Perturbing and Protecting a Traceable Block Cipher

Julien Bringer, Hervé Chabanne, Emmanuelle Dottax
Sagem Défense Sécurité

Abstract

At the Asiacrypt 2003 conference Billet and Gilbert introduce a block cipher, which, to quote them, has the following paradoxical traceability properties: it is computationally easy to derive many equivalent distinct descriptions of the same instance of the block cipher; but it is computationally difficult, given one or even up to k of them, to recover the so-called meta-key from which they were derived, or to find any additional equivalent description, or more generally to forge any new untraceable description of the same instance of the block cipher.

Their construction relies on the Isomorphism of Polynomials (IP) problem. We here show how to strengthen this construction against algebraic attacks by concealing the underlying IP problems. Our modification is such that our description of the block cipher now does not give the expected results all the time and parallel executions are used to obtain the correct value.

Keywords. Traitor tracing, Isomorphism of Polynomials (IP) problem, perturbation.

1 Introduction

Traitor tracing was first introduced by B. Chor, A. Fiat and M. Naor [4]. This concept helps to fight against illegal distribution of cryptographic keys. Namely, in a system, each legitimate user comes with some keys. We suppose that a hacker can somehow have access to them, maybe because some legitimate users are traitors. These keys can then be duplicated, or new keys can be created by a pirate, computed from legitimate ones. Traitor tracing enables an authority to identify one or all of the users in possession of the keys at the origin of the pirated ones.

Often traitor tracing is employed in a broadcast network. An encrypted signal is broadcasted and each legitimate user has the keys needed to decrypt it.

Today, many traitor tracing schemes are based on some key distribution and management techniques; the distribution of the keys is dependent on some combinatorial construction. A novelty comes in 1999 with D. Boneh and M. Franklin [3] (see also [10]) where public key cryptosystems are considered.

At the Asiacrypt 2003 conference, Billet and Gilbert [2] propose a traitor tracing scheme taking place at a different level as the block cipher which allows the decryption of the signal, also permits the traitor tracing functionality. To this aim, a block cipher which have many descriptions is introduced. All descriptions give – of course – the same result. Their idea relies on the Isomorphism of Polynomials (IP) trapdoor [12], based on algebraic problems for multivariate polynomials over finite fields. It was supposed that from one or many descriptions of this block cipher it is not possible to create new ones both allowing to decrypt the broadcasted signal and preventing the authority to trace back pirates. However, recently, Faugère and Perret [9] have presented a new algorithm for solving IP-like instances and have achieved to solve a challenge proposed in [2].

Following the internal modifications of the Matsumoto-Imai cryptosystem from Ding [6], we add perturbations to Billet and Gilbert's traceable block cipher. Doing so, we want to protect the trapdoors from direct algebraic attacks (as for instance the recent algorithms of [7] and [9]), i.e. we want to alter the formal description of each round which forms the block cipher. However, here, we must still keep the traceable property with regard to the original block cipher. To manage this constraint, the pertubations are chosen in a particular way and we run in parallel, for each round, multiple descriptions of this round. None of them always gives the right result but we can show that a majority of these descriptions actually does, leading us to the expected value.

The paper is organized as follows. In Sect. 2, we recall a description of the traceable block cipher given by Billet and Gilbert. In Sect. 3, we give the principles of our modification of this traceable block cipher. In Sect. 4, we introduce the polynomials and techniques we use to fullfil our goal. In Sect. 5, we give practical implementations of our ideas. Starting from the examples given in [2], we describe their modified versions. We also show how to trace back pirates with our modified traceable block cipher.

2 A traceable block cipher

The traceable block cipher of Billet and Gilbert is made of a succession of rounds. Each round is given by a system of equations in a finite field $\mathbb{F}$. The authority possesses a meta-key which allows it to compute the secret representations of the block cipher. The public representations consist of the suitable systems of polynomials $G_{i,j}$.

The left part of Figure 1 illustrates the secret authority description. Each round is made of a non-linear part preceded and followed by a linear transformation.

The invertible linear transformations $L_{i,j}$ depend on user j, the same is true for the order in which non-linear parts occur in the block cipher. We call σ_j this permutation of the rounds. Thus, for user j, the system of polynomials, giving his public representation of the rounds, is uniquely determined by the linear parts of the round $L_{i,j}$ and σ_j. It is computed from the secret representation by the authority and lies in the right part of Figure 1. For user j, we denote them by $G_{1,j}, \ldots, G_{r,j}$.

