Empirical Study of Key Authentication Scheme in Public Key Cryptography

[a] P.Kumaraswamy, [b]C.V.Guru Rao, [c] V.Janaki
[a] Assistant Professor, Dept. of CSE, S.R.Engineeing College, Warangal, India
[b]Professor, Dept. of CSE, S.R.Engineeing College, Warangal, India
[c] Professor, Dept. of CSE, Vaagdevi college of Engineering, Warangal, India

Abstract

Public key cryptosystem plays major role in many online business applications. In public key cryptosystem, public key is needing not be protected for confidentiality, but the authenticity of public key is needed. Earlier, many key authentication schemes are developed based on discrete logarithms. Each scheme has its own drawbacks. We developed a secure key authentication scheme based on discrete logarithms to avoid the drawbacks of earlier schemes. In this paper, we illustrate the empirical study to show the experimental proof of our scheme.

KEYWORDS— Chinese remainder theorem, discrete logarithms, Public key, Private key, Certificate, Server.

I. Introduction

Cryptography is the study of mathematical techniques concerned with keeping Information protected from adversaries. Several cryptographic schemes have been developed so that data transmitted over the network is encrypted and cannot be anticipated by adversaries. Cryptography encompasses several areas of computer science and mathematics such as computational theory, information theory, number theory, complexity theory, algorithms, and probability. A cryptographer focuses on designing and analyzing cryptographic algorithms and protocols. Cryptography algorithms are broadly divided into two categories:[14] Symmetric key cryptography and Public key cryptography. In Symmetric key cryptography, two people should mutually agree on a cryptosystem and a secret key which is to be distributed in a secure manner with the help of trusted key distribution center (KDC). If one person wants to transmit messages to the other, the mutually recognized secret key is used to encrypt and decrypt messages. The serious problem in symmetric key cryptosystem is the key distribution, in a secure manner. This problem is the motivation to develop the public-key cryptosystem. In public-key cryptography, two people mutually agree on a cryptosystem and generate a pair of different keys, named as public key and private key. The public key of each user is publicly available and is accessible through public-key directory. If a sender wants to transmit message he/she uses the public key of the receiver accessed from the public-key directory to encrypt the messages. When the receiver receives the encrypted messages, receiver uses his/her private key to decrypt the messages. In public-key cryptosystem, there is a possibility that an intruder can modify the public key of the authorized user. It leads to public key authentication problem. But public-key cryptosystems are secure only if the authenticity of the public key is assured.

Many key authentication schemes such as ID based schemes [15], certificate-based schemes [16, 17], and self-certified public-key scheme [18] have been proposed to provide assurance on public key of authorized users. In ID based schemes, each public key is purely the user's identity itself. No public file is needed to store users' public

keys, thus there is no key authentication problem in such schemes. However, there is an authority, called as trusted center (TC), for calculating the corresponding private key of each user. Hence a TC, having the privilege to know each user's private key, can easily impersonates any user at any time. In certificate-based schemes, there is an additional authority called a key authentication center (KAC). The KAC stores all users' public keys and the related signatures after their identifications are verified. The signature, also called a certificate, of a public key is calculated by applying a one-way function to it. The one-way function is supposed to be known by every user. If a user wants to send a message to the receiver, the certificate is then recalculated by the user and compared to the one stored by the KAC to check whether the public key is forged or not. The schemes also have no key authentication problem, since the certificate stored in the system cannot be forged. Moreover, the KAC of the system does not know the private key of each user. However, the weakness in the schemes is that the KAC can still impersonate a user by generating false certificates. In self-certified schemes, the authority cannot compute the private keys of the users. Another advantage of such schemes is that, if there are two or more KACs, and hence two or more self-certified public keys for each user, then if there exists at least one honest KAC, the cheater or cheaters can be easily detected. Furthermore, it can be proven that the authority generates false certificates, if it does so. All the above schemes require at least one authority as a trusted center or third party for ratification. The authority has to be trusted, hence the honesty of the authority dominates the security of the system. Besides, it is a complicated task to make all authorities work together.

Some more key authentication schemes [8-13] were developed based on discrete logarithms and without using any certifying authorities. The certificate is generated by the user with the help of trusted server, not via a TC or a KAC. Each user is associated with three public information elements in server for authentication: the public key, the encrypted password, and a certificate. These schemes were also proved insecure as it was easy to forge the public key, and the server is solely responsible for generation of certificates. This leads to serious security flaw in case, when the server is compromised.

