ElGamal Digital Signature Algorithm of Adding a Random Number

Xiaofei Li, Xuanjing Shen and Haipeng Chen
College of Computer Science and Technology, Jilin University, Changchun, China
Email: xiaofei09@mails.jlu.edu.cn, {xjshen, chenhp}@jlu.edu.cn

Abstract—As for the problem that ElGamal digital signature scheme's security is constantly being challenged and increasingly becomes increasingly serious, an improved ElGamal digital signature algorithm is proposed. As the original ElGamal algorithm has its own security disadvantages that only one random number is used, in order to improve its security, the scheme presented in this paper improved this demerit by adding a random number to the original one and increasing difficulty of deciphering key. The security of the improved signature scheme is the same with the ElGamal signature scheme which is based on the difficult computable nature of discrete logarithm over finite fields. Then issues about how to increase the complexity between the random number and the key by adding a random number is discussed. Last, we analyzed the improved signature scheme from the following two aspects: security complexity and time complexity. The analysis showed that the safety of the improved signature scheme was higher than that of the original one, and the improved one has a smaller time complexity.

Index Terms—ElGamal type; digital signature; random number; program improvement

I. INTRODUCTION

Digital signature is one of the main application for public key crytography. At present, the wider application of digital signature systems are RSA signature scheme[1] and ElGamal-type signature scheme[2], such as the Schnorr signature[3], DSA signature[4]. Although the hash function can avoid some attacks, however, if the system parameters are inappropriate, there is security risk[22]. Therefore, this is necessary to how to choose digital signature in study of application examples' design.

In 1985, ElGamal raised one of the two most important digital signature scheme, which is the ElGamal digital signature system[2]. It's security is based on the difficult computable nature of discrete logarithm over finite fields. The signature scheme with message recovery has many obvious advantages: a shorter signature for short message, and the shorter produced verification. If this scheme is applied to group signature, it has the difficult computable nature of discrete logarithm over finite fields and the advantage of message recovery in the digital signature[5-6]. As a widely used digital signature algorithm, ElGamal algorithm has the security on the discrete logarithm problem[16]. Generalized ElGamal-type signature scheme has stronger security than the original ElGamal algorithm through improving the original ElGamal-type signature scheme[7] and increasing the number of hidden parameters. This paper presents an improved ElGamal signature scheme based on generalized ElGamal digital signature scheme, and the safety of the improved signature scheme was higher than that of the original one. Not only does it broaden the scope of application, but can be used for the signature of the group users[8]. Group signature, as a branch of a subject oriented group cryptography, was proposed by Chaum and Van Heyst in 1991. This improved ElGamal algorithm makes ElGamal Digital Signature have a more extensive application in the fields of authentication and e-commerce system[9-13].

The rest of this paper is organized as follows: Sect. 2 gives a background on ELGamal algorithm and presents the research work related to digital signature in general, and mainly analyzes various attack means of ELGamal digital signature algorithm. The improved idea of ELGamal digital signature algorithm is given in Sect. 3. In Sect. 4, we introduce the specific improved ElGamal digital signature algorithm, including the specific process steps and implementation of improved algorithm. In Sect. 5 gives the performance analysis of the improved algorithm, and compares the performance of the original algorithm, it's security has significantly improved. Finally, the drawn conclusions and the planned future work are discussed in Sect. 6.

II. BACKGROUND AND RELATED WORK

This section provides an introduction to ElGamal digital signature algorithm on how it to be proposed and developed. Though research into this subject has been taking an important part, its security and efficiency are attacked gradually. This paper mainly analyzed the performance of ElGamal algorithm, and the security risks existing in the traditional ElGamal digital signature algorithm are discussed from four aspects.

A. Background

In 1985, ElGamal proposed a cryptosystem based on the discrete logarithm problem which can be used for both data encryption and digital signature [2]. Since it was put forward, the new cryptosystem aroused widespread interested in the password academic field.
because of its good cryptographic properties, which is the second most prominent signature scheme after the RSA signature scheme. Although the study of ElGamal's digital signature rose many years ago, it still has many problems with application in practice, especially the problems of its safety and efficiency. With the rapid development of e-commerce, the research on this project will be crucial.

The ElGamal digital signature scheme is non-deterministic like ElGamal public-key cryptosystem. Namely, the same clear message has different signatures due to the different parameters selected randomly, and each signature only corresponds to a random number [14], which has brought a great hidden danger to the security of ElGamal digital signature scheme [15-17].

