Digital Signature Schemes from Two Hard Problems

Binh V. Do, Minh H. Nguyen and Nikolay A. Moldovyan

Abstract In this paper, we propose two new signature schemes and a novel short signature scheme from two hard problems. The proposed schemes have two prominent advantages. Firstly, they are developed from some signature schemes where the security and efficiency have been proven. Therefore, they inherit these properties from the previous schemes. Secondly, the security of the proposed schemes is based on two hard problems. Therefore, they are still safe even when cryptanalysis has an effective algorithm to solve one of these problems, but not both. Moreover, we also propose a method for reducing signatures and this is the first attempt to reduce signatures based on two hard problems. Therefore, our proposed schemes are suitable for the applications requiring long-term security in resource limited systems.

Keywords Cryptographic protocol · Digital signature · Factorization problem · Discrete logarithm problem · Short signature scheme

1 Introduction

One of the vital objectives of a information security systems is providing authentication of the electronic documents and messages. Usually this problem is solved with digital signature schemes (DSSes) [1]. There were many proposals for

B. V. Do (✉)
Military Information Technology Institute, Hanoi, Vietnam
e-mail: binhdv@gmail.com

M. H. Nguyen
Le Qui Don Technical University, Hanoi, Vietnam
e-mail: hieuminhmta@ymail.com

N. A. Moldovyan
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, 14 Liniya, 39, St., Petersburg, Russia199178,

J. J. (Jong Hyuk) Park et al. (eds.), *Multimedia and Ubiquitous Engineering*,
Lecture Notes in Electrical Engineering 240, DOI: 10.1007/978-94-007-6738-6_100,

signature schemes published based on a single hard problem such as factoring (FAC), discrete logarithm (DL) or elliptic curve discrete logarithm (ECDL) problems [1, 2]. However, these schemes only guarantee short-term security. In order to enhance the security of signature schemes, it is desirable that the signature schemes are developed based on multiple hard problems. This makes it much harder to attack these schemes sine it requires solving multiple problems simultaneously. Some schemes based on two problems, FAC and DL, have been published [3–5]. However, designing these schemes is not easy. Moreover, most of them have been proven that they are not secure [6–8]. Therefore, it is necessary to develop new safe signature schemes based on two hard problems.

In bandwidth and resource limited systems, it is important that the signature schemes have a short signature length. So far, the problem of signature reducing is only investigated for the schemes with single hard problem [9, 10]. We can easily implement a combination of two or more hard mathematical problems in a unified DSS. Breaking such schemes requires simultaneously solving all hard problems. Such implementations require increasing signature length, because the signatures must be present elements belonging to different mathematical problems. It is therefore of interest to develop DSSes, that provide an acceptable signature length. The rest of this paper is organized as follows. In Sect. 2, describes the DSSes based on two hard problems (FAC and DL). Section 3, presents the design of two new DSSes, which requires the simultaneous breaking of FAC and DL problems. Section 4 proposes a novel and efficient short signature scheme. Section 5, describes the security analysis of our schemes. Section 6, describes the performance analysis of our schemes. In the last section, the conclusion of our research is presented.

2 Signature Schemes Based on Factoring and Discrete Logarithms

Previously, DSSes were proposed based on the difficulty in solving the factorization and discrete logarithm problems. For example, the scheme in [11] used a prime modulo p with a special structure $p = 2n + 1$, where $n = q'q$, q' and q are large prime numbers with at least 512 bits. We use the following notations to describe these signature schemes. H is a hash value computed from the signed document M. F is a one-way function, for which can be used to calculate the value of $H = F_H(M)$. α is a primitive element in Z_p^* with order q satisfying $\alpha^q \equiv 1 \bmod p$. The value of λ is a bit length of q, where q is a prime divisor value of n.

The public key is a triple of (p, α, λ). The private key is q.

Signature generation procedure:

(1) Compute $r = F_H(\alpha^k \bmod p)$, where k is a secret random number, $1 < k \leq q - 1$.
(2) The equation generating the parameter S is given by the following equation: $S = k(Hr)^{-1} \bmod q$.

The signature is a pair of values (r, S), in which the length of the second value is equal to $|S| \leq \lambda$;

When using 1024-bit prime p and a compression function F whose output is a t-bit length and assuming $t = 160$ bits, the length of the digital signature is $|F| + |q| \approx 160 + 512 \approx 672$ bits.

Signature verification procedure:

The verification equation is as follow: $r = F_H(\alpha^{HSr} \bmod p)$.