The purpose of key authentication schemes discussed above was to verify and prevent the forging the public key of a legal user. Therefore, key authentication forms a basis for security and survival of public key. It can be understood that key authentication is the challenging task in secret communications and data security. Thus, there is a need to design a new key authentication scheme with more efficiently.

Earlier schemes [8-13] were implemented using discrete logarithms alone and proved as insecure schemes. Discrete logarithms can produce maximum security in calculating public key and certificate. The use of discrete logarithm alone gives the vulnerabilities in the design of the key authentication schemes. Based on the above experiences an improved key authentication scheme is proposed using discrete logarithms and Chinese remainder theorem. The Chinese remainder theorem produces a unique value which is not easy to anticipate by an attacker. This new scheme considers the forging of public key and certificate produces a maximum level of complexity.

This proposal is made up of three phases: First is a setup phase, second is a registration phase and last is authentication phase. In the setup phase key server initializes public parameters like a large prime number, prime divisor and one-way function. Then in the registration phase, every user computes a set of values like public key, private key and some more parameters. With the help of these parameters, now user finds an unique value using Chinese remainder theorem. This unique value plays a

vital role in maintaining the security of public key and generation of certificate. After generation of certificate, user sends his/her certificate, public key and some other parameters are to be sent and verified by key server. Finally during Authentication phase, public key and certificate of user (receiver) is verified by the other user (sender) who wants to send information to the receiver. If the result of verification process is true then the public key and the certificate pair of the user is accepted otherwise it is rejected.

Our proposal of using the key server as a trusty authority for publishing and accessing resources. This avoids the generation of false certificate and forging of the public key of user even though the sever is compromised. Therefore, our new key authentication scheme is successful and found efficient for server based, banking, and military applications.

The rest of the paper is organized as follows. Section2 reveals a survey of related research works carried out on the topic under investigation. Section 3 describes the Chinese remainder theorem and discrete logarithms. Our new key authentication scheme is described in Section4. Numerical analysis of the scheme discussed in Section5.Section 6 concludes the research work followed by bibliography.

II. Related work

The key authentication schemes were examined and related work is presented in this section. Many key authentication schemes such as ID based schemes [15], certificate-based schemes [16, 17], and self-certified public-key scheme [18] have been developed with the help of trusted third party and proved as unsuccessful.

Some more key authentication schemes [8-13] were developed without any trusted third party and also proved as insecure schemes. Horng and Yang [13] proposed a key authentication scheme, HY-scheme, which uses a server as an authority. In their scheme, each user generates his/her certificate using the combination of password from the server and his/he private key. Later, Zhan et al.[12] proved that HY-scheme was prone to the password guessing attack. An improved scheme, ZLYH-scheme succeeded in preventing the guessing attack, but failed to achieve nonrepudiation.

A new key authentication scheme was proposed by Chi Lee, Shiang, Hwang and Hua Li [11], LHL-scheme, to achieve non-repudiation. It is based on the research work on key authentication schemes of HY-scheme and ZLYH-scheme. But the LHL-scheme has two security flaws, as suggested by A.Peinado [9], the primary being the recovery of user's private key from his/her certificate and other public values. The second significant drawback is certificate verification process is independent of certificate, for a given public key. Later, A.Peinado[9] improved LHL-scheme through the use of access control equation for public key verification without generating the certificate.

After few years Zhang and Kim [10] independently performed cryptanalysis of LHL-scheme and proposed a modified LHL-scheme. Their scheme is based on three security parameters and it is almost hard to calculate private key from certificate and other public parameters. Also it prevents guessing attack and achieves nonrepudiation along with thorough certificate verification. Finally, Zuhua Shao proposed a new key authentication scheme for cryptosystems based on discrete logarithms [8]. It is based on A.Peinado, Zhang and Kim and LHL research work. He proved that A.Peinado's modification is not secure as it suffers from guessing attack and also revealed that Zhang and Kim's scheme which is based on three security prerequisites is expensive in

an open environment. To overcome these drawbacks, Zuhua Shao proposed a new scheme in which a trusted key server is used to generate certificate for each legal user .

