

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

Sylvain Colin & Gaspard Férey

March 8, 2014

Contents

1	Introduction	1
1.1	Our Work	1
2	Preliminaries	2
2.1	A cryptosystem based on Reed-Solomon codes	2
2.2	Equivalence between Reed-Solomon codes	2
3	The Sidelnikov-Shestakov Attack	4
4	Application to the Chor-Rivest Cryptosystem	5
4.1	The Chor-Rivest Cryptosystem	5
4.2	Link with Reed-Solomon codes	6
4.3	A First Attack using Reed-Solomon codes	6
4.4	Small power of g	6
4.5	Wider set of generators	7
5	Vaudenay attack	8
5.1	Generating more rows...	8
5.2	Simulation	9
5.2.1	Number of linearly independent polynomials	10
6	Draft	10
7	Conclusions	11

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 improve the attack to the Chor-Rivest Cryptosystem proposed by Vaudenay [\[2\]](#).

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 $t \in \mathbb{F}_q$ with algebraic degree h
- a generator g of \mathbb{F}_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 Link with Reed-Solomon codes

Trying to attack this cryptosystem show some relations between this problem and the previous one studied in section [2](#). ^{sec:Pre1}In particular we have the following theorem.

Theorem 1. *Let $2 \leq k \leq p-2$. Suppose there exists $(Q_i)_{0 \leq i \leq k-1}$ k polynomials of $\mathbb{F}_p[X]$ linearly independent with degree smaller than $k-1$. Suppose the evaluations $m_{i,j} := Q_i(\alpha_{\pi(j)})$ is known for all i and j . Then the permutation π can be recovered in polynomial time using a Sidelnikov-Shestakov attack on the matrix $M = (m_{i,j})_{i,j} \in \mathcal{M}_{k,p}(\mathbb{F}_p)$.*

Proof. We suppose here that one of the Q_i has a degree exactly k . Then we write the square non singular matrix $H = (h_{i,j}) \in \mathcal{M}_k(\mathbb{F}_p)$ of the coefficients of the Q_i

$$Q_i(X) = \sum_{j=0}^{k-1} h_{i,j} X^j$$

If we still consider

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

We have the equality

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

with H non singular and since $k \leq p-2$, this is exactly the public key of a cryptosystem based on the Reed-Solomon codes described in section [2](#). ^{sec:Pre1}□

So a possible way to attack the Chor-Rivest cryptosystem would be to find the evaluations of enough small degree polynomials in the $\alpha_{\pi(i)}$.

4.3 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 $\alpha_{\pi(j)} \in \mathbb{F}_p$ and A and B are elements of \mathbb{F}_{p^h} .

A naive attack would be then to try to guess at random the generator g . We will see that although finding the precise g is very unlikely, there is a family of generators that can still allow us to retrieve π .

4.4 Small power of g

As an attempt to guess g , we can choose a random generator g_0 of \mathbb{F}_q^* . We have $g_0 = g^u$ and

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

if we write this quantity in a certain base $(e_i)_{1 \leq i \leq h}$ of \mathbb{F}_{p^h} , we notice that each coordinate is a polynomial Q_i in the $\alpha_{\pi(j)}$.

$$g_0^{c_j} = \sum_{i=1}^h Q_i(\alpha_{\pi(j)}) e_i$$

where Q_i has its coefficients in \mathbb{F}_p . Q_i depends on A , B , u and obviously on i . However, Q_i does not depend on j .

Besides, we have

- $\deg Q_i \leq u$

- $\deg Q_i < p$ since $\alpha_{\pi(j)}^p = \alpha_{\pi(j)}$.

This means that we have access to the evaluations in the $\alpha_{\pi(j)}$ of h polynomials of degree smaller than u . According to Theorem [1](#), a sufficient condition for this attack to work is $u \leq h - 1 \leq p - 3$. The last inequality is most likely true since h is chosen close to $p/\log p$. However there are only $h - 1$ different elements of \mathbb{F}_{p^h} fulfilling the first one. This only slightly improves the exhaustive research of g .