An important part of the verification procedure is to verify the authenticity of a digital signature with the condition $|S| \leq \lambda$, because signature (r, S') with second element of which has the size $|S'| \approx 1023$ bits (if $|p| \approx 1024$ bits) can be easily generated without knowing of the secret parameter q. Such signature (r, S') will satisfy the verification equation. However the signatures (r, S') do not satisfy the condition $|S'| \leq \lambda$. Computing the forged signature (r, S') satisfying both the verification equation and the condition $|S'| \leq \lambda$ without knowing the private key q is not easier than factoring the number $n = (p - 1)/2$ [11]. Security of the considered DSS is based on the difficulty of solving any of the following two problems, factorization and discrete logarithm. Indeed, it is easy to show that solving the factorization problem or solving the discrete logarithm problem allows one to compute the private key and to forge the signature.

In the Schnorr signature in [1], we can use a prime module with the structure of $p = 2n + 1$. This leads to the DSS with public key in the form of four values (p, α, λ, y), where the first three parameters are defined as in the scheme [11] and y is calculated by the formula $y = \alpha^x \bmod p$, where x is one element of the secret key.

Signature generation procedure:

(1) Compute $R = \alpha^k \bmod p$, where k is a secret random number, $1 < k \leq q - 1$.
(2) Compute $E = F_H(M||R)$.
(3) Compute $S = k - xE \bmod q$, such that $R = \alpha^S y^E \bmod p$.

The signature is the pair (R, S).

Signature verification procedure:

(1) If $|S| \leq \lambda$, then calculating the value of $R^* = \alpha^S y^E \bmod p$. Otherwise, the signature is rejected as invalid.
(2) Compute $E^* = F_H(M||R^*)$.
(3) Compare the values E^* and E. If $E^* = E$, then signature is valid.

Breaking the last signature scheme can be done by simultaneously solving the discrete logarithm problem, which allows to find the secret key x and the factorization problem, which allows to find the value of q, required to compute the value of signature S, whose size will not exceed the value of $\lambda|q|$.

However, the simultaneously solving of these two independent hard problems is not necessary to break this scheme. Indeed, the secret parameters of the scheme can be calculated by solving only the discrete logarithm problem.

This can be done as follow:

We choose an arbitrary number t, the bit length does not exceed the value $\lambda - 1$. Then calculate the value of $Z = \alpha^t \bmod p$. After that we find the logarithm of Z on

the basis of α, using the index calculus algorithm [1]. This gives a value of T, calculated modulo $n = (p - 1)/2$. With a probability close to 1, the size of this value is equal to $|T| \approx |n| > |t|$. Because α is number with order q over Z_p^* then we have $t = T \bmod q$, so q evenly divides the difference between $T - t$. This means that by following the factorization of $T - t$, we can find the secret parameter q. The probability that a factorization of $T - t$ will have a relatively low complexity is quite high. This means that following the above procedure several times, we will find the value of $T - t$, which can be easily factored.

Thus, for breaking of the two DSSes in this section, we only need to solve discrete logarithm problem modulo a prime. In order to design the DSS, which requires simultaneous solving both the factorization problem and the discrete logarithm problem to break, the last signature scheme should be modified. For example, one can use the value α having order equal to n and introduce a new mechanism for calculating the value S, which will require knowledge of the factors of n while computing S.

3 New Signature Schemes Based on Difficulty of Solving Simultaneously Two Hard Problems

In this section, we propose two new signature schemes from two hard problems. Breaking the modified signature schemes described below requires simultaneous solving two different hard problems, computing discrete logarithm in the ground field $GF(p)$ and factoring n.

3.1 The First Scheme

The following modifications have been introduced in the first signature scheme: (i) as parameter α it is used a value having order equal to n modulo p; (ii) instead of the value S in the signature verification equation it is introduced the value S^2.

Key generation:

(1) Choose large distinct primes q' and q in the form $4r + 3$, and compute $n = q'q$.
(2) Choose randomly a secret key x with $x \in Z_p^*$.
(3) Compute $y = \alpha^x \bmod p$.

The public key is (p, α, y). The secret key is (x, q', q).

Signature generation procedure:

(1) Compute $R = \alpha^k \bmod p$, where k is a secret random number, $1 < k \leq n - 1$.
(2) Compute $E = F_H(M||R)$.

(3) Calculate the value S, such that $S^2 = k - xE \bmod n$.

The signature is the pair (E, S).

Signature verification procedure:

(1) Compute $R^* = \alpha^{S2} y^E \bmod p$
(2) Compute $E^* = F_H(M||R^*)$.
(3) Compare the values E^* and E. If $E^* = E$, then signature is valid.

It is easy to see that, the advantage of using this exponent 2 (calculate the value S) is computational load smaller compared to larger exponents. The disadvantage is if $S^2 = k - xE \bmod n$ has no solution, the signature cannot be directly generated [1].

3.2 The Second Scheme

The following modifications have been introduced in the second signature scheme: (i) as parameter α it is used a value having order equal to n modulo p; (ii) it is used one additional element e of the public key; (iii) it is used one additional element d of the private key; (iv) instead of the value S in the signature verification equation it is introduced the value S^e. The values e and d are generated like in the RSA cryptosystem [1].