However, Zuhua Shao scheme's dependency on key server as a trusted third party is not secure when the server is compromised. Moreover, in this scheme, the availability of passwords in plaintext form of each user in server as (ID, PWD) pairs is not preferable as suggested by Purdy [19]. The Zuhua Shao scheme is similar to a public key authentication scheme based on certificate authority. But, the prime requisite of earlier schemes like HY , ZLYH and others is to secure users' passwords in the server and to make users generate their own certificates without any trusted third party.

In 2007, A provably secure short signatures scheme based on discrete logarithms was developed by Zuhua shao[7]. The new scheme offers a better security guarantee than existing discrete-logarithm-based signature schemes. The main advantage of this scheme over the DSA signature scheme is that it has a one-fourth reduction in both the signature length and the verification computation; the level of security is preserved. The new short signatures are needed to low-bandwidth communication, low-storage and low-computation environments, and particularly applicable to smart cards and wireless users. Although this model has been strongly accepted and the proofs in this model give a good sense of the actual security of the scheme, in general, security proofs in this model are not sound with respect to the standard model.

In 2009, Zuhua Shao proposed Security of self-certified signatures [6]. This paper describes Verifiers can validate both the signature on the message and the related certificate information simultaneously. A malicious certificate authority can generate a signature on a message and the related certificate information without the knowledge of long-term private key of the signer, or a malicious signer can collude with a malicious certificate authority to generate a temporary key pair without being authorized by other certificate authorities.

In 2013, Certificate-based verifiably encrypted RSA signatures was developed by Zuhua Shao and and Yipeng Gao [5]. This paper shows that the signer can unilaterally choose a certificate authority as his/her adjudicator in fair exchange.It does not work on non random oracle model.

In 2014, A Provably Secure Signature Scheme based on Factoring and Discrete Logarithms developed by Zuhua Shao and and Yipeng Gao [4]. In this paper he described that it gives more confidence to the users in digital signatures and The computation requirement and the storage requirement are slightly larger.

In 2015, Certificate-based Fair Exchange Protocol of Schnorr Signatures in Chosen-key Model was developed by Zuhua Shao and and Yipeng Gao [3]. It describes, each participant is allowed to choose his Schnorr key pair freely without showing his knowledge of the private key. In abnormal cases, two shortcomings would come into view compared with previous fair exchange protocols of signatures.

We come across that the forgery of public key is possible in Zuhua Shao scheme when the server is compromised [1]. To prevent the drawbacks of earlier schemes, we proposed an improved key authentication system based on Chinese remainder theorem and discrete logarithms [2]. In our proposal, each user generates his/her certificate without involvement of the server. This scheme achieved the three major requirements of security system confidentiality, authentication and nonrepudiation along with thorough certificate verification.

III. Overview of Chinese remainder theorem and discrete logarithms

In our key authentication scheme, we employ the combination of Chinese remainder theorem and discrete logarithms to develop secure key authentication scheme.

i. Chinese remainder theorem

The Chinese Reminder Theorem is an ancient but important calculation algorithm in modular arithmetic. The Chinese Remainder Theorem enables one to solve simultaneous equations with respect to different moduli in considerable generality.

Theorem: Let m_1…….. m_k be integers with gcd(m_i, m_j) = 1 whenever i ≠ j. Let m be the product m = m_1.m2…… m_k. Let a1,,….. a_k be integers. Consider the system of congruences:

$$x = a1 \pmod{m1}$$
$$x = a2 \pmod{m2}$$
..
..
$$x = a_k \pmod{m_k}.$$

Then there exists exactly one x ∈ $\mathbf{Z}_m$ satisfying this system.

Example:

Use the Chinese Remainder Theorem to find all solutions in $\mathbf{Z}_{60}$ such that

x ≡3 mod 4

x ≡2 mod 3

x ≡4 mod 5

We solve this in steps.

Step 0: Establish the basic notation. In this problem we have k = 3, a1 = 3, a2 = 2, a3 = 4, m1 = 4,m2 = 3,m3 = 5, and m = 4. 3.5 = 60.

Step 1: Implement step (1). z_1 = m/m1 = 60/4 = 3 .5 = 15, z_2 = 20, and z_3 = 12.