4.5 Wider set of generators

We can notice that if $u = u'p$,

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

and the coordinates of this quantity are polynomials of degree u' in $\alpha_{\pi(j)}$. This also means that if u is written $u = \sum_{i=0}^{h-1} u_i p^i$ in base p , then

$$g_0^{c_j} = \prod_{i=0}^{h-1} (A^{p^i} + \alpha_{\pi(j)} \cdot B^{p^i})^{u_i}$$

whose coordinates are a polynomial of degree $w_p(u) := \sum_{i=0}^{h-1} u_i$ in the $\alpha_{\pi(j)}$.

This mean that all u such that $w_p(u) < h$ allow to retrieve the permutation and break the cryptosystem. The number of such u is

$$\left(\binom{h+1}{h-1} \right) = \binom{2h}{h-1} = \Theta(4^h \sqrt{h})$$

This is a drastic improvement in the exhaustive research of g . However this remains quite small compared to the number $\phi(p^h - 1)$ of different generators in \mathbb{F}_{p^h} which is comparable to p^h .

5 Vaudenay attack

We can see that the previous attack requires to find a generator of \mathbb{F}_q among the elements that can be written g^u with $w_p(u) \leq h$. Since there are very few of such elements compared to the $\phi(p^h)$ different generators of \mathbb{F}_q , Vaudenay suggests [2] to consider a generator g_{p^r} of the sub-field \mathbb{F}_{p^r} of \mathbb{F}_{p^h} . He introduces the following theorem

Theorem 2. *For any factor r of h , there exists a generator g_{p^r} of the multiplicative group of the subfield \mathbb{F}_{p^r} of \mathbb{F}_q and a polynomial Q with degree h/r whose coefficients are in \mathbb{F}_{p^r} and such that $-t$ is a root and for all j , we have $Q(\alpha_{\pi(j)}) = g_{p^r}^{c_j}$.*

If we chose a base $(e_i)_{1 \leq i \leq r}$ of \mathbb{F}_{p^r} , we can write the coefficients of $g_{p^r}^{c_j}$ in this base as polynomials Q_i in $\alpha_{\pi(j)}$. We get

$$g_{p^r}^{c_j} = \sum_{i=1}^r Q_i(\alpha_{\pi(j)}) e_i$$

with $\deg Q_i \leq h/r$. We get the evaluation of r polynomials of degree smaller than h/r .

This means that instead of searching a generator among the approximately p^h generators of \mathbb{F}_q , we could search only within \mathbb{F}_{p^r} with r as small as possible. We notice that to be able to apply Theorem 2, we must have $h/r < r$. This yields the following theorem.

Theorem 3. *When $r > \sqrt{h}$, there exists a polynomial "known g_{p^r} " attack on the Chor-Rivest cryptosystem.*

This theorem is basically the main result from Vaudenay's article [2]. It states that to attack the Chor-Rivest cryptosystem, one can only search for the generator g_{p^r} among $\Theta(p^r)$ possible choices instead of the exhaustive research for g ($\Theta(p^h)$ choices).

However the "known g_{p^r} attack" suggested in Vaudenay's article can be improved in two ways.

- First it is only polynomial when $r > \sqrt{h}$. We will see that using a Reed-Solomon attack, we can still retrieve π in polynomial time provided we manage to get enough linearly independent polynomials in the $\alpha_{\pi(j)}$ which is possible even for small values of r using $g_{p^r}^u$ with u small enough.
- Besides, the Reed-Solomon attack doesn't require the knowledge of all the c_j . Only $O(h)$ (or a little more when considering the attack for r small) of them. So actually only a small proportion of them is enough. This makes this attack strong.

