Iran J Sci Technol Trans Electr Eng (2019) 43 (Suppl 1):S277–S287
https://doi.org/10.1007/s40998-018-0129-6

RESEARCH PAPER

New Blind Signature Protocols Based on Finite Subgroups with Two-Dimensional Cyclicity

Hai Nam Nguyen[1] · Duc Tan Nguyen[2] · Minh Hieu Nguyen[1] · Nikolay Adreevich Moldovyan[3]

Received: 14 November 2016 / Accepted: 14 August 2018 / Published online: 28 August 2018

Abstract

In this paper, we propose two new blind digital signature protocols based on the difficulty of the discrete logarithm problem (DLP) modulo a composite number $n = p \cdot q$. These are the first protocols of such type that are based on the computational difficulty of the DLP modulo a composite number. The use of the last difficult problem provides increased security of the signature protocols due to reducing the probability of the potential breaking the protocols, which is connected with potential appearance of the breakthrough solutions of the following two computationally difficult problems: (1) finding discrete logarithm modulo prime and (2) factoring composite number n containing two unknown prime divisors. The designed protocols are based on using finite groups possessing two-dimensional cyclicity. When selecting parameters providing 80-bit security, the signature size in the proposed blind protocols is equal to 240 bits.

Keywords Cyclicity · Cryptographic protocol · Blind digital signature · Hard problem · Factoring · Discrete logarithm

1 Introduction

In automated information systems, digital signatures (DSs) to electronic documents are usually computed using public-key DS protocols (Menezes et al. 1996). The security of DS protocols is based on the following two facts: (1) the best known algorithms for forging a signature are computationally infeasible, and (2) the probability of the appearance of a breakthrough algorithm for solving the computationally difficult problem put in the base of the protocols is negligibly small.

In practice, the well-known DS schemes based on the difficulty of the discrete logarithm problem (DLP) (Menezes et al. 1996) have the 4ρ-bit signature size and provide ρ-bit security (i.e. successful forging a signature requires 2^{ρ} exponentiation operations). The DS schemes based on the difficulty of the integer factoring problem (IFP) usually have a much larger signature size (Menezes et al. 1996; Rivest et al. 1978).

An important class of digital signature schemes involved in information technologies relates to the blind digital signature (DS) protocols (Chaum 1982, 1983). The last are used in online voting systems and electronic cash schemes (Chaum et al. 1988, 1989). The basic idea of the blind signature schemes, first introduced by Chaum (1982), is that the person that signs some electronic message M (signer) does not know the content of the message M. The person presenting M for signing is called requester. They are formulated in the following two requirements to the blind DS protocols: (1) the signer cannot get access to the content of the message, while signing M; (2) later after the signature generation the signer cannot link signed message to requester. The last requirement is known as the requirement of anonymity (untraceability). Thus, blind DS protocols are useful for applications in which anonymity is desirable (Abe 2001; Boldyreva 2003; Fischlin and

✉ Minh Hieu Nguyen
hieuminhmta@gmail.com

Hai Nam Nguyen
nnhaivn61@gmail.com

Duc Tan Nguyen
tanducslc@gmail.com

Nikolay Adreevich Moldovyan
nmold@mail.ru

[1] Academy of Cryptography Techniques, Ha Noi, Vietnam

[2] Posts and Telecommunications Institute of Technology, Ha Noi, Vietnam

[3] St.Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, St. Petersburg, Russia 199178

Schroder 2009; Fuchsbauer et al. 2015; Hanzlik and Kluczniak 2016; Moldovyan et al. 2012).

To improve the security of blind DS protocols, the papers (Minh et al. 2012; Tahat et al. 2008, 2009) propose a design for which forging a signature requires simultaneously solving two independent computationally difficult problems: the IFP and the DLP. In these blind DS protocols, the DLP with the prime p having the following structure $p = en + 1$ is used, where the composite number n is difficult for factoring. Such blind DS protocols are composed so that forging a signature requires both computing the discrete logarithm problem modulo p and factoring n. However, the known design method gives sufficiently low performance, large size of the signature, and it is not evident how to apply it to construct the crypto schemes of other type based on difficulty of simultaneously solving the DLP and IFP.