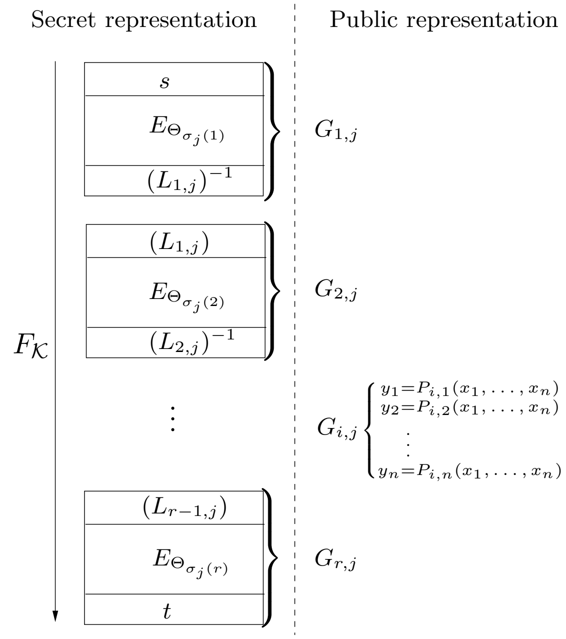
Secret representation
Public representation
s
$E_{\Theta_{\sigma_j(1)}}$
$(L_{1,j})^{-1}$
$G_{1,j}$
$(L_{1,j})$
$E_{\Theta_{\sigma_j(2)}}$
$(L_{2,j})^{-1}$
$G_{2,j}$
$F_{\mathcal{K}}$
$G_{i,j}$
$y_1 = P_{i,1}(x_1, \ldots, x_n)$
$y_2 = P_{i,2}(x_1, \ldots, x_n)$
$y_n = P_{i,n}(x_1, \ldots, x_n)$
$(L_{r-1,j})$
$E_{\Theta_{\sigma_j(r)}}$
t
$G_{r,j}$

Figure 1: A traceable block cipher

Here,

- r is the number of rounds,
- n stands for the number of variables,
- s, t and the $L_{i,j}$ are linear,
- the $E_{\Theta_{\sigma_j(i)}}$ are non-linear,
- the polynomials $P_{i,1}, \ldots, P_{i,n}$ are homogeneous of degree d.

Remark 1 *The linear transformations s and t are shared by all the users of the system.*

What made this block cipher traceable is the property that $E_{\Theta_{i_1}} \circ E_{\Theta_{i_2}} = E_{\Theta_{i_2}} \circ E_{\Theta_{i_1}}$, i.e. the non-linear parts commute, always leading to the same function $F_{\mathcal{K}} = t \circ E_{\Theta_{\sigma_j(r)}} \circ \cdots \circ E_{\Theta_{\sigma_j(1)}} \circ s$ independently of the order σ_j in which the rounds are given. The permutation σ_j on the order of the rounds is unique for each user and allows the authority to recover him. More precisely, to this aim of finding a user from his block cipher description, first, the authority computes in turn, for each $i \in \{1, \ldots, r\}$,

$$G_{1,j} \circ s^{-1} \circ {E_{\Theta_i}}^{-1}, \tag{1}$$

guessing the right value i by testing the simplicity of the result, i.e. by estimating the degree and the number of monomials. When $\sigma_j(1)$ has been found, the authority continues its procedure with $G_{2,j} \circ G_{1,j} \circ s^{-1} \circ {E_{\Theta_{\sigma_j(1)}}}^{-1} \circ {E_{\Theta_i}}^{-1}$, for $i \neq \sigma_j(1)$, trying to find back $\sigma_j(2)$, and so on, until the permutation σ_j is entirely recovered, see [2] for details.

3 Our protection in a nutshell

We write $\tilde{0}$ for a polynomial which often vanishes and $\tilde{P} = P + \tilde{0}$. By the way, $\tilde{S}$ stands for a system S of equations where some substitutions are made, replacing some polynomials P by $\tilde{P}$.

Example 1 *Over $GF(q)[X]$, we have $\tilde{0} = X^{q-1} - 1$.*

Our idea is to simply replace $G_{i,j}$ by $\widetilde{G_{i,j}}$, for $i = 1, \ldots, r$. This way, the IP problem structure of each round are made less accessible to an attacker.