In terms of the ElGamal digital signature scheme, the algorithm’s security depends on the security of the private key \( x \). Once the private key \( x \) is intercepted by the hacker, the entire digital signature algorithm is accessible to anyone, and no security exists at all [16]. Therefore, the primary target of the attacker is the private key \( x \). Therefore, the primary target of the attacker is the private key \( x \). The following are the common ways of attack, though in which the author compared the traditional ElGamal digital signature scheme with the ElGamal digital signature scheme after adding a random number, then analyzed and verified its security that is improved, it turns out that the private key \( x \) and random number \( k \) are unknown to the attacker.

B. Analysis of ElGamal Digital Signature Algorithm Security

In general, the following are the main ways of attack: a direct hack on the private key, arbitrary forged signature attack, substitution attack according to known signatures, the assault in homomorphism (using the same random number \( k \)), the assault in homomorphism (use the relevant random number \( k_1, k_2, k_3 \)).

1) A direct hack on the private key

Since the attacker's primary target is the private key \( x \), obviously the direct attack on the private key \( x \) is the most direct methods of attack.

The first thing we need to determine is: ElGamal digital signature scheme is based on the discrete logarithm problem. At present, there is no feasible solution to this problem. Then the attacker can only use the simplest method of exhaustion to calculate the value of the private key \( x \) compulsorily, and use the equation \( \alpha^k \equiv \beta \mod p \) to work out the solution set of private key \( x \) through a large number of operations. Then test each values of these \( x \) to determine which is the private key used by signers that need to get another signature. And calculate the value of the random number \( k \) using the equation (1) in signature equation. Then test it though the equation (2).

\[
\delta = (m - x \gamma) k^{-1} \mod (p - 1) \quad (1)
\]

\[
\gamma = \alpha^k \mod p \quad (2)
\]

In the course of the hack, the hackers apparently need to solve logarithm for a time, then test the solution sets obtained from solving logarithm, and test of each solution all need to go through an inverse element and exponentiation. In addition, the middle data which are produced in the operations will be greater, so there is the need for corresponding processing algorithms and adequate storage space. The complexity of time and space are very high.

2) Arbitrary forged signature attack

The goal that hackers attack algorithm is to find the private key \( x \), but their fundamental goal is to forge the signature using the private key \( x \), then sign the selected file of the attacker or sign the message. As the calculation of the equation and verification of the signature equation are public, then the attacker may forge signature directly through a public key and digital signature and verification equation without knowing the private key \( x \).

The attacker get the public key \( p, \alpha \) and \( \beta \), as well as the equation which need to be used for signing and verifying from the public sources. To forge a digital signature \((\gamma, \delta)\). An attacker can first select a value of \( \gamma \), and through \( \gamma \) and the known public key to calculate the value of \( \delta \) using signature equation or verification equation, thus a digital signature is forged completely.

If we use equation (1) in the signature equation to calculate the value of \( \delta \), then relate to the unknown private key \( x \) and the value of the random number \( k \) will be involved. The two calculations will encounter discrete logarithm problem. Therefore, only consider using the verification equation to calculate the value of \( \delta \) is the only way can be considered. Moreover, if the value of \( \delta \) is calculated through the verification equation, then it is clear that this forged digital signature can certainly through validation.

Calculate the value of \( \delta \) using the following validation equation:

\[
\beta^\gamma \gamma^\delta \equiv \alpha^n \mod p \quad (3)
\]

Where, \( p, \alpha \) and \( \beta \) are public key, \( m \) and \( \gamma \) are selected by hacker. Then what we can obtain through equation (3):

\[
\beta^\gamma \gamma^\delta \equiv \alpha^n \mod p \\quad \Rightarrow \gamma^\delta \equiv \alpha^n \beta^{-1} \mod p
\]

It is clear that the process of solving the index \( \delta \) must face the discrete logarithm problem.