The present paper introduces a new method to design the blind DS protocols, which requires both the IFP and the DLP to be solved simultaneously. The idea of the proposed method relates to the fact that, excluding the exhaustive search algorithms, finding discrete logarithm modulo a composite number n can be performed by factoring n and finding discrete logarithm modulo for each prime factor (Menezes et al. 1996). Therefore, if the difficulty of factoring n is sufficiently large and approximately equal to the difficulty of finding DL modulo the largest prime factor of n, while using the best algorithms for solving these problems, then breaking some cryptosystem based on difficulty of computing DL modulo n requires solving two different difficult problems simultaneously, the IFP and the DLP. Therefore, the difficulty of breaking a blind DS protocol based on the DL modulo n problem does not change its order in the case when a breakthrough algorithm for solving the IFP or for solving the DLP will be invented.

In this paper, it is shown that using the computational complexity of the DLP modulo a composite number that is difficult for factoring enables one to significantly reduce the signature size and simultaneously increase the performance of the protocol. Besides, using this computationally difficult problem represents an attractive approach for extending the class of cryptographic protocols based on the computational difficulty of simultaneously solving the IFP and DLP modulo a prime number.

We consider the construction of a blind signature protocol with a 3ρ-bit signature size, which provides the ρ-bit security. The proposed protocols are based on the difficulty of the DLP modulo a composite number. These protocols are constructed using finite non-cyclic subgroup G of the multiplicative group Z_n^*, namely subgroup with two-dimensional cyclicity, i.e. subgroup generated by two generators of the same prime order r, such subgroup contains r^2 elements. The paper considers probabilistic and deterministic methods for setting such finite groups.

The rest of the paper is organized as follows: In Sect. 2, two methods are described for setting finite subgroups with two-dimensional cyclicity. In Sect. 3, we construct a new blind DS protocol and a new blind collective digital signature (CDS) protocol based on the difficulty of the DLP modulo a composite number. In Sect. 4, we analyse the output. The last section concludes our research work.

2 Setting Subgroups with Two-Dimensional Cyclicity

2.1 Deterministic Method

In this section, we construct a non-cyclic subgroup G of the multiplicative group Z_n^* of a finite ring Z_n, where n is a natural number equal to the product of two strong primes q and p (Gordon 1985) having the size $|q| \approx \lambda$ bits and $|p| \approx 2\lambda$ bits, respectively. The parameter λ is selected depending on the required security level, for example, $\lambda \approx 512$ bits in the case of 80-bit security and $\lambda \approx 1232$ bits in the case of 128-bit security. Numbers q and p are secret and have the following structure: $p = N_p r + 1$ and $q = N_q r + 1$, where N_p and N_q are two large even numbers; r is a ρ-bit prime number ($\rho = 80$ in the case of a security level equal to 2^{80} exponentiation operations).

One can note that the values p and q represent members of the arithmetic progression $1 + ir$, where $i = 1, 2,\ldots$. The Dirichlet's theorem shows the existence of many infinite prime numbers in arithmetic progressions $a + ib$ with relatively prime values a and b. The progression $1 + ir$ satisfies the last condition; therefore, the required pair of primes p and q exists. The prime number theorem (PNT) by Gauss describes asymptotic distribution of the prime numbers among the positive integers x as follows: $\psi(x) = x(\ln x)^{-1}$, where $\psi(x)$ is the prime counting function and $\ln x$ is the natural logarithm of x. Using the PNT, one can estimate that a uniformly random positive integer $p' < x$ is prime with probability $\Pr = x(\ln x)^{-1}$. For the cases $x < 2^{1232}$ and $x < 2^{2464}$, we have $\Pr \approx 0.0012$ and $\Pr \approx 0.0006$, correspondingly. For the cases of the 1232-bit uniformly random x ($2^{1231} < x < 2^{1232}$) and 2464-bit uniformly random x ($2^{2463} < x < 2^{2464}$), we have $\Pr \approx 0.0006$ and $\Pr \approx 0.0003$, correspondingly.

The subgroup G has the order r^2 and is generated by two integers α and β that generate two different cyclic subgroups of Z_n^*, which have order equal to the prime r.

We use the following deterministic algorithm to find values α and β that generate a non-cyclic primary subgroup G having order equal to r^2.

Algorithm 1:

(1) Generate a value γ having order equal to r modulo p.

(2) Generate a value δ having order equal to r modulo q.