The construction where only one description of a round is modified is mainly given for pedagogical purpose and as an introduction to Sect. 5.2. Actually, it conducts to wrong results.

In order to have a function which gives us always the correct result, we have to modify several instances of the block cipher. More precisely, we replace the system $G_{i,j}$ by 4 concurrent systems $\widetilde{G_{i,j}}$ where we can prove that two of them lead to what is expected. A majority vote allows to decide which result we have to retain. Note that this protection of one round can be seen as a protection of one IP-like instance, and this way, it could be applied to some other cryptographic schemes based upon IP.

4 Parasitizing the system with $\tilde{0}$-polynomials

Example 1 is not sufficient because it does not allow enough diversity to stay hidden from an attacker. In this section, we introduce new $\tilde{0}$-polynomials to this aim. We proceed following two steps.

First, we introduce a well-known class of polynomials, the q_0-polynomials. With them, we are able to compute polynomials which vanish on a predetermined set of points. However, as q_0-polynomials are univariate and strongly related to vector spaces, next, we have to compose them with random multivariate polynomials.

4.1 Linearized polynomials [11]

Definition 1 *For q_0 a power of 2 such that $q_0 \mid q$, a* q_0-polynomial *over $\mathbb{F} = GF(q)$ is a polynomial of the form $L(X) = \sum_{i=0}^{e} a_i X^{q_0{}^i}$, with $e \in \mathbb{N}$ and $(a_0, \ldots, a_e) \in \mathbb{F}^{e+1}$.*

Note that a q_0-polynomial L of degree q_0^e has at most $e + 1$ terms and a great number of roots in its splitting field. Indeed, if $a_0 \neq 0$, we see that L has only simple roots, so it has q_0^e zeroes in $\mathbb{F}$.

Example 2 *Let* $\mathrm{Tr} : x \mapsto \sum_{i=0}^{15} x^{2^i}$ *be the trace of $GF(2^{16})$ over $GF(2)$ and $\alpha \in GF(2^{16})$, then $L = Tr(\alpha.X)$ is a 2-polynomial with 16 terms and 2^{15} roots over $GF(2^{16})$.*

Proposition 1 *The set of a q_0-polynomial roots is a linear subspace of its splitting field, i.e. $L(X) = \sum_{i=0}^{e} a_i X^{q_0{}^i} = \prod_{\alpha \in V} (X - \alpha)^{\kappa}$ for V a linear* *subspace and some $\kappa \geq 1$. In fact, for a q_0-polynomial with simple roots, $\kappa = 1$.*

To count the number of q_0-polynomials with q_0^e roots of order 1, it suffices to count the number of $GF(q_0)$-subspaces of $GF(q)$ of dimension e:

Corollary 1 *For $q = q_0^m$, the number of q_0-polynomials with q_0^e roots of order 1 is equal to:*

$$\mathcal{G}(q_0, m, e) = \frac{(q_0^m - 1)\cdots(q_0^{m-e+1} - 1)}{(q_0^e - 1)\cdots(q_0 - 1)}.$$

Due to the finite field structure, it is clear that a q_0-polynomial has at most 2^{m-1} roots, so, if we want to construct $\tilde{0}$-polynomials with more roots, we need to multiply several q_0-polynomials together. But, there would be some intersection among the roots of different polynomials. Hence, to increase the number of roots more efficiently, we can combine some affine q_0-polynomials which are the relevant construction of q_0-polynomials with an affine set of roots.

Definition 2 *For q_0 a power of 2 such that $q_0 \mid q$,* an affine q_0-polynomial *over $\mathbb{F} = GF(q)$ is a polynomial of the form $A(X) = L(X) - \alpha$ where $\alpha \in \mathbb{F}$ and L is a q_0-polynomial.*

4.2 Multivariate lifting

In order to tranform a q_0-polynomial into a multivariate polynomial, we compose it naturally with a multivariate polynomial.

Let Q be an affine q_0-polynomial over $GF(q_0^m)$ which equals zero over the subspace U of dimension e, we construct a multivariate version of Q by choosing a multivariate polynomial $f \in GF(q_0^m)[X_1, \ldots, X_{n_f}]$ and computing $Q_f = Q(f(X_1, \ldots, X_{n_f}))$. In our context, two conditions have to be considered :

1. the resulting polynomial must have at least the same proportion $\frac{1}{2^{m-e}}$ of roots as Q,

2. Q_f should not have a large number of terms.

Hence, we restrict the choice for f so as to respect the previous conditions. In practice, we take a random f with a small number of terms and we check if at least $1/2^{m-e}$ points of $GF(q_0^m)^{n_f}$ have an image following f in U. So the polynomial Q_f will have more than $2^{m.n_f}/2^{m-e}$ roots.

