x_{i+1,P(j_2)}$, $x_{i+1,P(j_3)}$. The concatenation of these variables can be seen as an 8-dimensional binary vector and the constraints in (3) describe a 0/1-polytope in 8-dimensional space that contains all points that represent a valid information flow through an S-Box. For example, this polytope contains the points $(1,1,1,1,1,1,1,1)$ and $(1,0,1,0,0,0,0,0)$, which represent the information flow with fully known input propagated to the output and partial input that is not propagated, respectively. The polytope does not contain the point $(0,1,1,1,0,1,1,0)$, as would be desired for the example of the EPCBC S-Box mask above. To remedy this we can construct the polytope using its vertex representation, i.e. we construct the polytope as the convex hull of the set of points that all describe a valid information flow. Subsequently, the vertex representation can be converted into a set of equations and inequalities describing the same polytope using the Double Description Method [5][2]. Including this set of constraints into the MILP instead of the constraints in (3) for every S-Box yields an MILP that models the information flow for a specific S-Box and specific values. We solved the system for $r = 5$. If all partial S-Box inputs satisfy a mask corresponding to the respective vertex used in the solution in key schedule and encryption, the polynomial system could be reduced by up to 512 variables with the 64 guesses. However, our method neglects the fact that only certain values for partially known inputs of an S-Box actually yield information about certain output bits. For this reason the information flow returned by the MILP solver leads to conflicts, since the vertices used for some successive S-Boxes impose different values for the same bit of a state. We will discuss this problem a little further in the next section.

Results

We constructed the polynomial system corresponding to one encryption of EPCBC-96 under a secret key and known plain-/ciphertext for increasing number of rounds r. Solving this system yields a successful key recovery attack on the round-reduced cipher. In a standard approach, this system is fed into a solver[3] without guessing. We ran the attack on our testservers and our results are listed in Table 10a for $1 \leq r \leq 3$. For larger round numbers r the attack was practically infeasible.

In our second attack we employed the guessing strategy derived in Section to reduce the polynomial system. We denote the time needed by the SAT solver to prove a guess for k out of the 96 key

[1] IBM ILOG CPLEX V12.1 under the academic license

[2] Fukuda's `cddlib` accessed through SAGE interface

[3] CryptoMiniSat 2.9.2

bits to be incorrect as t_{false}^{96-k}. We use the term t_{eval} to denote a lower bound for the time needed to encrypt a plaintext. To estimate this bound we accounted one processor cycle for each substitution and each addition layer and assumed a processor speed equivalent to the one used in our experiments to achieve comparability. Our attack is considered successful, if $t_{false}^{96-k} < 2^{96-k} \cdot t_{eval}$. For further details we refer to [2]. We compared the average solving times to the bounds imposed by the brute force attack and the results are listed in Table 10b. The results show that using this method we have found attacks on EPCBC-96 for up to $r = 5$ rounds.

We compared our strategy with 10 random strategies for $r = 5$. For each of these strategies we selected 16 out of the 24 S-Boxes of the first substitution layer of the key schedule randomly and selected their input bits as a guessing strategy. Running the same experiment with them as we did for our optimized strategy showed that the best of these random strategies reduced the system by 272.84 variables on average and yielded an estimation of $t_{false}^{32} = 14.05s$, which is significantly slower than the estimation we obtained with the optimized strategy (cf. Table 10b). Almost all other strategies resulted in an estimation that is slower than the bound imposed by the brute force attack, i.e. did not result in a successful attack.

Finally, we solved the model derived in Section . We have already pointed out that the information flow is sometimes invalid, since the vertices used for successive S-Boxes may yield some conflicts. However, close inspection of our result revealed, that there was a set of crucial S-Boxes that allowed for significant information propagation, if the mask conditions were satisfied for these S-Box inputs. For the key schedule to take advantage of this S-Box adjusted information flow, there were 55 bits at the output of round 2 that needed to have a specific value each. We will denote them by *active* bits. A key will be denoted as *weak*, if the key schedule applied to it results in inner states, i.e. subkeys, meeting these requirements. There must be 2^{41} weak keys since two rounds of the key schedule yield a 96-bit permutation.

We also wanted to achieve the information flow for the encryption. Due to the symmetry of the key schedule and the encryption, we needed the partial inputs of the same crucial set of S-Boxes to have the same values as in the key schedule. Due to the key addition layer, these active bits were required to be 0 during the encryption. Given a key, such a plaintext can be constructed easily by fixing the 55 active bits to 0, choosing arbitrary values for the remaining bits and applying two rounds of decryption to this constructed state. In 100 experiments we were able to reduce the polynomial system by 497 variables on average.

With this in mind it is possible to construct a chosen plaintext attack under the assumption that the key is weak. Again, we consider the attack successful, if $t_{false}^{32} < 2^{32} \cdot t_{eval}$. Our results are listed in Table 10c and show, that for $r \leq 6$ our attack is successful for this specific class of weak keys. We believe, that especially the drastic differences in the observed average hardness of the polynomial system in Table 10b and 10c serve as support for our introductory claim: the more information can be inferred by guessing a set of bits, the easier the problem is to solve. Furthermore, we believe that extracting more (near-) optimal solutions of the MILP will yield further classes of weak keys which are potentially even successful for larger numbers of rounds.

Conclusion

We have demonstrated how to use a MILP to optimize guessing strategies for EPCBC-96. We were able to demonstrate practical attacks for the cipher with up to 3 rounds. Furthermore, we obtained theoretical attacks for up to 5 rounds. Finally, we identified a class of weak keys for which the attack is faster than brute force for up to 6 rounds.

References

[1] A. Bogdanov, L. R. Knudsen, G. Le, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES*, volume 4727 of

r	1	2	3
avg in s	0.017	0.28	556.7
stdev in s	0.005	0.014	2282.7

(a) Standard algebraic known plaintext attack

r	4	5	6
$2^{32} \cdot t_{eval}$ in s	29.9	37.3	44.8
avg in s	0.062	0.42	4324.7
stdev in s	0.040	0.038	5614.7

(b) Comparison of t^{32}_{false} to brute-force attack (Simple Model)

r	6	7
$2^{32} \cdot t_{eval}$ in s	44.8	52.3
avg in s	0.44	125.1
stdev in s	1.78	153.3

(c) Comparison of t^{32}_{false} for weak keys to brute-force attack

Table 10: Results of the attacks (all derived from 100 runs)

Lecture Notes in Computer Science, 2007.

[2] S. Bulygin and J. Buchmann. Algebraic Cryptanalysis of the Round-Reduced and Side Channel Analysis of the Full PRINTCipher-48. In Lin et al. [4], pages 54–75.

[3] L. R. Knudsen, G. Leander, A. Poschmann, and M. J. B. Robshaw. PRINTcipher: A Block Cipher for IC-Printing. In S. Mangard and F.-X. Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.

[4] D. Lin, G. Tsudik, and X. Wang, editors. *Cryptology and Network Security - 10th International Conference, CANS 2011, Sanya, China, December 10-12, 2011. Proceedings*, volume 7092 of *Lecture Notes in Computer Science*, 2011. Springer.

[5] T. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. The Double Description Method. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*. Princeton University Press, 1953.

[6] H. Yap, K. Khoo, A. Poschmann, and M. Henricksen. EPCBC - A Block Cipher Suitable for Electronic Product Code Encryption. In Lin et al. [4], pages 76–97.

M. Walter Technische Universität Darmstadt
michael.walter@swel.com
S. Bulygin Technische Universität Darmstadt
Stanislav.Bulygin@cased.de