3) Substitution attack according to known signatures

Set \((\gamma, \delta)\) is the signature of \( m \), if \( \gamma \) is reversible, then set \( k' = ek + n \), where \( e, n \) is any two integers, and satisfy:

\[
k' \in \mathbb{Z}_{p-1}
\]
\[ r' = r'^\times \alpha_x \mod p \]
\[ \delta' = \delta k \gamma^{-1} - \alpha_s (k(n)^{-1}) \mod (p - 1) \]

So, \((\gamma', \delta')\) is the signature of \(m'\), where:
\[ m' = \gamma'^{-1} \alpha_m \mod (p - 1) \]

Because \((\gamma, \delta)\) is a signature of \(m\), there is \(m = (\delta k + x\gamma) \mod (p - 1)\). If \(\gamma\) is reversible, there also is:
\[ x = \gamma'^{-1} (m - \delta k) \mod (p - 1) \]

From signature equation can obtain:
\[ m' = \delta' k^' + x\gamma'^{-1} \mod (p - 1) \]
\[ \delta', k', x, \gamma' \]

are substituted into the above equation, get:
\[ m' = \alpha'^{\times} \gamma'^{-1} m \mod (p - 1) \]

In this way, it also takes the attacker a long time to wait for the documents or information available after the digital signature has been forged. Take the long file into account, in order to reduce the computational times, the value of \(m\) obtained by using the open one-way function is used to sign. Thus the attacker can also consider attacking on the one-way function, so that make the selected file generate the summary corresponding with the value of \(m\). This relates to another algorithm of attack. At the same time attacked the two algorithms, the degree of difficulty is surely bigger than any method of attack by only attacking the ElGamal digital signature algorithm.

This method of attack allows an attacker to get a lot of legitimate digital signature, however, as the signatures associated with the value of \(m\), it is difficult to be practical.

4) The assault in homomorphism

There are two cases:

a) Using the same random number \(k\)

Among the parameters used in the ElGamal digital signature algorithm, the value of random number \(K\) is confidential in spite of the private key \(x\). \(k\) is randomly generated and are discarded after each signature, it seems there is no attack value, in fact, there are significant security vulnerabilities.

The random number \(k\) is only used once, because if the same value of \(k\) is used to sign on two or more signature files or messages, then the attacker can calculate the value of \(k\) by using the signature two times, and thus get indirectly the value of private key \(x\) indirectly.

Using the same random number \(k\), same \(\gamma\) in two signatures from (2) can be introduced. A hacker could use these two signatures which are \((\gamma_1, \delta_1)\) and \((\gamma_2, \delta_2)\), files or messages named m1 and m2. We can get group equation through (1):

\[ \begin{aligned}
\delta_1 k + x\gamma_1 &= m_1 \\
\delta_2 k + x\gamma_1 &= m_2
\end{aligned} \] (4)

According to this group equation, the hacker can easily work out the value of \(k\): \(k = (m_1 - m_0)(\delta_1 - \delta_0)^{-1}\). Then substitute the calculated the value of \(k\) into any equation in the group equations to calculate the value of private key \(x\), and then attacks will be successful.

In view of the shortcoming of random number \(k\), so ElGamal digital signature algorithm requires to use different value of \(k\) every time. As long as the signers strictly do this, the attacker can not use this simple and effective way to attack successfully.

For example, when the value of \(k\) with one signature is equal to the value of \(k\) with another signature, it is said that three random numbers \(k_1, k_2\) and \(k_3\) satisfy:

\[ k_3 = k_1 + k_2 \]

(5)

Obviously: \(\gamma_3 = \gamma_1 \gamma_2\). To calculate the value of the private key \(x\):

\[ x = (\delta_1\delta_2\delta_3 m_3 - \delta_1\delta_3 m_1 - \delta_2\delta_3 m_2) \]
\[ \left(\delta_1\delta_2\gamma_1\gamma_2 - \delta_1\delta_2\gamma_1 - \delta_1\delta_2\gamma_2\right)^{-1} \mod (p - 1) \] (6)

To improve security, \(h(m)\) is required instead of \(m\) in the signature equation, so, ElGamal digital signature algorithm signature equation is:

\[ \delta = (h(m) - x\gamma)k^{-1} \mod (p - 1) \]

According to the above analysis on ElGamal Digital Signature Algorithm, due to the discrete logarithm problem has yet no possible solution has been worked out yet, so the ElGamal type digital signature algorithm based on the question has high security keys. Before the discrete logarithm problem is effectively resolved, any direct attack on the keys, the computational needs are staggering.

According to the signature and the verify equation, the signer needs to complete a signature through a power operation and an inverse operation, and the verifier needs three power operations. Given that sign documents or information for the sign may be more, but the verifier is the different user; this conditions which the computational complexity of verifier is higher than the computational complexity of signers.

