Constructing Verifiable Random Number in Finite Field

Jun Ye[1], Xiaofeng Chen[2]*, and Jianfeng Ma[2]
[1]School of Science, Sichuan University of Science and Engineering
Zigong, Sichuan, China
yejun@suse.edu.cn
[2]School of Telecommunication Engineering, Xidian University
Xi'an, Shaanxi, China
xfchen@xidian.edu.cn, jfma@mail.xidian.edu.cn

Abstract

In information security field underlying cryptography, random sequences, which are the base of system security, play a very important role. Random sequences with high security are often needed in cryptography field. From the view of security, real random sequences should be completely unpredictable and reliable. In many circumstances, a random number not only need to be random, but also need to be verifiable. So verifiable random number is much needed in cryptography. Using linear equations in finite field, a method for constructing verifiable random number is proposed. It enjoys advantages of high efficiency and no error. Then the security properties such as unpredictability and unmanipulability are analyzed, and an example is given to show the feasibility of the method. Finally, a way for fast generation and verification of VRN with large amounts of data is given.

Keywords: Verifiable Random Number, Unpredictability, Verification, Linear equations

1 Introduction

In development of cryptography research, cryptographic techniques plays an increasingly important role in this field. The application of the random number is one of the most important aspects of cryptographic techniques, such as, key management, protocol of cryptography, digital signatures, and authentication. As the rapid development of technology of information security, new requirements for random sequences are emerging. In many circumstances, data should not only be random or pseudo-random, but its randomness should be verified for all participants so that they can believe that random data is not controlled by anyone.

In order to achieve verification, verifiable random function (VRF) [8] is proposed firstly by Micali, Rabin and Vadhan in 1999 [2]. Then some of other schemes are proposed. However, there is relatively less research on verifiable random functions and its applications. Most of the schemes are based on the assumptions of RSA and BDH [6, 1] difficult problems, such as the schemes in [2, 11, 3]. These researches give prototype theory of VRF, and the security is reliable. But there are much complex modular exponentiation in these schemes, the operational efficiency is low. VRF is impractical in electronic commerce.

Random number is essential in electronic commerce, so verifiable random number (VRN) is more practical in e-business activities. In this paper we propose a generation way for VRN based on linear equations. Linear equations are widely used in the mathematics. But the solutions to linear equations are always inexact in real number field. To restrict the domain to a finite field, there is no error to the solutions. For the high efficiency operation of homogeneous linear equations, we use it to generate VRN.

Journal of Internet Services and Information Security (JISIS), volume: 3, number: 3/4, pp. 106-115

*Corresponding author: School of Telecommunications Enginnering, Xidian University, China, Tel: +86-29-88204749, Web: http://ste.xidian.edu.cn/cxf/index.html

1.1 Related Work

Random sequences has played an important role in cryptography. In a sense, the security of whole system is depend on the security of random numbers. So security random numbers are highly required in cryptography. For the security, the only true random sequence is the most reliable, because it is completely unpredictable, and any random sequence is impossible to speculate other numbers. In real world, the operations about real random sequences, such as generate, copy and manipulate, are difficult to achieve. Therefore, in practical applications pseudo-random sequences and unpredictable sequences are usually used.

In many circumstances, we need random numbers, and we also need it can be verified for all participants, so that participants can believe that random number is not manipulated. So verifiable random number comes out. The idea of VRN is derived from VRF. It has good applications in e-voting and e-lottery schemes. VRF is proposed firstly by Micali, Rabin and Vadhan in 1999. Goldreich-Levin transformation is used in this scheme. But the efficiency of this transformation is very low, and the output range is very small. In order to avoid the low efficiency of Goldreich-Levin transformation, a more efficient scheme based on Diffie-Hellman problem is proposed by Lysyanskaya [6]. Then a new scheme is proposed by Dodis and Yampolskiy [1] based on bilinear group. Security hash functions are used in this scheme, and the input space is unlimited expanded. In the following years, some significant work is given in [2, 1, 7, 5, 10]. Some schemes about verifiable random functions with a shorter key and evidence based on bilinear group is proposed by Dodis and Yampolskiy [2]. Then Dodis [1], Naor, Pinkas and Reingold [7] propose a distributed way to generate verifiable random functions. Liu, Chen, and Wu [5] give the security proof of a verifiable random functions. After that, by the research of verifiable random functions, Liu, Ye, Cao [10] give a fast way to generate VRN with large amounts of data over finite field, in 2010. Then in 2012, a scheme for the fast generation of VRN based on interpolating polynomial [9] is given by Ye [4].