Example 3 *If $Q = \mathrm{Tr}_{GF(2^4)/GF(2)}(X)$, Q has 8 roots in $GF(2^4)$. Then the polynomial $f(X_1, X_2) = X_1 + X_1.X_2$ of $GF(2^4)[X_1, X_2]$ gives a polynomial Q_f with at least 32 roots in $GF(2^4)^2$.*

Eventually, this method allows to obtain a multivariate polynomial and also to randomize the construction by breaking its linear structure.

5 Some practical considerations

In Sect. 5 of [2], the authors provide two examples of a system for 10^6 users.

In the first one, the base field is $GF(2^{16})$ and there are 5 variables. The block cipher has 32 rounds and each equation is homogeneous of degree 4, hence each round has at most 350 monomials, and there is at most 11200 monomials for the whole system. We will refer to this example as the Case 1.

In the second one, which we call Case 2, the base field is $GF(2^9)$, there are 19 variables, the block cipher has 33 rounds and each equation is homogeneous of degree 3. So each round and the system have, respectively, at most 25270 and 833910 monomials.

5.1 Protecting one round

In this section, we introduce a modified system leading to the correct result more than half time. In particular, we explain the interferences of our parasitic $\tilde{0}$ with the original public user representation; we show how we can choose some component H of $\tilde{0}$ to prevent an attacker to retrieve the original system.

Let $\tilde{0} = L(f(X_1, \ldots, X_{n_f}))H(X_1, \ldots, X_n)$ where

- L is a 2-polynomial with 2^{m-1} roots,
- f is a random polynomial of degree d_f in $2 \leq n_f \leq n$ variables and $t_f \geq 2$ terms such that $1/2$ of its values are roots of L,
- H is a random polynomial in n variables over $\mathbb{F}$ with t terms.

Proposition 2 *The polynomial $\tilde{0}$ has about $N_1(m, t, t_f)$ terms and at least $1/2$ of roots where*

$$N(m, t, t_f) = m \times t \times t_f.$$

We add a parasitic $\tilde{0}$ to every equation of the round, taking the same 2-polynomial L for all equations of a given round but with different random polynomials H. This method allows the construction of a round function $\widetilde{G_{i,j}}$ that gives the correct result with a probability greater than 1/2.

We introduce the polynomial H to generate enough monomials of degree d to avoid the capability of recovering P, a homogeneous multivariate polynomial of degree d, from the knowledge of $P+\tilde{0}$. In fact, starting from $P+\tilde{0}$, one can immediately compute the polynomial $\tilde{0}$ without its monomials of degree d, then knowing the form (i.e. designed as above) of $\tilde{0}$, one can try the two following ideas:

1. Guess the unknown monomials and their coefficients among all of the different possibilities, in order to obtain a polynomial with the same specific structure as $\tilde{0}$. There are $M_{n,d} = \binom{n+d-1}{d}$ monomials of degree d in n variables, so even if one guesses the number k of missing monomials, there would be $\binom{M_{n,d}}{k}q^k$ cases.

2. Analyse the terms of $P + \tilde{0}$ to guess the missing monomials, then, by deducing the generic form of H, try to find the missing coefficients by solving an overdefined system of equations, at least quadratic, in $t + l$ variables over $\mathbb{F}$ (where l is the number of variables coming from the unknown 2-polynomial of $\tilde{0}$ and from f). This kind of problem has been extensively studied these last years (see [5], [8] for example), and in general, one can not provide attacks in less than $q^{(t+l)/2}$, so we should consider t such that $q^t \geq 2^{160}$.

The choice of f and H is made in the following way: we choose f with at least one term of degree 1 in X_1 and if I is the set of $L(X_1)$ exponents, then we draw a polynomial H as

$$H(X_1, \ldots, X_n) = \sum_{i \in I \cap \{1,\ldots,2^m-1\}} h_i(X_1, \ldots, X_n){X_1}^{2^m - i.d_f},$$

where the $h_i \in \mathbb{F}[X_1, \ldots, X_n]$ are homogeneous of degree $d-1$. For each i, let t_i be the number of terms of h_i, then H has nearly $t = \sum_i t_i$ terms and the product $L(f(X_1, \ldots, X_{n_f}))H(X_1, \ldots, X_n)$ has at least t monomials of degree d. Hence, the number of monomials k which are masking the original polynomial P is greater than t, so a choice of t, such that $q^t \geq 2^{160}$ to avoid the second strategy above, allows also to thwart the first idea.