In terms of the current computing power, it is not difficult for the traditional ElGamal digital signature to compute. But insecurity caused by the random numbers has posed a great threat. An attacker can easily use the link among random numbers to access the value of private key \(k\) after the uncomplicated calculation. Analysis of the selected contact is the most simple two links; I believe there are other links making a threat to the security algorithm. Signer can of course find out the value of these random numbers to avoid using them to ensure the security of the algorithm. But this also brought another problem. Since the selection of random numbers is greatly reduced, and each random number can only be
used one time, and then discarded, which has seriously affected the life of the algorithm. Especially with network communication increased due to the development of the Internet, the need for digital signatures in more places, and the signer has to replace the value of the key x from time to time in order to ensure the safety of the signature algorithm, or even replace the public key $\alpha$. The replacement of the public key needed to be re-issued through public channels. Therefore, some customers maybe fail to access to the updated information and still used the no-consistent public key for authentication leading to the erroneous conclusion. As Internet users and the need to sign a document or a piece of information increased, Authenticator to the public key of the update delays will become more apparent because of this updating. We can also consider sending the public key and signature together. In fact, this is the public key as a part of the signature, which increases the length of the signature and the occupier of the amount of network resources in the transmission process. The error probability will increase resulting from increased data transmission. So, there will be more digital signature retransmission due to the error in transfer process.

Thus, in the ElGamal digital signature algorithm, the insecure random number constitutes a very large threat to its security. At the same time, measures taken to ensure the safety of the algorithm will lead to a series of problems, and the further extension of the algorithm has a significant restriction.

To improve the algorithm's security, we have improved the ElGamal digital signature algorithm by adding a random number and strengthening the link between the random number and the private key to make it more difficult to decipher.

III. IMPROVING METHODOLOGY

According to the analysis of two attacking methods on random number, it was found that the hacker can easily calculate the value of random numbers or the value of the key by calculation of a random number if a signer uses the insecure random numbers. This generally resulted from that it is easier to hack the random number than hack the key or too intimate relationship between the random number and the key. So, for the vulnerability of random number vulnerability in the ElGamal digital signature algorithm and too simple link between the random number and the key, in this paper, improved program was proposed to enhance the security of the algorithm, which can make the link between the random number and the key more complicated.

There are two signature equations of the ElGamal digital signature algorithm, shown as follow:

\[
g = a^x \mod p \tag{7}
\]

\[
\delta = (m - xg)k^{-1} \mod (p-1) \tag{8}
\]

Public key $\beta$ is calculated by the follow equation:

\[
\beta = a^\beta \mod p \tag{9}
\]

Compared with (7), (9) is the same as (7) in form, with $\beta$ generated through the private key x and as a public key. In (7) $g$ is generated through the random number $r$ and as a part of the signature. Then we can regard these three equations as a signature equation, but what is calculated as a public key instead of as a signature. If we take $\beta$ as part of the signature, $\gamma$ as a public key, corresponding to x as a random number, k as the private key, and the equation (8) is changed to follow:

\[
\delta = (m - k\beta)x^{-1} \mod (p-1) \tag{10}
\]

Verification equation is changed as follow:

\[
a^n = \gamma^\beta \beta^\delta \mod p \tag{11}
\]

This result of replacement is also an ElGamal digital signature algorithm. It can be seen that there is no essential difference between the random number k and the private key x. They are in different positions only because (8) is different. Then we can consider adding such a random number, and a corresponding increase in a form such as the type (9) of the equation to the signature equation, namely:

\[
l = a^l \mod p \tag{12}
\]

Accordingly, it is needed to make the appropriate changes to the signature equation and the verify equation. The signature equation in (8) is changed as follow:

\[
m = (xy + k\lambda + t\delta) \mod (p-1) \tag{13}
\]

The authentication equation is changed as follow:

\[
a^n = \beta^l \gamma^\delta \lambda^\delta \mod p \tag{14}
\]

In this way, the linkage between the random number and the private key x is established based on (13). As for (13), the values of $x$, $k$ and $t$ are to be identified. If the hackers obtained a random number value successfully and wanted to continue hacking, this is clearly more difficult than that for the original algorithm. Improved the above scheme further, signature equation of the new digital signature algorithm's signature equation can be obtained as follow:

\[
\beta = a^\beta \mod p \tag{15}
\]

\[
\gamma = a^\gamma \mod p \tag{16}
\]

\[
\lambda = a^\lambda \mod p \tag{17}
\]

\[
m = (xy + k\lambda + t\delta) \mod (p-1) \tag{18}
\]