5.1 Generating more rows...

When we find g_{p^r} such that $g_{p^r}^{c_j} = Q(\alpha_{\pi(j)})$ where $\deg Q \leq h/r$, we only have r rows corresponding to the r different polynomials of the coordinates of $g_{p^r}^{c_j}$ in a certain base of \mathbb{F}_{p^r} . Being able to generate more row would allow to chose a lower r and improve drastically the attack.

We could consider now the coordinates of $g_{p^r}^{uc_j}$ for u such that $w_p(u)$ remains small. For example, we could decide $w_p(u) \leq w$.

This should yield (assuming $w \gg r$) up to

$$\left(\binom{r+1}{w} \right) = \binom{r+w+1}{w} = \binom{r+w+1}{r+1} = \Theta \left(\frac{w^{r+1}}{(r+1)!} \right)$$

different polynomials of degree at most $w \cdot \frac{h}{r}$. Unfortunately, it seems probable that these are strongly linearly dependent...

For example, the coordinates of $g_{p^r}^{pc_j}$ are linearly dependent on the coordinates of $g_{p^r}^{c_j}$. Indeed, when decomposed in a normal base of \mathbb{F}_{p^r} , these two sets of vectors of coordinates only differs by a rotation.

5.2 Simulation

We run a simulation with the following parameters

- $p = 197$, $h = 24$, $r = 3 < \sqrt{h}$.
- We define \mathbb{F}_q as the quotient of $\mathbb{F}_p[X]$ by the polynomial

$$\begin{aligned} X^{24} &+ 192X^{23} + 152X^{22} + 25X^{21} + 75X^{20} + 67X^{19} + 92X^{18} + 23X^{17} + 45X^{16} + 97X^{15} \\ &+ 2X^{14} + 21X^{13} + 106X^{12} + 130X^{11} + 128X^{10} + 136X^9 + 195X^8 + 95X^7 + 155X^6 \\ &+ 34X^5 + 51X^4 + 180X^3 + 97X^2 + 23X + 87 \end{aligned}$$

- We choose $g := X + 2$ the private multiplicative generator.
- We compute

$$\begin{aligned} g_{p^r} &= g^{\frac{p^h-1}{p^r-1}} \\ &= 153X^{23} + 168X^{22} + 167X^{21} + 45X^{20} + 128X^{19} + 68X^{18} + 103X^{17} + 11X^{16} \\ &\quad + 139X^{15} + 190X^{14} + 75X^{13} + 73X^{12} + 190X^{11} + 64X^{10} + 173X^9 + 34X^8 \\ &\quad + 88X^7 + 30X^6 + 139X^5 + 146X^4 + 111X^3 + 80X^2 + 136X + 48 \end{aligned}$$

- We choose different values for d , t and π , the results remain the same.
- We choose the base $(g_{p^r}, g_{p^r}^p, g_{p^r}^{p^2})$ for the \mathbb{F}_p -vector space \mathbb{F}_{p^r} (but this is of no influence on the results).
- We choose $(u_i)_{1 \leq i \leq 11} = (1, 2, p+1, 3, 2p+1, p+2, 4, p+3, 3p+1, 2p+2, 2p^2+p+1)$
We have $w_p(u_i) \leq 4$ so the coordinates of $g_{p^r}^{u_i c_j}$ are polynomials of degree smaller than $4h/r + 1 = 33$ in the $\alpha_{\pi(j)}$.

As a result, we obtain $11 \times 3 = 33$ lines of coordinates linearly independent (the simulation is the verification of this independence). This would allow the attack to retrieve the permutation π using the attack on the cryptosystem based on Reed-Solomon codes.

This proves that it is possible to duplicate the number of lines at the expense of the degree of the polynomials considered. Therefore the condition $r \geq \sqrt{h}$ is not absolutely necessary and we can hope to get a (far) smaller lower bound on r .

