

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

Sylvain Colin & Gaspard Férey

March 4, 2014

Contents

1	Introduction	1
1.1	Our Work	1
2	Preliminaries	1
2.1	A cryptosystem based on Reed-Solomon codes	1
2.2	Equivalence between Reed-Solomon codes	2
3	The Sidelnikov-Shestakov Attack	2
4	Application to the Chor-Rivest Cryptosystem	3
4.1	The Chor-Rivest Cryptosystem	3
4.2	A First Attack using Reed-Solomon codes	4
4.2.1	Problem	5
4.2.2	Further...	5
5	Conclusions	6

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

1.1 Our Work

2 Preliminaries

2.1 A cryptosystem based on Reed-Solomon codes

We study here the public-key cryptosystem introduced by Sidelnikov and Shestakov [1] 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 usual properties ($1/\infty = 0$, etc). We call \mathfrak{A} the following matrix:

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

where $\alpha_i \in \mathbb{F}$ and $z_i \in \mathbb{F}_q \setminus \{0\}$ for all $i \in \{1, \dots, n\}$. Note that, if $\alpha_i = \infty$, we replace the i^{th} column by the vector $z_i(0, \dots, 0, 1)^T$, so that all the coefficients of the matrix are finite.

In the considered cryptosystem, the secret key consists of

- The set $\{\alpha_1, \dots, \alpha_n\}$;
- The set $\{z_1, \dots, z_n\}$;
- A random nonsingular $k \times k$ -matrix H over \mathbb{F}_q .

The public key is

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

The codewords are then the vectors $c = b.M$ where $b \in \mathbb{F}_q^k$. So, the different codewords have necessarily the following form :

$$c = (z_i f_c(\alpha_i))_{1 \leq i \leq n}$$

where f_c is a polynomial whose degree is at most $k - 1$.

Thus, given a message to send, which is actually a vector b of \mathbb{F}_q^k , one will have to transmit the vector $b.M + e$ where e is a random vector of \mathbb{F}_q^n with Hamming weight at most $t = \lfloor \frac{n-k}{2} \rfloor$. So, since a GRS code correct at most $t = \lfloor \frac{n-k}{2} \rfloor$ error, the original message can be recovered by computing $b' = b.M$, finding the closest codeword from the received message, and then computing $b'M^{-1}$. However, the original message can not be easily recovered when not knowing the GRS code used in the secret key.

2.2 Equivalence between Reed-Solomon codes

Sidelnikov and Shestakov show [1] that for all $a \in \mathbb{F}_q \setminus \{0\}$ and $b \in \mathbb{F}_q$, there exists $H_1, H_2, H_3 \in \mathcal{M}_{\mathbb{F}_q}(k, k)$ invertible such that

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

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

$$M = H' \mathfrak{A}(0, 1, \infty, \alpha'_4, \dots, \alpha'_n, z'_1, \dots, z'_n)$$

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

So, when M is given, it is impossible to compute the original matrices \mathfrak{A} and H since many pairs of such matrices lead to the same public matrix M . However, computing an equivalent pair is sufficient since it will allow to decipher the messages as well as the original secret pair of matrices. So, the attack will consist of finding H and $\mathfrak{A}(0, 1, \infty, \alpha'_4, \dots, \alpha'_n, z'_1, z'_2, \dots, z'_n)$, equivalent to the original pair. We can also assume that $z'_1 = 1$. Indeed, if we multiply all the elements z'_i by a factor $a \in \mathbb{F}_q$ and all the elements of the matrix H by a^{-1} , the resulting matrix M will be the same.

3 The Sidelnikov-Shestakov Attack

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

First we assume that the public key is as described 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 .

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

Since the echelon form can be computed only with left multiplication of the matrix M , the k lines of $E(M)$ are codewords. As a consequence, if we call f_i the polynomial associated to the i^{th} line, we have :

- $\forall 1 \leq i \leq k, f_i(\alpha_i) = 1$
- $\forall 1 \leq i \neq j \leq k, f_i(\alpha_j) = 0$
- $\forall 1 \leq i \leq k \forall k+1 \leq j \leq n, f_i(\alpha_j) = b_{i,j}$

So, since all the α_i are different, the polynomial f_i has $k - 1$ simple roots. As a consequence, $b_{i,j} \neq 0$ for all $1 \leq i \leq k$ and $k+1 \leq j \leq n$. Moreover, we know the general form of the polynomial f_i :

$$f_i(X) = c_i \cdot \prod_{1 \leq j \leq k, i \neq j} (X - \alpha_j)$$

where $c_i \in \mathbb{F} \setminus 0$.

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

Public keys consist of all

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

The message consists in a bitstring $m = [m_0 \dots m_{p-1}]$ 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

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

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

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

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 $\{0, \dots, p-1\}$.

and public key consists in all the

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

The ciphertext becomes

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

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

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

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

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

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

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 $\text{GF}(p)$. 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 $(A(X) + \alpha_{\pi(j)} \cdot B(X))^u$ seen in $\text{GF}(p)[X]$ 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

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

We call

- $P_i[j] \in \text{GF}(p)$ the j -th coefficient of the polynomial P_i .
- $H = (P_i[j])_{i,j} \in \mathcal{M}_{\mathbb{F}_p}(h, p)$
- $M = (P_i(\alpha_{\pi(j)}))_{i,j} \in \mathcal{M}_{\mathbb{F}_p}(h, p)$.

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

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 $g_0^{c_j} \in \text{GF}(q)$, the coefficients $P_i(\alpha_{\pi(j)}) \in \text{GF}(p)$ 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 another.

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

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

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

References

- hes92 [1] V. M. Sidelnikov and S. O. Shestakov. On insecurity of cryptosystems based on generalized reed-solomon codes. *Discrete Math. Appl.*, 2(4).