Verification equation can be drawn as follow:

\[
a^n = \beta^l \gamma^\delta \lambda^\delta \mod p \tag{19}
\]

IV. IMPROVED DIGITAL SIGNATURE ALGORITHM

The difference between the improved algorithm and the original ElGamal digital signature algorithm is mainly reflected in adding a random number aiming to make the
original algorithm more complicated and more difficult to decipher. The specific algorithm is as follow:

Step 1: A large prime number \( p \) is produced by system, \( \alpha \) is a generator of \( \mathbb{Z}_p^* \), \( x(1 < x < \varphi(p)) \) is the signers' private key, the corresponding signature public key \( \beta \) can be calculated by \( \beta = \alpha^x \mod p \), and opened to the public key.

Step 2: Two different random numbers \( t \) and \( k \) are randomly selected by system. Where \( t, k \) and \( x \) must be co-prime and there is inverse. \( \gamma \) and \( \lambda \) are calculated by the \( \gamma = \alpha^k \mod p, \lambda = \alpha^x \mod p \) and retain \( \gamma \) and \( \lambda \).

Step 3: Signature explicitly \( m, \delta \) is calculated using the results of the first two steps as well as the extended Euclidean algorithm. \( \gamma \) and \( \lambda \) are obtained. The private key \( \lambda \) is calculated by \( \lambda = \gamma^t \mod p \) and retain \( \gamma \) and \( \lambda \).

Step 4: Discarded the random number \( k \) and \( t \), then the required public key \( p, \beta \) and \( \alpha \) are obtained. The private key \( x \) is calculated using the plaintext \( m \) and signature \( \gamma, \lambda, \delta \).

Step 5: \( (\gamma, \lambda, \delta) \) is sent to the corresponding customers by system. The customers use the following equation to verify the correctness of plaintext \( m \) digital signatures. If \( m \) is calculated by \( \gamma \) and \( \lambda \), the signature is correct. Otherwise, the signature is incorrect or transmission errors. The equation as follows:

\[
\alpha^m = \beta^\gamma \lambda^\delta \mod p
\]

Algorithm flow chart is shown in Fig. 1.

In the above-mentioned improved ElGamal digital signature algorithm, the same message \( m \) corresponded to different random number \( k, t \). And they can be all verified through the validation algorithm, which characterizes with uncertainty of signature and improved security.

When signers take \( \lambda \) as a signature, they need to finish one more computing power each time, which increases the amount of the signers' operations. When taken \( \lambda \) as a public key, the signer calculates a value of \( \lambda \), which could be used as many times as the value of \( \beta \). So the signer's computation amount is almost the same as that of the original algorithm. But for authenticator, each authentication has one more computing power, but increased with only about 0.5 time computation. As for the computer which can easily verify the ElGamal-type digital signature, verifying the signature of the improved algorithm does not consume more time.

\( \lambda \) can be used as public key if it is taken into account that the amount of the signers' operation is not increased. But whether this is safe, it is still needed to be determined through the analysis of safety. If the public key \( \lambda \) as a random number will lead to insecurity, \( \lambda \) should be taken as a part of the signature. This would increase the operation of the signer. But as long as the signature is not carried out in large amount, this computation can still be accepted.

V. ANALYSIS OF THE IMPROVED ALGORITHM

A. Security Analysis

ElGamal digital signature scheme is based on the difficult problems of seeking discrete logarithm in prime domain. At present, there are several types of hacks, which include a direct order to discrete logarithm algorithms, the use of special mathematical structure and parameters and so on [18]. But for the ElGamal digital signature scheme the hack comes from two aspects: on one hand, the key \( x \) is needed to be restored; on the other, the signature can be forged without restoring the key \( x \). The improved program is analyzed from the perspectives of following five kinds of hacks.

