The Sidelnikov-Shestakov's Attack applied to the Chor-Rivest Cryptosystem

Sylvain Colin & Gaspard Férey

March 3, 2014

Contents

Abstract

In this article, we discuss about the Sidelnikov-Shestakov Attack on cryptosystems based on Reed-Solomon codes. Then we describe how this algorithm can be used to attack the Chor-Rivest Cryptosystem.

1 Introduction

intro

1.1 Our Work

2 Preliminaries

efnot

2.1 A cryptosystem based on Reed-Solomon codes

We study here the public-key cryptosystem introduced by Niederreiter [?] applied to the generalized Reed-Solomon codes. Let $\mathbb{F}_q$ be a finite field with $q = p^h$ elements and $\mathbb{F} = \mathbb{F}_q \cup \{\infty\}$, where ∞ has natural properties ($1/\infty = 0$, etc). We call $\mathfrak{A}$ the following matrix:

$$\mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n) := \begin{pmatrix} z_1\alpha_1^0 & z_2\alpha_2^0 & \cdots & z_n\alpha_n^0 \\ z_1\alpha_1^1 & z_2\alpha_2^1 & \cdots & z_n\alpha_n^1 \\ & & \ddots & \\ z_1\alpha_1^{k-1} & z_2\alpha_2^{k-1} & \cdots & z_n\alpha_n^{k-1} \end{pmatrix} \in \mathcal{M}_{\mathbb{F}_q}(k, n)$$

In the considered cryptosystem, the secret key consists of

- $(\alpha_i)_{1 \leq i \leq n}$ distinct elements of $\mathbb{F}_q$.
- $(z_i)_{1 \leq i \leq n}$ elements of $\mathbb{F}_q - \{0\}$.
- a random nonsingular $k \times k$-matrix H over $\mathbb{F}$.

The public key is

- the representation of the field $\mathbb{F}$. In particular the polynomial used to define $\mathbb{F}$ is public.
- The two integers k and n such that $0 < k < n \leq q$.
- $M := H \cdot \mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n)$.

2.2 Equivalence between Reed-Solomon codes

Sidelnikov and Shestakov show [?] that for all $a \in \mathbb{F}_q - \{0\}$ and $b \in \mathbb{F}_q$, there exists $H_1, H_2, H_3 \in \mathcal{M}_{F_q}(k, k)$ invertible such that

$$\begin{aligned} H_1\mathfrak{A}(a \cdot \alpha_1 + b, \ \dots, a \cdot \alpha_n + b, c_1 z_1, \ \dots, c_n z_n) &= \mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n) \\ H_2\mathfrak{A}\left(\frac{1}{\alpha_1}, \ \dots, \frac{1}{\alpha_n}, d_1 z_1, \ \dots, d_n z_n\right) &= \mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n) \\ H_3\mathfrak{A}\left(\alpha_1, \ \dots, \alpha_n, a \cdot z_1, \ \dots, a \cdot z_n\right) &= \mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n) \end{aligned}$$

This means that for any cryptosystem $M = H\mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n)$, for any birationnal transformation

$$\phi : x \mapsto \frac{ax + b}{cx + d}$$

$M = H_\phi \mathfrak{A}(\phi(\alpha_1), \ \dots, \phi(\alpha_n), z_1', \ \dots, z_n')$ and by using the unique transformation ϕ that maps $(\alpha_1, \alpha_2, \alpha_3)$ to $(0, 1, \infty)$, we get that for any cryptosystem $M = H\mathfrak{A}(\alpha_1, \ \dots, \alpha_n, z_1, \ \dots, z_n)$, M can be uniquely written

$$M = H'\mathfrak{A}(0, 1, \infty, \alpha_4', \ \dots, \alpha_n', 1, z_2', \ \dots, z_n')$$

with H' nonsingular, $z_i' \neq 0$ and α_i distincts elements of $\mathbb{F}_q - \{0, 1, \infty\}$.

3 Attack of Sidelnikov-Shestakov

The attack of Sidelnikov-Shestakov consist of the following steps.