Furthermore, the number of choices for f and H is very large and so the amount of ways to interfere an equation is large enough.

Let us apply our strategy to the two practical examples of [2]:

- Case 1: We choose L, f, H such that $t_f = 2$ and $t = 10$, as described above. This implies $N(16, 10, 2) = 320$ terms more for each equation, and thus 1600 terms more for one round $\widetilde{G_{i,j}}$. This represents nearly 6 times the size of the original round.
- Case 2: For $t_f = 3$ and $t = 18$ such that $q^t \geq 2^{160}$, we have $N(9, 18, 3) = 486$ more terms for each equation. The resulting $\widetilde{G_{i,j}}$ has hence around 1,4 times the size of $G_{i,j}$.

Remark 2 *Roughly counting, there are more than $\Lambda = G(2, m, m-1) \times \binom{d_f}{t_f - 1} \times 2^{m^{t+t_f}}$ different ways to interfere an equation with such polynomials $\tilde{0}$. In case 1, $\Lambda \geq 2^{208}$, and in case 2, $\Lambda \geq 2^{189}$.*

5.2 Getting the correct value

For a given round $G_{i,j}$, we use four parallel modified descriptions $\widetilde{G_{i,j}}$ with correlated $\tilde{0}$-polynomials to recover the expected result.

To achieve this goal, we partition $\mathbb{F}$ and construct $\tilde{0}$-polynomials accordingly. As shown in Sect. 5.1, it is possible to cover more than half of $\mathbb{F}$. So, we partition $\mathbb{F}$ twice into two sets of same size $\mathbb{F} = E_1 \cup \overline{E_1} = E_2 \cup \overline{E_2}$ and we construct $\tilde{0}_1$, $\overline{\tilde{0}_1}$, $\tilde{0}_2$ and $\overline{\tilde{0}_2}$ such that the polynomial $\tilde{0}_\kappa$ (resp. $\overline{\tilde{0}_\kappa}$) vanishes over E_κ (resp. over $\overline{E_\kappa}$), $\kappa = 1$ or 2.

With this construction, for any input value, there is always two $\tilde{0}$-polynomials which vanish and so at least two descriptions $\widetilde{G_{i,j}}$ which give the expected result. Furthermore, as the constuction of an $\tilde{0}$-polynomial is partially random (see Sect. 5.1), the non-zero values of the two other $\tilde{0}$-polynomials look like random ones. Hence, with an overwhelming probability, the two other descriptions take 2 different results and so we can easily decide which value is correct according to a majority decision.

5.3 The final construction

Our new description of the entire public representation consists thus in modifying each round independently as described in Sect. 5.2. We obtain four parallel systems, with a majority vote at each level to decide which value has to be sent to the next round. See Fig. 2 for the resulting description.

Then, the size of this description according to the two practical examples of [2] is:

- Case 1: For the same choice of parameters as in Sec. 5.1, we have 1600 terms more for one round, i.e at most 1950 terms for each round.