1) A direct hack on the private key

If the ElGamal digital signature scheme with a added random number is used to sign, at the initial stage the hacker can compute the solution set of the private key \( x \) as described above, and determine the private key, but the two random numbers \( k \) and \( t \) by encrypting equation cannot be calculated in the next step. The encrypting equation as follows:

\[
m = (xy + k\lambda + t\delta) \mod (p - 1)
\]

Similarly, the hacker will not be able to use (16) and (17) to verify the correctness of the private key \( x \) which has been known. Therefore, in improved ElGamal digital signature scheme, a hacker cannot attack the following formula:
Then the hacker will be able to think by attacking style (16) and (17) to derive the solution set of the two random numbers k and t, respectively. A signature is still needed to determine k, t values. And then it is substituted into (20) to verify.

In the course of the hack, the hackers apparently need to solve logarithm for three times, then test the solution sets obtained from solving logarithm, and test each solution need to go through an inverse element and exponentiation. This process is clearly more complex than the traditional method mentioned above. This also brings about more difficulties for the hackers.

In addition, as long as the value of private key x that selected by the signer is large enough, we can ensure that a hacker in the foreseeable period of time cannot not figure out the value of the private key x. While this has increased the amount of the signer’s operations, as for the public key β, it can be reused by calculating only one time. While the private key x in each signature involves only one multiplication operation, the increase in computational complexity by increasing x is not significant.

It can be said that such attacks on the security of ElGamal-type digital signatures are not a threat before the appearance of efficient algorithm for solving the discrete logarithm.

2) Arbitrary forged signature attack

The main attack will be focused on the digital signature verification equation in this way. However, algorithm verification equation is as follow after adding a random number:

\[ \alpha^* = \beta^r \beta^\lambda \alpha^x \mod p \]  \hspace{1cm} (21)

\[ \delta \] is calculated is as follow:

\[ \lambda^\delta = \alpha^\alpha \beta^\beta \gamma^\gamma \mod p \]  \hspace{1cm} (22)

Compared (21) with (22), the calculation of the latter is clearly more difficult because the latter one has one more inversion than the former.

And even hacked successfully, the value m of the selected files or messages m is the known quantity during the calculation process, so the calculation results can only be used for the signature of this document or message. The value of their attacks is clearly lowered.

3) Substitution attack according to known signatures

As for the improved algorithms, compared with many traditional digital signature algorithms, a hacker can still easily access signer's signature (γ, δ) of the message or document m, and then forged a number of legitimate digital signatures [19].

Since the improved algorithm introduces more parameters, the possibility that we want to adapt file m' can be reduced naturally. And forged methods need to be re-examined. Even if the method of forging signature can be found, the attack imposed on it is more complex than that on the ElGamal digital signature algorithm. Valid signature that is obtained in this way is still corresponding with value of m'. It is still a more daunting task to find the file or message to satisfy m'.

Methods of attack with an added signature data are more difficult to be constructed. Its computation is much more even if it succeeds. But the real difficulties for the hacker are still a valid signature that was forged at the same time, as well as produced the corresponding the value of m'. The attacker can successfully make the value of m' to the actual application. Hacker attacking in this way is only in the believers of his computing power consumption.

4) The assault in homomorphism

There are two cases:

a) Using the same random number k

Using the same random number k, same γ in two signatures from (04) can be introduced. A hacker could use these two signatures which are \( (γ_1, \delta_1) \) and \( (γ_2, \delta_2) \), files or messages named m1 and m2, and public key named γ. We can get group equation through (18).

\[
\begin{align*}
\{ m_1 &= x \gamma + k \lambda + t \delta_1 \\
02 &= x \gamma + k \lambda + t \delta_2 \\
\} 
\end{align*}
\]  \hspace{1cm} (23)

According to this group equation, the hacker can easily work out the value of t: \( t = (m_1 - m_2)(\delta_1 - \delta_2)^{-1} \).

Taken λ as a public key, then the corresponding t value is equivalent to a private key, that is, the hacker has obtained the value of a private key. But substituted the calculated t value into (23), the value of another private key cannot be immediately calculated. Get a binary linear equation: \( m_1 = x \gamma + k \lambda + \delta(m_2 - m_1)(\delta_1 - \delta_2)^{-1} \).

According to this equation we can get a solution set, and we can derive the value of x from verifying it by putting it into the signature equation one by one.

It has to go through a power calculation to verify the value of each x on the solution. But compared to the previous attack methods, it can be done at least in a limited period of time. That is, after an increase of a random number t and the corresponding x as the public key, when using the equal value of k to sign, the attacker can not obtain the value of the private key x through a simple calculation while the hacker can still attack successfully in a limited period of time. Therefore, it should be avoided to use the equal k-value to sign for the improved algorithm.