(3) Select at random values $0 < h < r$ and $0 < k < r$, and find the value α that satisfies the following system of congruence:

$$\begin{cases} \alpha \equiv \gamma^k \bmod p \\ \alpha \equiv \delta^h \bmod q \end{cases}. \tag{1}$$

(4) Select at random values $0 < g < r$ and $0 < m < r$ satisfying the condition $gh \neq km \bmod r$, and find the solution β of the following system of congruence:

$$\begin{cases} \beta \equiv \gamma^g \bmod p \\ \beta \equiv \delta^m \bmod q \end{cases}. \tag{2}$$

The output parameters α and β belong to different cyclic subgroups having order r; thus, the products (modulo n) of all possible powers of the values α and β compose a primary subgroup having order r^2. The order of each of the numbers α and β is equal to r since the following formulas hold:

$$\left\{\left\{\alpha^r \equiv \gamma^{kr} \equiv 1 \bmod p\right\} \cup \left\{\alpha^r \equiv \delta^{hr} \equiv 1 \bmod q\right\}\right\} \Rightarrow \alpha^r \equiv 1 \bmod n; \tag{3}$$

$$\{\{\beta^r \equiv \gamma^{gr} \equiv 1 \bmod p\} \cup \{\beta^r \equiv \delta^{mr} \equiv 1 \bmod q\}\} \Rightarrow \beta^r \equiv 1 \bmod n. \tag{4}$$

The following statement also holds.

Statement 1 The output of Algorithm 1 is the values α and β such that inequality $\alpha \neq \beta^d \bmod n$ holds for all values $d \in \{1, 2, \ldots, r\}$.

Proof Clearly, the inequality $\alpha \neq \beta^r \bmod n$ holds. Suppose that for some value $d \in \{1, 2, \ldots, r\}$, the equality $\alpha = \beta^d \bmod n$ holds.

From (1), one can obtain the following:
$\{\beta^d \equiv \gamma^k \bmod p\} \Rightarrow \{\beta \equiv \gamma^{k/d} \bmod p\}$
and
$\{\beta^d \equiv \delta^h \bmod q\} \Rightarrow \{\beta \equiv \gamma^{h/d} \bmod q\}$
From (2), one can obtain the following:
$\{\gamma^g \equiv \gamma^{k/d} \bmod p\} \Rightarrow \{g \equiv k/d \bmod r\}$
and
$\{\delta^m \equiv \delta^{h/d} \bmod q\} \Rightarrow \{m \equiv h/d \bmod r\}$.

Therefore, we have $\{d \equiv k/g \bmod r\}$ and $\{d \equiv h/m \bmod r\}$; thus, $km \equiv hg \bmod r$. This result contradicts condition $gh \neq km \bmod r$, used in step 4 of the algorithm when choosing g and m. The assertion is thus proven. □

Due to Statement 1, the product (mod n) of all possible powers of the values α and β generates r^2 different values of the form $\alpha^i\beta^j \bmod n$, each of which has order equal to r:

$$\left(\alpha^i\beta^j\right)^r \equiv \alpha^{ir}\beta^{jr} \equiv 1 \cdot 1 \equiv 1 \bmod n.$$

2.2 Probabilistic Method

In this section, we construct a non-cyclic subgroup G of the multiplicative group Z_n^* of a finite ring Z_n, where n is a natural number equal to the product of two strong primes q and p having the size $|q| \approx |p| \approx 512$ bits. Numbers q and p are secret and have the following structure: $p = N_p r^2 + 1$, and $q = N_q r^2 + 1$, where N_p and N_q are two large even numbers containing a large prime divisor; r is a ρ-bit prime number ($\rho = 80$ in the case of a security level equal to 2^{80} exponentiation operations). The multiplicative group Z_n^* of a finite ring Z_n is generated by a basis containing two elements. This result follows from the fact that the value of the generalized Euler function $L(n)$ of the number n is less than the value of the Euler function $\phi(n)$ of the number n:

$$\begin{aligned} \phi(n) &= (q-1)(p-1) \\ &= \gcd(q-1, p-1)\mathrm{lcm}[q-1, p-1] \\ &= \gcd(q-1, p-1)L(n) \geq r^2 L(n), \end{aligned}$$

where gcd is the greatest common divisor and lcm is the least common multiple.