On the following tabular, we present the results of several simulations for different value of p , h . Each time, we choose the smallest value for r that allow an attack on the system. We display here the maximum weight w chosen for the exponents u_i of g_{p^r} and the number of linearly independent lines these $g_{p^r}^{u_i}$ allow to generate (should be $w \cdot \frac{h}{r} + 1$).

p	h	r	$w_p(u_i)$	Number of linearly independent lines
197	24	2	w	$\frac{w(w+3)}{2} < 12w + 1$ Attack impossible.
197	24	3	4	33
197	24	4	2	13
197	24	$r \geq 6$	1	$1 + h/r$
193	36	2	w	$\frac{w(w+3)}{2} < 18w + 1$ Attack impossible.
193	36	3	6	73
193	36	4	3	28
193	36	6	2	13
193	36	$r \geq 9$	1	$1 + h/r$
251	60	2	w	$\frac{w(w+3)}{2} < 30w + 1$ Attack impossible.
251	60	3	8	161
251	60	4	4	61
251	60	5	3	37
251	60	6	2	21
251	60	$r \geq 10$	1	$1 + h/r$

When we choose, $r = 2$ and $w_p(u_i) \leq w$, the number of linearly independent lines is $\frac{w(w+3)}{2}$. This does not allow to attack the cryptosystem since to have $\frac{w(w+3)}{2} \geq 12w + 1$, we would need $w \geq 22$ and if $w_p(u_i) = 2$, then the degree of the corresponding polynomials is $22 \times 12 + 1 = 265 > 197$.

Regarding that last set of examples, the Chor-Rivest attack would require to do an exhaustive research on approximately 251^{10} different possible generators of $\mathbb{F}_{p^{10}}$. With our approach, the cryptosystem can be attacked using the right generator g_{p^3} of \mathbb{F}_{p^3} . The research is clearly a lot faster.

5.2.1 Number of linearly independent polynomials

We can study now the number of linearly independent polynomials that the writing of $g_{p^r}^{uc_j}$ can generate for all u such that $w_p(u) \leq w$.

For example, for $r = 2$, it seems (experimentally) that the number of such linearly independent polynomials is always $\frac{w(w+3)}{2}$ and does not depend on p or h .

For $r = 3$, the number of such linearly independent polynomials is $\frac{(w+1)(w+2)(w+3)}{6} - 1$.

We could guess that for bigger r , this number is always $\Theta\left(\frac{w^r}{r!}\right)$.

6 Draft

We should prove that indeed at least $C \frac{w^r}{r!}$ linearly independent polynomials can be found in the coordinate of $g_{p^r}^{uc_j}$ with $w_p(u) \leq w$.

Then, we have that all we need for the attack to work is to choose r such that there exists w such that $w \cdot \frac{h}{r} + 1 \leq p - 2$ and

$$\begin{aligned}
C \frac{w^r}{r!} &\geq w \frac{h}{r} \\
\Leftrightarrow C w^{r-1} &\geq h \cdot (r-1)! \\
\Leftrightarrow C w^{r-1} &\geq h (r-1)^{r-1} \\
\Leftrightarrow w &\geq (r-1) \left(\frac{h}{C}\right)^{\frac{1}{r-1}} \geq C' r h^{\frac{1}{r-1}}
\end{aligned}$$

This yield

$$\begin{aligned} C'h \cdot h^{\frac{1}{r-1}} &\leq p-3 \\ \frac{\log h}{r-1} &\leq \log(p-3) - \log h + C'' \\ r-1 &\geq \frac{\log h}{\log(p-3) - \log h + C''} \end{aligned}$$

besides, h must be chosen such that

$$h \simeq \frac{p}{\log p}$$

so

$$r \geq \frac{\log p - \log \log p}{C'' + \log \log p} \geq C''' \frac{\log p}{\log \log p}$$

meaning that the research for a generator g_{p^r} is almost polynomial...

7 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):439–444, 1992.
- Vau01 [2] S. Vaudenay. Cryptanalysis of the chor–rivest cryptosystem. *Journal of Cryptology*, 14:87–100, 2001.