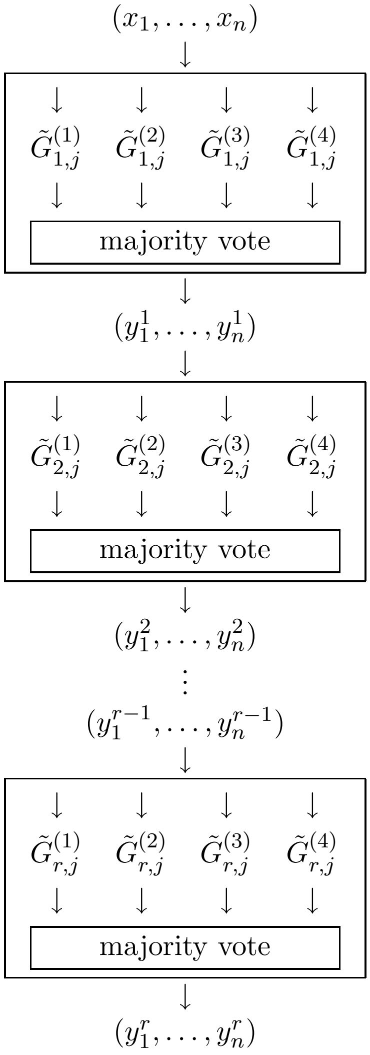
$(x_1, \dots, x_n)$
$\tilde{G}^{(1)}_{1,j}$ $\tilde{G}^{(2)}_{1,j}$ $\tilde{G}^{(3)}_{1,j}$ $\tilde{G}^{(4)}_{1,j}$
majority vote
$(y^1_1, \dots, y^1_n)$
$\tilde{G}^{(1)}_{2,j}$ $\tilde{G}^{(2)}_{2,j}$ $\tilde{G}^{(3)}_{2,j}$ $\tilde{G}^{(4)}_{2,j}$
majority vote
$(y^2_1, \dots, y^2_n)$
$\vdots$
$(y^{r-1}_1, \dots, y^{r-1}_n)$
$\tilde{G}^{(1)}_{r,j}$ $\tilde{G}^{(2)}_{r,j}$ $\tilde{G}^{(3)}_{r,j}$ $\tilde{G}^{(4)}_{r,j}$
majority vote
$(y^r_1, \dots, y^r_n)$

Figure 2: New public representation

Thus, the final function (with 4 parallel systems of 32 rounds) contains around 22 times more terms than the original description.

- Case 2: Here, each equation contains at most 1816 monomials which leads to a final description with nearly 6 times the size of the original representation.

5.4 Tracing procedure

Following [1], the authority can trace back pirates by looking at correlations between differential characteristics of the input and differential characteristics of the output. Thus, such a procedure relies only on the evaluation of rounds at given input contrary to the procedure described in [2] which is based on polynomials compositions.

This procedure, via evaluations, is still compatible with our new description and can be used by the authority to trace back the traitors.

Acknowledgments. The authors thank gratefully Stéphanie Alt and Reynald Lercier for their comments on the tracing procedure.

References

[1] Stéphanie Alt and Reynald Lercier, private communication.

[2] Olivier Billet and Henri Gilbert, *A Traceable Block Cipher*, Advances in Cryptology – ASIACRYPT 2003 (C.S. Laih, Ed.), vol. 2894, 2003, pp. 331–346.

[3] Dan Boneh, and Mathew Franklin, *An efficient public key traitor tracing scheme*, Advances in Cryptology – CRYPTO'99 (Michael J. Wiener, Ed.), vol. 1666, 1999, pp. 338–353.

[4] Benny Chor, Amos Fiat, and Moni Naor, *Tracing Traitors*, Advances in Cryptology – CRYPTO'94 (Yvo Desmedt, Ed.), vol. 839, 1994, pp. 257–270.

[5] Nicolas Courtois, A. Klimov, Jacques Patarin, and Adi Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, Advances in Cryptology – EUROCRYPT 2000, vol. 1807, 2000, p. 392 ff.

[6] Jintai Ding, *A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation*, Public Key Cryptography, 2004, pp. 305–318.

[7] Jintai Ding, Jason E. Gower and Dieter S. Schmidt, *Zhuang-Zi: A New Algorithm for Solving Multivariate Polynomial Equations over a Finite Field*, Cryptology ePrint Archive, Report 2006/038, 2006. http://eprint.iacr.org/.

[8] Jean-Charles Faugère and Antoine Joux, *Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases*, Advances in Cryptology – CRYPTO'03 (Dan Boneh, Ed.), vol. 2729, 2003, pp. 44-60.

[9] Jean-Charles Faugère and Ludovic Perret, private communication, February 2006.

[10] Aggelos Kiayias and Moti Yung, *Traitor Tracing with Constant Transmission Rate*, Advances in Cryptology – EUROCRYPT 2002 (Lars Knudsen, Ed.), vol. 2332, 2002, pp 450–465.

[11] Rudolf Lidl and Harald Niederreiter, *Intoduction to finite fiels and their applications*, Cambridge University Press, 1986.

[12] Jacques Patarin, Louis Goubin, and Nicolas Courtois, *Improved Algorithms for Isomorphisms of Polynomials*, Advances in Cryptology – EUROCRYPT'98 (Kaisa Nyberg, Ed.), vol. 1403, 1998, pp. 184–200.