Key generation:

(1) Choose randomly an integer $e \in Z_n$ such that $\gcd(e, n) = 1$.
(2) Calculate a secret d such that $ed \equiv 1 \bmod \phi(n)$.
(3) Choose randomly a secret key x with $x \in Z_p^*$.
(4) Compute $y = \alpha^x \bmod p$.

The public key is (e, α, y). The secret key is (x, d).

Signature generation procedure:

(1) Compute $R = \alpha^k \bmod p$, where k is a secret random number.
(2) Compute $E = F_H(M||R)$.
(3) Calculate the value S, such that $S^e = k - xE \bmod n$, i.e. $S = (k - xE)^d \bmod n$ such that $R = \alpha^{Se} y^E \bmod p$.

The signature is the pair (E, S). It is easy to see that the length of signature is $|E| + |S| \geq 1184$ bits.

Signature verification procedure:

(1) Compute $R^* = \alpha^{Se} y^E \bmod p$.
(2) Compute $E^* = F_H(M||R^*)$.
(3) Compare the values E^* and E. If $E^* = E$, then signature is valid.

4 Novel Short Signature Scheme

One of important problems is developing digital signature schemes with short signature length [9]. To reduce the signature length in the case of DSSes from two hard problems we use signature formation mechanism, which is based on solving a system of equations [10].

We use the signature formation mechanism that can be applied while developing DSSes with three-element signature denoted as (k, g, v).

The mechanism is characterized in using a three element public key with the structure (y, α, β),where $y = \alpha^x \bmod p$; α is the δ order element modulo p, i.e. $\alpha^\delta \bmod p = 1$; β is the γ order element modulo n, i.e. $\beta^\gamma \bmod n = 1$ ($p = 2n + 1$, where $n = q'q$) and in solving a system of three equations while generating signature. The secret key is γ.

In this scheme, q and q' are strong primes and easy to generate using Gordon's algorithm [1]. The prime q and q' are supposed to be of large size $|q| \approx |q'| \geq 512$ bits. Gordon's algorithm allows to generate strong primes q and q' for which the numbers $q - 1$ and $q' - 1$ contain different prime devisors γ' and γ'', respectively.

Some internal relation between the β and n values provides potentially some additional possibilities to factorize modulus n. This defines special requirements to the β element of the public key [10]. One should use composite γ, i.e. $\gamma = \gamma'\gamma''$, where $\gamma' | q - 1$, $\gamma'' | q' - 1$, $\gamma'' \nmid q - 1$ and $\gamma' \nmid q' - 1$. To choose the size of the γ value we should take into account that the β value can be used to factorize the n modulus calculating $\gcd(\beta^i \bmod n - 1, n)$ for $i = 1, 2, \ldots \min\{\gamma', \gamma''\}$. Therefore we should use the 80-bit values γ' and γ''. Thus, for γ we get the following required length: $|\gamma| = 160$ bits.

A secure variant of the DSS with the 480-bit signature length is described by the following verification equation: $k = \left(y^k \alpha^{gH} \bmod p + \beta^{kgv+H} \bmod n\right) \bmod \delta$, where δ is a specified prime number and H is the hash value of the signed message.

The signature generation is performed as follows:

(1) Generate two random number u_1 and u_2 calculate $z_1 = \alpha^{u1} \bmod p$ and $z_2 = \beta^{u2} \bmod n$.
(2) Solve simultaneously three equations:

$$k = (z_1 + z_2) \bmod \delta;\ g = (u_1 - kx)H^{-1} \bmod \delta;\ v = (u_2 - H)k^{-1}g^{-1} \bmod \gamma.$$

Breaking this scheme requires the simultaneously solving of the factorization the modulus n and the discrete logarithm modulo p.

In this scheme the signature length is compared for different DSSes in the case of minimum security level that can be estimated at present as 2^{80} operations [1]. The minimum level of security provided under the following size parameters: $|p| \geq 1024$ bits, $|n| \geq 1024$ bits, $|\delta| \geq 160$ bits and $|\gamma| \geq 160$ bits. It is easy to see that the size of a digital signature is $|k| + |g| + |v| \geq 480$ bits.

5 Security Analysis

This section presents an analysis on the security of the proposed signature schemes. The results show that the new schemes are only broken when two hard problems, DL and FAC, are solved simultaneously.

The first scheme: In this scheme, solving the DL problem in $GF(p)$ is not sufficient for breaking the modified scheme. The solution of the DL problem leads to the computation of the secret key x and to the possibility to calculate the value $S^* = (k - xE) \bmod n$. However, calculating the signature S requires to extract the square root modulo n from the value S^*. The last represent a hard problem until the value n is factorized.