1.2 Our Contributions

In the scheme of VRN, we need the data, which given by all the participants, play a role in the generation of VRN, and the participants can verify whether the random number is generated from these data. Most of the schemes for the generation of VRN are based on RSA and BDH problems. But much complex modular exponentiations are used in the scheme, this leads to the low efficiency of the schemes. A new way to generate VRN based on linear equations is proposed in this paper,which is simple, convenient and efficient, and can achieve fast generation and verification for a large number of data.

The organization of this paper is as follows. Some preliminaries are given in Section 2. The verifiable random number based on linear equations is given in Section 3, and an example is also given to show the feasibility of our scheme. The security analysis is given in Section 4. The fast generation and verification of VRN with large amounts of data is given in Section 5. Finally, conclusion will be made in Section 6.

2 Preliminaries

Some preliminaries are listed in this section, which are used in the following sections.

2.1 Uniqueness of Solution of Full Rank Linear Equation over Finite Field

Given a linear equation $AX = Y \mod p$, here A is a full rank $n \times n$ matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & & \cdots \\ a_{j1} & a_{j2} & \cdots & a_{jn} \\ \cdots & \cdots & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

and

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

is an unknown vector, and

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

is a known vector. There exists a unique X over the finite field, which is satisfied the equation $AX = Y \mod p$.

Proof. For A is a full rank $n \times n$ matrix, then

$$d = |A| = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1(i-1)} & a_{1i} & a_{1(i+1)} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2(i-1)} & a_{2i} & a_{2(i+1)} & \cdots & a_{2n} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{j1} & a_{j2} & \cdots & a_{j(i-1)} & a_{ji} & a_{j(i+1)} & \cdots & a_{jn} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{n(i-1)} & a_n & a_{n(i+1)} & \cdots & a_{nn} \end{vmatrix} \neq 0 \mod p.$$

Then we can get d_i,

$$d_i = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1(i-1)} & y_1 & a_{1(i+1)} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2(i-1)} & y_2 & a_{2(i+1)} & \cdots & a_{2n} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{j1} & a_{j2} & \cdots & a_{j(i-1)} & y_i & a_{j(i+1)} & \cdots & a_{jn} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{n(i-1)} & y_n & a_{n(i+1)} & \cdots & a_{nn} \end{vmatrix} \mod p.$$

By using Cramer Rule,

$$x_i = \frac{d_i}{d} = \frac{\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1(i-1)} & y_1 & a_{1(i+1)} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2(i-1)} & y_2 & a_{2(i+1)} & \cdots & a_{2n} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{j1} & a_{j2} & \cdots & a_{j(i-1)} & y_i & a_{j(i+1)} & \cdots & a_{jn} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{n(i-1)} & y_n & a_{n(i+1)} & \cdots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1(i-1)} & a_{1i} & a_{1(i+1)} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2(i-1)} & a_{2i} & a_{2(i+1)} & \cdots & a_{2n} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{j1} & a_{j2} & \cdots & a_{j(i-1)} & a_{ji} & a_{j(i+1)} & \cdots & a_{jn} \\ \cdots & & \cdots & & \cdots & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{n(i-1)} & a_n & a_{n(i+1)} & \cdots & a_{nn} \end{vmatrix}} \mod p.$$

From this we know x_i is existent and unique.

2.2 Hash Function

A hash function is a PPT algorithm that takes an arbitrary block of data as input and returns a fixed-size bit string, the hash value, such that any (accidental or intentional) change to the data will (with very high probability) change the hash value.

The secure hash function has four main properties:

1. It is easy to compute the hash value for any given message: $\forall x$, there is an efficient computation algorithm $h(\cdot)$ to compute $h(x)$;

2. It is infeasible to generate a message when given a hash result: $\forall y$, it is computationally infeasible to find x, such that $h(x) = y$;

3. It is infeasible to modify a message without changing the hash: given x_1, it is computationally infeasible to find x_2, such that $h(x_1) = h(x_2)$;

4. It is infeasible to find two different messages with the same hash: it is computational infeasible to find x_1 and x_2, such that $h(x_1) = h(x_2)$.

3 Verifiable Random Number Based on Linear Equations

In this section the generation of VRN is proposed and then an example is given. In this scheme, the members can confirm the final number is unpredictable and unmanipulated.