First we realize that the public key can be uniquely written as in the previous section.

$$M = H'\mathfrak{A}(0, 1, \infty, \alpha'_4, \ \dots \ , \alpha'_n, 1, z'_2, \ \dots \ , z'_n)$$

We compute then the echelon form of M.

$$E(M) = \begin{pmatrix} 1 & 0 & \cdots & 0 & b_{1,k+1} & \cdots & b_{1,n} \\ 0 & 1 & \cdots & 0 & b_{2,k+1} & \cdots & b_{2,n} \\ & & \ddots & & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & b_{k,k+1} & \cdots & b_{k,n} \end{pmatrix} = H'' \cdot M$$

For $2 \leq k \leq n-2$, this attack works with a complexity of ...

4 Application to the Chor-Rivest Cryptosystem

4.1 The Chor-Rivest Cryptosystem

Secret keys consist of

- an element $t \in \mathrm{GF}(q)$ with algebraic degree h
- a generator g of $\mathrm{GF}(q)^*$
- an integer $d \in \mathbb{Z}_{q-1}$
- a permutation π of $\{0, ..., p-1\}$.

Public keys consist of all

$$c_i = d + \log_g(t + \alpha_{\pi(i)}) \mod q-1$$

The message consists in a bitstring $m = [m_0 ... m_{p-1}]$ of length p such that $\sum_i m_i = h$. The ciphertext is

$$E(M) := \sum_{i=0}^{p-1} m_i c_i$$

To decipher this message, we compute

$$g^{E(M)-hd} = \prod_i \left(t + \alpha_{pi(i)}\right)^{c_i}$$

When we attack this cryptosystem, we can consider a generator $g_0 = g^u$ with u unknown and $\gcd(u, q-1) = 1$ we then have

$$g_0^{c_i} = \left(g^d \left(t + \alpha_{\pi(i)}\right)\right)^u = \left(A + \alpha_{\pi(i)} \cdot B\right)^u$$

We can then consider that the secret key is

- $A \in \mathbb{F}_q$.
- $B \in \mathbb{F}_q$ such that $t = A \cdot B^{-1}$ has algebraic degree h.

- $0 < u < q-1$ prime with $q-1$.
- the permutation π of $\{0, ..., p-1\}$.

and public key consists in all the

$$d_i := \left(A + \alpha_{\pi(i)} \cdot B\right)^u \in \mathbb{F}_q$$

The ciphertext becomes

$$E'(M) := \prod_{i=0}^{p-1} d_i^{m_i} = g^{uE(M)} = B^{uh} \left(\prod_i \left(t + \alpha_{pi(i)}\right)^{c_i} \right)^u$$

Knowing u, B and h, it is easy to compute from $E'(M)$, the following quantity

$$\prod_i \left(t + \alpha_{pi(i)}\right)^{c_i}$$

which allow us to retrieve all the c_i.

4.2 A First Attack using Reed-Solomon codes

We have for all j

$$g^{c_j} = g^d \cdot (t + \alpha_{\pi(j)}) = A + \alpha_{\pi(j)} \cdot B$$

where $\alpha_{\pi(j)} \in \mathrm{GF}(p)$ and A and B are elements of $\mathrm{GF}(p^h) \subset \mathrm{GF}(p)[X]$ and can be seen as polynomials of the variable X with coefficients in $\mathrm{GF}(p)$. Then if we consider an other generator g_0 of $\mathrm{GF}(q)$, we have $g_0 = g^u$ and

$$g_0^{c_j} = \left(A(X) + \alpha_{\pi(j)} \cdot B(X)\right)^u \mod \mu(X)$$

where μ is the polynomial of degree h defining the field $\mathrm{GF}(q)$.

As an attempt to guess g, we can choose a random generator g_0 and compute the quantities

$$g_0^{c_j} = \sum_{i=0}^{h-1} P_i(\alpha_{\pi(j)}) X^i$$

where P_i is a polynomial with coefficients in $\mathrm{GF}(p)$. P_i depends on $A(X)$, $B(X)$, u and obviously on i. However, P_i does not depend on j.