Step 2: Implement step (2). We solve $z_i y_i \equiv 1 \bmod m_i$, i = 1,2,3. In this problem, we need to solve

$15y_1 \equiv 1 \bmod 4$

$20y_2 \equiv 1 \bmod 3$

$12y_3 \equiv 1 \bmod 5$

The y_i can be computed using the tally table version of the generalized Euclidean algorithm. For example, in the first equation for y_1, the tally method automatically solves $15y_1 + 4t = 1$ for y_1 and t, and we find that y1 = 3. Continuing, we find that y_1 = 3, y_2 = 2, and y_3 = 3.

Step 3: Implement step (3). $x \equiv a_1y_1z_1 + a_2y_2z_2 + a_3y_3z_3 \pmod{60}$. Substituting, we obtain 3 .3 . 15 + 2 . 2 . 20 + 4 . 3 . 12 = 359 which reduces to $x \equiv 59 \pmod{60}$.

ii. Discrete logarithm

The discrete logarithm problem presents itself as a simple mathematical problem but there is a computational presumption that it is difficult. It is important because of its wide applications in the field of cryptography. Nevertheless, its study always bears great academic significance. Indeed, it has become the subject of interest among cryptographers and mathematicians in recent times because of its computational

difficulty. Cryptosystems are considered secure under certain computational assumptions. For instance, the RSA scheme of Rivest, Shamir, and Adleman rests its security on the difficulty of the Factoring Problem. Many others, such as ElGamal, are based on the assumption that the Discrete Logarithm Problem is difficult to compute for certain groups.

Definition: The Discrete logarithm problem states: "Given a multiplicative group G and elements g, h∈G, find an integer n, if it exists, such that "$g^n = h$ ". This number n is the discrete logarithm of h to the base g, written more concisely as $n = \log_g (h)$.

In cryptographic applications, the existence of such an integer n is naturally presumed. Consequently, the problem is reduced to finding the number n. The word discrete "is used to distinguish those situations involving finite groups, like the ones being dealt herein, from the classical case. In 1976, Whitfield Diffie and Martin Hellman published a paper in which they proposed the discrete logarithm problem as a good source of a one-way function. That marked the inception of the discrete logarithm problem in cryptography. For the purpose of this study, we may think of a "one-way" function as a function $f : X \rightarrow Y$ for which given $x \rightarrow X$, it is easy to compute f(x), however, given $y \rightarrow Y$, it is difficult to compute a value $x \rightarrow X$ such that $f(x) = y$, at least for most values of y. In other words, from the standpoint of realistic computability, the function f is not invertible, without further information, and it is for this reason that such function is otherwise known as a trapdoor function.

Example: Consider the equation $3^k \equiv 13 \pmod{17}$ for *k*. From the example above, one solution is $k = 4$, but it is not the only solution. Since $3^{16} \equiv 1 \pmod{17}$—as follows from Fermat's little theorem—it also follows that if *n* is an integer then $3^{4+16n} \equiv 3^4 \times (3^{16})^n \equiv 13 \times 1^n \equiv 13 \pmod{17}$. Hence the equation has infinitely many solutions of the form $4 + 16n$. Moreover, since 16 is the smallest positive integer *m* satisfying $3^m \equiv 1 \pmod{17}$, i.e. 16 is the order of 3 in $(\mathbf{Z}_{17})^\times$, these are the only solutions. Equivalently, the set of all possible solutions can be expressed by the constraint that $k \equiv 4 \pmod{16}$.

IV. Key authentication scheme

This section describes our proposal on an improved key authentication scheme [2]. Every user is allotted with a unique identity called user ID and a password (PWD) to login in to the system. In general the system keeps a password table in a key server for all the authentic users and uses the key server as an authority. The password table stores each user's hashing password, f(PWD), where PWD is the password of the user and f() is a one-way function. Hence, the server cannot derive and know the PWD of the user because one-way function cannot inverse [12, 13].This proposal considers three phases. Setup phase, registration phase and authentication phase. These three phases are described as below:

A. Setup phase

In this phase, the key server initializes the following public parameters.

1. p, where p is a largest prime number
2. q, where q is a prime divisor of p-1
3. g , where g is a generator of order q in the finite field of Galois Field(p).
4. f( ) is a one way function defined as

$$f(x)=g^x \bmod p \qquad (1)$$

Therefore, upon initialization the public parameters such as p, q, g and f( ) are being setup. The next subsection considers these parameters for registration.

B. Registration phase

In this registration phase, the initialized parameters are considered for registration of every user. For registration, every user is to compute his/her public key and certificate.