3.1 Generation

A random number r is needed as a secret key among the members $U_1, U_2, \ldots, U_n$ to encrypt the message. The good method is to allow everyone to participate in generating the random number r impartially and everyone can use his/her own secret key to verify whether r is randomly generated or r is manipulated. A method of constructing verifiable random number based on linear equations is proposed. The steps of generating verifiable number are as follows:

1. U_i selects a vector $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in}), \alpha_{ij} \in \mathbb{Z}_p, 1 \leq j \leq n$ randomly, and a random number $\alpha_{i(n+1)} \in \mathbb{Z}_p$, then sends it to Computing Center (CC).

2. CC verifies whether the n vectors are linearly independent or not. If the n vectors are not independent, CC asks some members to select another vectors, until the n vectors are independent.

3. CC uses the n elements $\alpha_{i(n+1)}$ to construct a new vector

$$y = \begin{pmatrix} \alpha_{1(n+1)} \\ \alpha_{2(n+1)} \\ \vdots \\ \alpha_{n(n+1)} \end{pmatrix}.$$

4. CC can construct a linear equations

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} a = y \mod p.$$

   And CC can get the solution vector $a = (a_1, a_2, \ldots, a_n)$ by computing

$$a = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}^{-1} y = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \mod p.$$

   In this step, CC can also use Gaussian elimination which leads to less computation.

5. $a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$ is called linear equations solution random. CC publishes r as VRN. Let $r = a_1 \parallel a_2 \parallel \cdots \parallel a_n$ ($\parallel$ denotes concatenation operator) be the random verifable munber. CC publishes the solution vector $a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$ for verification.

3.2 Example

For example there are 3 users, the finite field is $\mathbb{Z}_7$, and the 3 independent vectors and the 3 random numbers are

$$\{(1,3,2),5\}, \{(0,2,5),3\}, \{(6,2,3),2\}.$$

So we have the linear equations

$$\begin{pmatrix} 1 & 3 & 2 \\ 0 & 2 & 5 \\ 6 & 2 & 3 \end{pmatrix} X = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix} \mod 7.$$

We can get

$$X = \begin{pmatrix} 5 \\ 2 \\ 4 \end{pmatrix}.$$

So the verifiable random number is $r = 5 \parallel 2 \parallel 4 = 524$.

4 Security Analysis

The verifiability, unpredictability and unmanipulability are discussed as following.

4.1 Verifiability

If U_i suspects the authenticity of r, u_i can verify whether his/her vector is used in the generation of r. The steps of the VRN verification are as follows:

U_i verifies whether the equation

$$\alpha_i a = \begin{pmatrix} \alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \alpha_{i(n+1)} \mod p$$

is true or false. If it is true, α_i is used in the process of creating r.

4.2 Unpredictability

Theorem 4.1. *If one of the n vectors $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ the adversary does not get, the probability of the verifiable random number can be predicted is at most $1/p$.*

Proof. Without loss of generality, we assume that $\{\alpha_2, \alpha_3, \ldots, \alpha_n\}$ and the vector $\begin{pmatrix} \alpha_{2(n+1)} \\ \alpha_{3(n+1)} \\ \vdots \\ \alpha_{n(n+1)} \end{pmatrix}$ are fixed, only α_1 and $\alpha_{1(n+1)}$ the adversary do not get. So adversary can get p vectors about $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ from the linear equations

$$\begin{pmatrix} \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \cdots & \cdots & & \cdots \\ \alpha_{j1} & \alpha_{j2} & \cdots & \alpha_{jn} \\ \cdots & \cdots & & \cdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \alpha_{2(n+1)} \\ \vdots \\ \alpha_{j(n+1)} \\ \vdots \\ \alpha_{n(n+1)} \end{pmatrix} \mod p.$$

The verifiable number is one of the p vectors. So the probability of the verifiable random number can be predicted is $1/p$.

Similarly, if there are t vectors the adversary do not get, the probability of the verifiable random number can be predicted is $\frac{1}{p^t}$.

So the probability of the verifiable random number can be predicted is at most $1/p$.

4.3 Unmanipulability

Theorem 4.2. *If one of the members is not manipulated by adversary, that means one of the n vectors* $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ *the adversary do not get, the probability of the verifiable random number can be manipulated is no more than* $\frac{1}{p^2}$.

Proof. If adversary want to control the output $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$, he/she should control all inputs. If not, the probability of the verifiable random number can be manipulated is no more than $\frac{1}{p^2}$.