Besides, we have

- $\deg P_i \leq u$ since the coefficients of $\left(A(X) + \alpha_{\pi(j)} \cdot B(X)\right)^u$ seen in $\mathrm{GF}(p)[X]$ are polynomials of degree smaller than u in $\alpha_{\pi(j)}$.
  When we compute the remain in the division of this polynomial by $\mu(X)$, these coefficients remain polynomials of degree smaller than u in $\alpha_{\pi(j)}$.
- $\deg P_i < p$ since $\alpha_{\pi(j)}^p = \alpha_{\pi(j)}$.

We now consider the matrix

$$\mathfrak{A} := \left(\alpha_{\pi(j)}^i\right)_{0 \leq i,j \leq p-1} \in \mathcal{M}_{\mathbb{F}_p}(p, p)$$

We call

- $P_i[j] \in \mathrm{GF}(p)$ the j-th coefficient of the polynomial P_i.

- $H = (P_i[j])_{i,j} \in \mathcal{M}_{\mathbb{F}_p}(h,p)$
- $M = \left(P_i(\alpha_{\pi(j)})\right)_{i,j} \in \mathcal{M}_{\mathbb{F}_p}(h,p)$.

$$H \cdot \mathfrak{A} = M$$

We suppose now that we try to guess the private generator g but only find a generator g_0 such that $g_0 = g^u$ with $u < h$.

We can compute the elements $g_0^{c_j} \in \mathrm{GF}(q)$, the coefficients $P_i(\alpha_{\pi(j)}) \in \mathrm{GF}(p)$ and eventually the matrix M.

Since $\deg P_i \leq u$, we know that only the u first columns of the matrix H are non zero. Therefore we consider now the matrix H' build from the u first columns of H (the other columns being equal to 0) and $\mathfrak{A}'$ the u first rows of $\mathfrak{A}$. We get

$$H' \cdot \mathfrak{A}' = M$$

We suppose now that the first u rows of M are linearly independent. This allow us to consider only the first u lines of the matrices H' and M (H'' and M'') which gives us

$$H'' \cdot \mathfrak{A}' = M''$$

with

- $H'' \in \mathcal{M}_{\mathbb{F}_p}(u,u)$
- $\mathfrak{A}' \in \mathcal{M}_{\mathbb{F}_p}(u,p)$
- $M'' \in \mathcal{M}_{\mathbb{F}_p}(u,p)$

We use then the attack described in the first section to compute $\mathfrak{A}'$ which yields the permutation π.

4.2.1 Problem

It seems quite unlikely that $g_0 = g^u$ with a small u. Indeed, there are $\phi(p^h - 1)$ generators which is comparable to p^h and the order of h is only (in the suggested parameters) around 24.

This could be solved if we had a way to rapidly check whether one generator is a small power of an other.

4.2.2 Further...

If u is a small multiple of p, the previous arguments still apply since then $u = pu'$ with $u' < h$ and we get

$$g_0^{c_j} = \left(\left(A(X) + \alpha_{\pi(i)}B(X)\right)^p\right)^{u'} = \left(A^p(X) + \alpha_{\pi(i)}^p B^p(X)\right)^{u'} = \left(A'(X) + \alpha_{\pi(i)}B'(X)\right)^{u'}$$

This only changes the polynomials A and B but still allow to compute the permutation $\pi(i)$ on these conditions.

We actually also have this conclusion if u is a small multiple of p^r for all $0 \leq r < h$. In fact a condition for the previous to work is that when u is written in base p, the sum of its digits does not exceed h.

Remains to see how many different u this methods allows to check... Is it reasonable to try this method with several value for g_0 until we find g ? I guess not...

Besides, as explained in Sidelnikov and Shestakov's article, if the previous reasoning excludes a set of candidates u_i, it also excludes $p \cdot u_i$ and even $p^r \cdot u_i$ for all $0 \leq r < h$. Actually, this doesn't excludes any more candidate since the writing of $p \cdot u_i$ and u^i modulo $p^h - 1$ in base p are just rotated.

5 Conclusions