First, to compute the public key (Pub) every user selects his/her private key (Prv) and utilize function f() on private key as follows.

Pub = f(Prv) (2)

But, from equation (1) stated above, equation (2) can be rewritten as

$Prv = g^{Prv} \bmod p$. (3)

Where g, p are parameters already initialized in setup phase. Therefore Pub = f(Prv) can be computed using equation(3).

Next, the certificate of the user is computed by the following three steps listed here under:

Step1: The set of new parameters such as r, S, R, T, X, v1, v2, k1, k2, α, and β are computed by the user using the existing parameters such as PWD, Prv, Pub, and f().The set of these new parameters are being explained as below:

r is a random number.
S, R, T, X are variables to hold computed values.
v1, v2 are two random numbers.
k1, k2 are two constants.
α , β are two one-way functions.

The computation of these parameters is shown as below:

i) Choose a random number r in Z_q^*, where Z_q^* contains all integers from 1 to q-1 and q is initialized to a value in setup phase

ii) Compute S , R and T values using the existing parameters PWD, r and f( ) as follows:

a) S = f (PWD + r) (4)

b) R= f(r) (5)

c) T= f (PWD * r) (6)

iii) Select two random numbers v1 and v2 , which are co-prime in the field Z_q^*

iv) Apply the Chinese remainder theorem to compute a value X

$X \equiv Prv \pmod{v1}$ (7)

$X \equiv (PWD * r) \pmod{v2}$ (8)

The value X is unique in the modulo of v1* v2.

v) Once again select k1 and k2 values to satisfy the following two equations

X = Prv + k1 X v1 (9)

X = (PWD X r) + k2 X v2 (10)

vi) Finally compute two more parameters α and β as follows

$\alpha = g^{v1} \bmod p$ (11)

$\beta = g^{v2} \bmod p$ (12)

The step1 computes all the values of r, S, R, T, X, α and β. These are considered in step2.

Step2: In this step, the user sends to the key server the parameters that were calculated in.

The key server verifies them with the following two equations (13) and (14) . These two equations are generated using the existing equations from (4-6), and (9-12).

$S = (PWD) * R \bmod p$ (13)

$Pub * \alpha^{k1} = T * \beta^{k2} \bmod p$ (14)

If the values of the received parameters satisfy the above two equations, then the key server accepts them. The parameters such as T, α, β, k1, k2 are stored in public password table The remaining parameters such as S, R, Pub are stored in secret password table. In case, if the verification process by the key server fails, the parameters of the user are not stored. The user is declared as unauthentic.

This completes the verification of public key. Now, the user computes his/her own certificate. This process of generating certificate is described in the next step.

Step3: If the verification is successful in step2, then user computes his/her certificate (C) with the computed parameters such as PWD, r , X, Pub, q, α and β as follows

$$C = [\alpha * (PWD * r) + \beta * X * Pub)] \bmod q \qquad (15)$$

Now, each user registers his/her computed public key and certificate in public key directory of key server as a pair of (C, Pub). First all users, is need to complete their registration successfully. If any two users want to communicate, they need to authenticate using C and Pub. This authentication process is described in the next subsection.

C. **Authentication phase**

In this phase, if a user (sender) wants to communicate with the other user (receiver) then they needs to authenticate each other. First, the sender obtains the pair (C, Pub) of the receiver from public key directory and also accesses all other parameters corresponding to the user from public password table stored in the key server and verifies using the equations (16) and (14) . The equation (16) is generated using the existing equations (4-6), (9-12) and (15).

$$f(C) = [\ T^{\alpha}\ ] * [T * \beta^{k2}]^{\ \beta * Pub} \bmod p \qquad (16)$$

If the above two equations are satisfied, the sender accepts the public key of the receiver otherwise the sender rejects it.

V. Implementation Results

This section demonstrates the Implementation results of our proposed key authentication scheme. We developed the implementation of our scheme with **GNU Multiple Precision Arithmetic Library** (**GMP**) .GMP allows us to use integers whose sizes can grow dynamically to the required precision. So by putting many words together, it can support 128, 256 ,512, 1024 or 2048 bits. The library will dynamically allocate memory for accommodating extra bits of precision as and when needed. Here we are presenting the implementation results our scheme for input size data 512 bits.

Here, two cases are described to represent user key authentication case1 represents legal user key authentication and case2 represents illegal user.