Without loss of generality, we assume that $\{\alpha_2, \alpha_3, \ldots, \alpha_n\}$ and the vector $\begin{pmatrix} \alpha_{2(n+1)} \\ \alpha_{3(n+1)} \\ \vdots \\ \alpha_{n(n+1)} \end{pmatrix}$ are controlled, only α_1 and $\alpha_{1(n+1)}$ can not be manipulated. So X is satisfied with these equations

$$\begin{pmatrix} \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \cdots & \cdots & & \cdots \\ \alpha_{j1} & \alpha_{j2} & \cdots & \alpha_{jn} \\ \cdots & \cdots & & \cdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \alpha_{2(n+1)} \\ \vdots \\ \alpha_{j(n+1)} \\ \vdots \\ \alpha_{n(n+1)} \end{pmatrix} \mod p.$$

But for the equation

$$(\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1n}) \begin{pmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{pmatrix} = \alpha_{1(n+1)}$$

X is fixed but there are p choices of $\alpha_{1(n+1)}$, and for every $\alpha_{1(n+1)}$ there are p^{n-1} vectors satisfies this equation. There are p^n choices of $(\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1n})$ in $\mathbb{Z}_p$. So if adversary wants to manipulate X, he/she should get the exact $\alpha_{1(n+1)}$ and $(\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1n})$, that means $\alpha_{1(n+1)}$ should be given from the corresponding p elements, and $(\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1n})$ should be given from the corresponding p^{n-1} vectors. The probability is

$$P = \frac{p^{n-1}}{p^n} \frac{1}{p} = \frac{1}{p^2}.$$

Similarly, if there are t vectors are not controlled by adversary, the probability of the verifiable random number can be manipulated is $\frac{1}{p^{2t}}$

So the probability of the verifiable random number can be manipulated is no more than $\frac{1}{p^2}$.

5 VRN Produced by Large Amounts of Data

If the number of participants n is huge, it will take a long time to solve the linear equations. To overcome this deficiency, a multi-matrix scheme for the generation of VRN by using linear equations is proposed. When n is exponential increased, the computing time increases linearly. Then the improved scheme is as follows.

5.1 Generation

CC can generate VRN as follows.

1. n participants are divided into m groups, the number of participants in every group is $n_i, 1 \leq i \leq m$ which is less than n' (here n' is smaller than the number of equations in linear equations which we can get the solutions rapidly, and $\sum_{i=1}^{i=m} n_i = n$). Then there are m sets of linear equations.

2. The kth participant in the group i, which is made up by n_i members, gives an n_i-dimensional vector $\beta_{ik}, 1 \leq k \leq n_i$ and a random number β_{ik,n_i+1} to CC.

3. CC verifies whether the n_i vectors are linearly independent or not. If the n_i vectors are not independent, CC asks some member to select another vector, until the n_i vectors are independent.

4. CC uses the n_i vectors and the n_i random numbers to construct a linear equations.

$$\begin{pmatrix} \beta_{i1} \\ \beta_{i2} \\ \vdots \\ \beta_{in_i} \end{pmatrix} a_i = \begin{pmatrix} \beta_{i1,n_i+1} \\ \beta_{i2,n_i+1} \\ \vdots \\ \beta_{in_i,n_i+1} \end{pmatrix} \mod p.$$

   And CC can get the solution vector $a_i = (a_{i1}, a_{i2}, \ldots, a_{in_i})$ by computing

$$a_i = \begin{pmatrix} \beta_{i1} \\ \beta_{i2} \\ \vdots \\ \beta_{in_i} \end{pmatrix}^{-1} \begin{pmatrix} \beta_{i1,n_i+1} \\ \beta_{i2,n_i+1} \\ \vdots \\ \beta_{in_i,n_i+1} \end{pmatrix} \mod p.$$

   And then CC can get the solution vector

$$a_i = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in_i} \end{pmatrix}.$$

5. $a_i = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in_i} \end{pmatrix}$ is called linear equations solution random. And CC computes $r_i = h(a_{i1} \parallel a_{i2} \parallel \cdots \parallel a_{in_i})$ as the ith random verifiable random number.

6. CC computes $r = h(r_1 \parallel r_2 \parallel \cdots \parallel r_m)$, (here $h(\cdot)$ is a secure hash function) and publishes r as VRN. CC also publishes the solution vector $a_i = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in_i} \end{pmatrix}$ $(1 \leq i \leq m)$ for verification.

5.2 Verification

Participants can verify the VRN by the following steps.