The second scheme: Similar to the first scheme, solving the DL problem in $GF(p)$ is not sufficient for breaking the modified scheme. To break this signature scheme it is required to know the factorization of n. The solution of the DL problem leads to the computation of the secret key x and to the possibility to calculate the value $S^* = (k - xE) \bmod n$. However, to calculate the signature S, it is required to extract the eth root modulo n from the value S^*. This requires factoring the modulus n.

Theorem 1 *If an ORACLE O can solve DL and FAC problems, then it can break the proposed schemes.*

In other words, if an ORACLE O has the prime factors (q', q) of n and (x, k) by solving FAC and DL problems, then (E, S) will be the eligible sign of document M generated by the proposed methods.

We indicate that the following attacks can be used to break the proposed schemes.

Attack 1: In order to break these schemes, the adversary needs to calculate all secrete elements in the systems. In this case, the adversary needs to solve DL problem to calculate values (x, k). Moreover, the adversary also have to solve FAC problem. It means that the adversary have to solve both DL and FAC problems in order to break the proposed schemes.

Attack 2: The adversary may receive values (R, E, S). By selecting S arbitrarily and computing $E = F_H(M||R)$, the adversary try to find S satisfying equation $R = \alpha^{Se} y^E \bmod p$. In order to solve this equation, the adversary also needs to solve both DL and FAC problems.

Attack 3: All attacks on RSA, Rabin, Schnorr [1] can not be successful on the proposed schemes, because these schemes are the combination of two fundamental algorithms.

Table1 Time complexity comparison of the proposed schemes and the scheme of [5]

	Time complexity (our first scheme)	Time complexity (our second scheme)	Time complexity [5]
Key generation	T_{EXP}	$T_{EXP} + T_{INV}$	$T_{EXP} + T_{INV}$
Signature generation	$T_{EXP} + T_{MUL} + T_{SR} + T_H$	$2T_{EXP} + T_{MUL} + T_H$	$3T_{EXP} + 3T_{MUL} + 2T_{SR} + T_H$
Signature verification	$3T_{EXP} + T_{MUL} + T_H$	$3T_{EXP} + T_{MUL} + T_H$	$4T_{EXP} + 2T_{MUL} + T_H$

6 Performance Analysis

The performance of the proposed algorithms is evaluated based on the complexity of the following procedures: key generation, signing generation and verification. For the sack of comparison, we use the following notations. T_{EXP} denotes Time complexity for executing the modular exponentiation. T_{MUL} denotes Time complexity for executing the modular multiplication. T_H denotes Time complexity for performing hash function. T_{SR} denotes Time complexity for executing the modular square root computation. T_{INV} denotes Time complexity for executing the modular inverse computation.

The results in Table 1 show that the proposed scheme have better performance than the previous scheme in [5].

7 Conclusion

This paper presents the ability to efficiently develop signature schemes based on the widely used fundamental schemes. Based on some well-know schemes, RSA, Rabin and Schnorr, we proposed two new signature schemes. The proposed schemes possess the higher security than well-know schemes because they are based on two independently difficult problems.

The paper also introduces a new method for reducing signature length. This leads to the proposed signature schemes have the shortest signature length in comparison with the other schemes based on two hard problems.

References

1. Menezes AJ, van Oorschot PC, Vanstone SA (1996) Handbook of applied cryptography. CRC Press, Boca Raton
2. Pieprzyk J, Hardjono T, Seberry J (2003) Fundamentals of computer security. Springer, New York
3. Harn L (1994) Public-key cryptosystem design based on factoring and discrete logarithms. IEEE Proc Comput Digit Tech 141(3):193–195

4. Tzeng SF, Yang CY, Hwang MS (2004) A new digital signature scheme based on factoring and discrete logarithms. Int J Comput Math 81(1):9–14
5. Ismail ES, Tahat NMF (2011) A new signature scheme based on multiple hard number theoretic problems. ISRN Commun Netw
6. Li J, Xiao G (1998) Remarks on new signature scheme based on two hard problems. Electron Lett 34(25):2401–2402
7. Chen T-H, Lee W-B, Horng G (2005) Remarks on some signature schemes based on factoring and discrete logarithms. Appl Math Comput 169:1070–1075
8. Buchmann J, May A, Vollmer U (2006) Perspectives for cryptographic long term security. Commun ACM 49(9):50–55
9. Boneh D, Lynn B, Shacham H (2001) Short signatures from the Weil pairing. In: ASIACRYPT '01, vol 2248. LNCS, pp 514–532
10. Moldovyan NA (2009) Short signatures from difficulty of factorization problem. Int J Netw Secur 8(1):90–95
11. Dernova ES (2009) Information authentication protocols based on two hard problems. PhD Dissertation, St.Petersburg State Electrotechnical University. St. Petersburg, Russia