Case1: Legal user Key authentication

The public key Pub and certificate C is computed by user A(receiver) and registered as a pair in public key directory.Then user B (sender) get the information as follows:

A. **Setup phase:** User A (receiver) has to initialize p,q,g . So p,q,g values are

p: 13046590947410331827

q: 6523295473705165913

g: 5940081968261655414

B. Registration Phase : This phase calculates values:P,S,R,T, α, β,k1,k2 and X .

Pub: 11273483910247387682

S: 6136982508694857965

R: 11441037113809626421

T: 10389870701216010533

X: 23727192487611219503387625938682810869

K1: 5933840194776709664

K2:588260733325831863935726587002882560742924893519075978425218681163503531545610811994822954429682322391091202462186289243400798091269486500687728773540708218545643082430666820104900300587518365800747534638782115423894478893679382486229075346876214003224852254312236974057965589664072433470065310092128929620 3

α: 6702906520560225638

β: 8447959612466220742.

Now, except X all values are send to server and server has to verify and register all the values as follows:

Pub: 11273483910247387682

S: 6136982508694857965

R: 11441037113809626421

T: 10389870701216010533

K1: 5933840194776709664

K2:588260733325831863935726587002882560742924893519075978425218681163503531545610811994822954429682322391091202462186289243400798091269486500687728773540708218545643082430666820104900300587518365800747534638782115423894478893679382486229075346876214003224852254312236974057965589664072433470065310092128929620 3

α: 6702906520560225638

β: 8447959612466220742

g: 5940081968261655414

p: 13046590947410331827

Server registered

Sucess!!!

C. Authentication Phase:

Now User B (Sender) has to verify the values of receiver which are stored in server as follows:

T: 10389870701216010533

K1: 5933840194776709664

K2:588260733325831863935726587002882560742924893519075978425218681163503531545610811994822954429682322391091202462186289243400798091269486500687728773540708218545643082430666820104900300587518365800747534638782115423894478893679382486229075346876214003224852254312236974057965589664072433470065310092128929620 3

α: 6702906520560225638

β: 8447959612466220742

C:88624589211025437

Pub: 11273483910247387682

Client 2 verification Done!! **Legal user**

Here, sender accepts the public key Pub of the receiver.

Case2: Illegal user key authentication

If the public key Pub of user A (receiver) is modified.Then user B (sender) get the information as follows:

C. Authentication Phase: Now User B (sender) get the information as follows:

T: 10389870701216010533

K1: 5933840194776709664

K2:588260733325831863935726587002882560742924893519075978425218681163503531545610811994822954429682322391091202462186289243400798009126948650068772877354070821854564308243066682010490030058751836580074753463878211542389447889367938248622907534687621400322485225431223697405796558966407243347006531009212892962035

α: 6702906520560225638

β: 8447959612466220742

C: 88624589211025437

Pub: 112734839102473876

Client 2 verification Done!! **Illegal user**

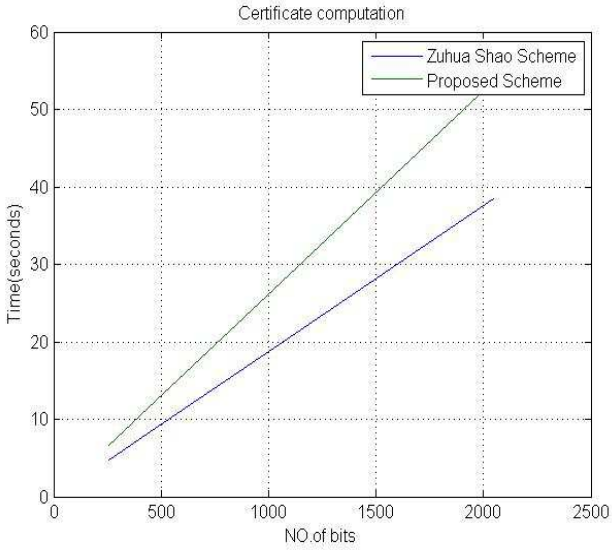
Here, sender rejects the public key Pub of the receiver

The following table shows the computation time **(Seconds)** of certificate calculation in Zuhua Shao scheme and proposed scheme

Table1: Computation time (Seconds) of certificate in Zuhua Shao scheme and proposed scheme

Size of input	Zuhua Shao Scheme	Proposed Scheme
256 bits	6.702	4.808
512 bits	13.404	9.616
1024 bits	26.808	19.232
2048 bits	53.616	38.464

The following graph shows the graphical representation of table1.