1. The kth participant in the group i verifies whether the following equation holds.

$$\beta_{ik} a_i = \beta_{ik} \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in_i} \end{pmatrix} = \beta_{ik,n_i+1}.$$

2. If the above equation holds, the participant verifies whether the following equation holds.

$$r = h(r_1 \parallel r_2 \parallel \cdots \parallel r_m).$$

If the two equations are holds, the participant can believe r is a VRN.

6 Conclusion

We propose a way to construct verifiable random number by using the linear equations over finite fields, and give the security analysis of our scheme, as well as prove the verifiability, unpredictability and unmanipulability of the VNR. We also give an example to show the feasibility of the method. Because of the limited accuracy, rounding error and truncation error of computer in real number field, the solutions to linear equations are always not accurate. But from the result we know there is no error for the solution to linear equations over finite field. And we give a way of fast generation and verification for VRN by large amounts of data. This verifiable random number has a wide range of applications in the field of cryptography.

Acknowledgements

We are grateful to the anonymous referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61272455 and 61100224), and China 111 Project (No. B08038).

References

[1] Y. Dodis. Efficient construction of (distributed) verifiable random functions. In *Proc. of the 6th International Workshop on Practice and Theory in Public Key Cryptography (PKC'03) , Miami, Florida, USA, LNCS*, volume 2567, pages 1–17. Springer-Verlag, January 2003.

[2] Y. Dodis and A. Yampolskiy. A Verifiable Random Function with Short Proofs and Keys. In *Proc. of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Les Diablerets, Switzerland, LNCS*, volume 3386, pages 416–431. Springer-Verlag, January 2005.

[3] S. Hohenberger and B. Waters. Weak Verifiable Random Functions. In *Proc. of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10), French Riviera, LNCS*, volume 6110, pages 656–672. Springer-Verlag, May-June 2010.

[4] Y.N. Liu et al. J. Ye, Y. Ding. Fast Construction and Security Analysis of Verifiable Random Number in Finite Field. *Computer Engineering & Science*, 34(5):35–39, October 2010.

[5] Y.N. Liu, S.X. Chen, and L.Wu. Security Proof for Verifiable Random Function. *Computer Engineering and Design*, 29(16):4172–4173, October 2008.

[6] A. Lysyanskaya. Unique Signatures and Verifiable Random Functions from the DH-DDH Separation. In *Proc. of the 22nd Annual International Cryptology Conference (CRYPTO'02), Santa Barbara, California, USA, LNCS*, volume 2442, pages 597–612. Springer-Verlag, August 2002.

[7] B. Pinkas M. Naor and O. Reingold. Distributed Pseudo-random Functions and KDC. In *Proc. of the 18th Annual International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'99), Prague, Czech Republic, LNCS*, volume 1592, pages 327–346. Springer-Verlag, May 1999.

[8] M. Rabin S. Micali and S. Vadhan. Verifiable Random Functions. *Proc. of the 40th IEEE Symposium on Foundations of Computer Science (FOCS'99), Atlanta, Georgia, USA*, pages 120–130, October 1999.

[9] J. Ye and Y.b. Su. Tow Construction Methods of Interpolation Polynomial in Finite Field. *Journal of Sichuan University of Science & Engineering (Natural Science Edition*, 23(5):521–523, October 2010.

[10] J.Y. Cao Y.N. Liu, J. Ye. Verifiable Random Number Based on Interpolating Polynomial over $\mathbb{F}_p$. *Journal of Sichuan University (Engineering Science Edittion*, 42(6):105–108, November 2010.

[11] G.N. Rothblum et al. Z. Brakerski, S. Goldwasser. Weak Verifiable Random Functions. In *Proc. of the 6th Theory of Cryptography Conference (TCC'09), San Francisco, California, USA, LNCS*, volume 5444, pages 558–576. Springer-Verlag, March 2009.

Author Biography

Jun Ye received the MS degree in Cryptography at the Guilin University of Electronic Technology in 2011. He is a lecturer at Sichuan University of Science and Technology. He is a doctoral candidate at the Xidian University. His research interests include cryptography and information security.

Xiaofeng Chen received his Ph.D. in cryptography from the Xidian University in 2003. He is currently a Professor at the School of Telecommunications Enginnering, Xidian University. His research interests include public key cryptography, financial cryptography, and cloud computing security.

Jianfeng Ma received the Ph.D. degree in Communication and Information System at the Xidian University in 1995. He is a professor at Xidian University. His research interests include cryptography and information security.