As for four numbers unknown, clearly, it is too difficult to solve with two equations. A hacker can get one ternary linear equation through this equation group and thus obtain a solution set, then test each solution set. Hackers have two ways to seek ternary linear equation. First, eliminating k-value, and second, eliminating x-value, then the solution set was about random numbers. Further, they need to solve the private key after the successful test. Eliminating k value, you can directly test the value of the obtained private key x.
In view of this shortcomings of the random number $k$, ElGamal-type digital signature algorithm will be required by using different values of $k$ for each time. As long as the signers do this strictly, a hacker cannot take advantage of this simple and effective method to attack successfully [20].

b) To use the relevant random numbers $k_1, k_2, k_3\ldots$

Suppose three random numbers $k_1, k_2, k_3$ as follows:

$$k_3 = k_1 + k_2$$  \hspace{1cm} (24)

From the signature equation (04) we obtain the relationship between $\lambda$:

$$\gamma_3 = \alpha^{k_3} = \alpha^{k_1 + k_2} = \gamma_1 \gamma_2$$  \hspace{1cm} (25)

The type (24) and type (25) are substituted into (18) obtained equations:

$$\begin{align*}
m_1 &= x\gamma_1 + k_1\lambda + t\delta_1 \\
m_2 &= x\gamma_2 + k_2\lambda + t\delta_2 \\
m_3 &= x\gamma_3 + (k_1 + k_2)\lambda + t\delta_3
\end{align*}$$  \hspace{1cm} (26)

A hacker can obtain a binary linear equation from equations, and then get a solution set about the private key $x$ and $t$. Then we verify the value of the solution set substituted into the signature equation one by one to find out the key pairs used by signer. This approach has one more power operation in the validation, but it can also be completed within a limited time. Moreover, for the same random number, using the relevant random number is more likely. A hacker would need to work out six unknowns from the three equations, and receive a quaternary linear equation. The complexity of its solution set and computing of the solution set of validation both have increased a lot.

Compared with the ElGamal digital signature algorithm, the improved algorithm is obviously much safer. Even in the state of insecurity, a hacker still needs to go through a great amount of computing to hack successfully. Signature can figure out the time limit, and change the private key and public key timely. As for the complex link among the random number $k$, their attacks will be more difficult. When the degree of difficulty is so large that the hacker cannot figure out the value of the private key $x$ within a limited period of time, algorithm can be considered safe.

Similar to ElGamal digital signature algorithm, the improved algorithm can still be used by a hacker when the random number meets the requirement of the more complex relationship. The process is more complicated, but it can still be completed within a limited time and calculations to obtain the private key. Signers also need to avoid using the same or related the value of a random number $k$ to sign.

Time complexity

Digital signature scheme based on ElGamal involves a large integer arithmetic problem in the process of signature and verification. It is especially large integer modular arithmetic, including modular power, modular inversion and so on. These operations greatly affect its speed, especially in the signature process for each user. In addition to select $x$ as his private key, also need to select secretly random number $k$, and for safety reasons, $k$ can not be reused, so users need to choose $k$ continuously. And the inverse model of $k$ needed to be calculated, while modular inversion of the large number is a very time-consuming thing. So in order to enhance the signature verification speed, on one hand, to improve the algorithm itself and improve processing speed; on the other, to improve digital signature scheme under the premise of not affecting the safety of signature.

The time complexity of ElGamal digital signature algorithm is $(\ln m^2 + \ln 2m + n^2)$ [21]. The improved scheme proposed in this paper compared with the traditional ElGamal algorithm increases one more a random number $t$ with a corresponding increase in calculation. All of the calculations about the random number $k$ in the traditional algorithms must be repeated in the calculation of $t$. The random number of modular inversion, modular computing power should be calculated more than once. For the signature equation of (19) and the verification equation of (20) calculation is more complicated. The amount of the calculation also increase compared to the former. The obvious efficiency compared to traditional algorithms has decreased, but its security has improved greatly, which is acceptable in real life.