Graph1: Computation time (Seconds) of certificate calculation in Zuhua Shao scheme and proposed scheme

According tothe table1 and graph1, our scheme reduces the computation time of certificate compare to Zuhua Shao scheme.

VI. Conclusion

This paper shows the numerical proof of our key authentication scheme based on Chinese remainder theorem and discrete logarithms. One significant feature of this scheme is that the certificate is generated without explicit involvement of Prv and it uses a unique value X, which is computed using Chinese remainder theorem. This new concept of our scheme increases the complexity to a hacker to obtain the exact private key from the certificate. One more feature of this scheme is that the key server is used as a trusty authority only for publishing and accessing resources but not as a certificate generator. Besides, this scheme allows user to generate his/her certificate without the help of any trusted third party. Therefore, our proposed scheme demonstrates a secure and simple key authentication scheme.

References

1. P.Kumaraswamy, C.V.Guru Rao, V.Janaki, K.V.T.K.N.Prashanth. “Cryptanalysis of Zuhua Shao Key Authentication Scheme”, Procedia Computer Science, Elsevier B.V., pp. 95-99, 2016.
2. P.Kumaraswamy, C.V.Guru Rao, V.Janaki, K.V.T.K.N.Prashanth, “Key Authentication Scheme-based on Discrete Logarithms and Chinese Remainder Theorem”, Defence Science Journal Vol. 66, No. 6, November 2016, pp. 590-593, DOI : 10.14429/dsj.66.9649
3. Zuhua Shao and and Yipeng Gao, “Certificate-based Fair Exchange Protocol of Schnorr Signatures in Chosen-key Model”, Fundamenta Informaticae, 141 (2015) 95–114 95
4. Zuhua Shao and and Yipeng Gao, “A Provably Secure Signature Scheme based on Factoring and Discrete Logarithms”, Appl. Math. Inf. Sci. 8, No. 4, 1-6 (2014).
5. Zuhua Shao and and Yipeng Gao,”Certificate-based verifiably encrypted RSA signatures”, Transactions on Emerging Telecommunications Technologies,Trans. Emerging Tel. Tech. 2015; 26:276–289
6. Zuhua Shao, “Security of self-certified signatures”, Information Processing Letters 109 (2009) 1147–1150.
7. Zuhua Shao, “A provably secure short signature scheme based on discrete logarithms”, Information Sciences ,177 (2007), 5432–5440
8. Zuhua Shao, “A new key authentication for cryptosystems based on discrete logarithms”, Applied Mathematics and Computation, 2004
9. A. Peinado, “Cryptanalysis of LHL- key authentication scheme”, Applied Mathematics and Computation, 152:721–724, 2004.
10. F. Zhang, K. Kim. Cryptanalysis of Lee–Hwang–Li_s, “key authentication scheme”, Applied Mathematics and Computation, in press.
11. Cheng-Chi Lee, Min-Shiang Hwang, Li-Hua-Li, “A new key authentication scheme based on discrete logarithms”, Applied Mathematics and Computation, 139: 343-349, 2003
12. B. Zhan, Z. Li, Y. Yang, Z. Hu,” On the security of HY-key authentication scheme”, Computer Communication, 22:739–741, 1999.

13. G. Horng, C.S. Yang, “key authentication scheme for cryptosystem is based on discrete logarithms”, Computer Communication, 19:848–850,1996.
14. Bruce Schneier, “Applied Cryptography”, second ed., John Wiley & Sons, New York, 1996.
15. A.Shamir, “Identity based cryptosystems and signature schemes”, in: Advances in Cryptology,CRYPTO_84, Lecture Notes in Computer Science, 47–53, 1984
16. M. Kohnfelder, “A method for certification”, in: Tech. Rep. (MIT Laboratory for Computer Science), MIT Press, Cambridge, MA, 1978.
17. G. Simmons. Contemporary Cryptology: The Science of Information Integrity, IEEE Press,1992.
18. M.Girault, “Self-certified public keys, in: Advances in Cryptology”, EUROCRYPT_91, Lecture Notes in Computer Science, 491–497, 1991.
19. Purdy, G.B.,”A high security log-in procedure”. Communications of the ACM, 1974, 17, 442– 445