For ElGamal algorithm's time complexity, algorithm of traditional computing a large integer modular power multiplication is a binary to index of $m$, it is said that $m$ is expressed as a binary form. The time complexity is $O(n \ln m)$; Then a series of iterative calculations are done, which are from the peak of m's binary, if meets the bit of 1, it will multiply the results of the previous step iteration with base number. In view of safety, ElGamal choose the value of $m$ that is relatively large, so speed of this method is slower, the time complexity is $O(\ln m n)$. So, the time complexity of calculating $a^n \mod n$ is $O(\ln m n) + O(\ln 2m) = O(\ln m n + \ln 2m)$.

Similarly, the function complexity of great common divisor of is $O(n)$. The time complexity of extended Euclidean algorithm function is $O(n)$, and the time complexity of generating the main signature function is about $O(\ln m n + \ln 2m)$.

In summary, time complexity of ElGamal algorithm is about: $(\ln m^2 n + \ln 2m + n^2)$.

After algorithm being improved, the time complexity has also been changed, but there is no fundamental change. There is one more process of seeking the equation in signing step, the equation is $\lambda = \alpha' \mod p$. When it seeks signatures, the calculation formula in comparison to traditional algorithms one more power of large numbers should be calculated. But these should not cause too much increases of the time complexity. At least, the order of magnitude of change is also not available, so time complexity of the improved algorithm is as follows: $O(\ln m n^2 + \ln 2m + n^2)$. 

© 2011 ACADEMY PUBLISHER
VI. CONCLUSION

This paper systematically analyzes the security of ElGamal digital signature algorithm under the four attacks scheme. Analysis show that there are two attacks against random number, thereby indirectly access the value of private key. And these are the two of the most likely schemes to succeed. In order to enhance the security of algorithm in random, we proposed two improved ideas: (1) Enhanced security of random numbers, making it difficult for the success of the random number of hacks; (2) Establish more complex link between the random number and the private key, so it is difficult for a hacker to use random number to attack the private key indirectly. Based on ideas that established the more complex link between the random number and the private key, we proposed to add the signature equation of the same form of an equation with the improvement of the program, thereby increasing a random number and a signature data.

The improved algorithm of security is enhanced, so that while a hacker needs greater computing capacity, the amount of signature and verification operations also will be increased. On the whole, the hacker's computational complexity is increased significantly. There are still some restrictions in the use of random numbers when the users sign. For example, we cannot make the two random numbers which were all the same and for the signature of two or more times. But generally speaking, the use of random numbers is restricted more lax than the restriction in the ElGamal digital signature algorithm. These are still to be continued to be improve. The analysis of the improved algorithm modeled on the ElGamal digital signature algorithm analysis is carried out by comparison. The new algorithm has its own characteristics. Especially, after the increase in random numbers, there may be an attacked method, which in the ElGamal-type digital signature algorithm it has not had. This is also the need for further study.

In this paper, security and efficiency analysis showed that the improved ElGamal algorithm in these two areas had significant increase or improvement, making the application wider in the production and life. But I still have to find a fact that, though in the vast majority of cases we can prevent the hacker from a variety of attacks, there are two real problem closely related to its vitality: (1) the current mathematical community for the discrete logarithm problem is still difficult for an effective solution[23]; (2) the signer must be very careful for the choice of random numbers. If there is a little vulnerability in these two areas, the ElGamal digital signature algorithm in relation should fade into history.

In conclusion, in terms of the improved algorithm in terms of security has been greatly improved, which makes its scope of application even greater. The impact due to the increase of computation in the signature and verification operations will be weakened with the enhancement of the computing power of the processor.

REFERENCES


Xiaofei Li, female, was born in Yantai County, Shandong Province, February, 1985. She received bachelor degree in 2009 from Changchun University and Technology. Now she is a master candidate in the college of computer science and technology, Jilin University. Her research interests are digital image processing and pattern recognition.

Xuanjing Shen, male, was born in Helong County, Jilin Province, December, 1958. He received bachelor degree in 1982, master degree in 1984, and PhD degree in 1990 all from Harbin Institute of Technology respectively. He is a professor and PhD supervisor currently in the college of computer science and technology, Jilin University. His research interests are multimedia technology, computer image processing, intelligent measurement system, optical-electronic hybrid system, and etc.

Haipeng Chen, male, was born in Cao County, Shandong, June, 1978. He received bachelor degree in 2003 and master degree in 2006 both from Jilin University. Now he is a lecturer and a Ph.D candidate in the college of computer science and technology, Jilin University. His research interests are computer network security, digital image processing and pattern recognition.

Dr. Chen is is membership of China Computer Federation (E20-00 15167M).