We use a primary subgroup G having order r^2 of the multiplicative group Z_n^*, having two-dimensional cyclicity and generated by two elements α and β having prime order r.

All the elements of the subgroup G, except an identity element, have order r. The values of the basic elements α and β are generated by the following probabilistic algorithm:

Algorithm 2:

(1) Select a random value b such that $1 < b < n$;

(2) Compute the values $\gamma = L(n)/r$ and $z = b^\gamma \bmod n$;

(3) If $z \neq 1$, then the number α (the number β) takes the number z. Otherwise, repeat steps 1–3.

The correctness of the proposed algorithm is easy to prove. Indeed, if $z \neq 1$ holds for the generated number z then the equation $z = b^{L(n)/r} \bmod n$ also holds, and therefore according to the generalized Fermat theorem $z^r \equiv b^{L(n)} \equiv 1 \bmod n$, i.e. the order of the number z equals r. (It is known that if $z^r \equiv 1 \bmod n$ holds the order of z divides number r. Since the number r is prime divisor of the value $L(n)$, r is the order of some numbers modulo n.) When implementing this procedure twice, the two random numbers of an order r modulo n could be obtained.

The probability that these two numbers belong to the same cyclic subgroup is the ratio of the number of nonidentity elements in the cyclic subgroup of prime order r to the number of all elements having order r and contained in Z_n^*. Group Z_n^* contains the primary subgroup of order r^2 and is generated by two elements having order r. This primary subgroup contains $r^2 - 1$ elements having order r. Consequently, the previously specified probability is $r/(r^2 - 1) \approx 1/r \approx 2^{-80}$. This probability can be ignored, and the time-consuming procedure verifies that the generated numbers α and β that belong to the same cyclic subgroup of order r do not have to be performed.

This probability can be reduced to a value $\approx 2^{-160}$ when generating numbers α and β according to the following modified algorithm:

Algorithm 3:

(1) Select a random value b such that $1 < b < n$;

(2) Compute the values $\gamma = L(n)/r^2$ and $z = b^{\gamma} \bmod n$;

(3) If $z \neq 1$ and $\alpha'(\beta') = z^r \bmod n \neq 1$, then the number α' (the number β') takes the number $\alpha'^r \bmod n$ (the number $\beta'^r \bmod n$). Otherwise, repeat steps 1–3.

Reducing the probability is achieved because pre-generated numbers have order r^2; this number is raised to the r power, and the result is taken as the number α (the number β).

If the number generated by α' and β' having order r^2 belongs to different cyclic subgroups G_{p^2},, then the numbers α and β will also belong to different cyclic subgroups. The probability $\Pr(\alpha', \beta' \in G_{p^2})$ that α' and β' are in the same cyclic subgroup is the ratio of the number of the elements having order r^2 contained in a cyclic subgroup to the number of elements having order r^2 contained in Z_n^*.. Given the presence of primary subgroups in Z_n^*, generated by two elements having order r^2, expressing the number of elements of primary groups in a given order obtains the following estimated probability:

$$\Pr(\alpha',\ \beta' \in G_{p^2}) = \frac{r(r-1)}{r^2(r^2-1)} \approx \frac{1}{r^2} \approx 2^{-160.}. \tag{5}$$

Thus, the second algorithm for generating random values of α and β is preferred because it reduces significantly the probability of generating values α and β belonging to the same cyclic subgroup by a factor approximately equal to 2^{80}.

To generate a prime p having the form $p = Nr + 1$ and size $\approx \lambda$, where r is some given ρ-bit prime number, one can use the following algorithm:

Algorithm 4:

(1) Generate a random prime π having size $\lambda - \rho$ and compute the integer $p = 2\pi r + 1$.

(2) Set counter $i = 0$.

(3) Generate a random integer $\mu < p$.

(4) If the following four conditions hold: $\mu^{2\pi r} = 1 \bmod p$, $\mu^{\pi r} \neq 1 \bmod p$, $\mu^{2\pi} \neq 1 \bmod p$ and $\mu^{2r} \neq 1 \bmod p$, then go to step 6, otherwise go to step 5.

(5) If $i < 20$, then go to step 3, otherwise go to step 1.

(6) Output the prime number $p = 2\pi r + 1$.

To generate a prime p having the form $p = Nr^2 + 1$ and size $\approx \lambda$, where r is some given ρ-bit prime number, one can use the following algorithm:

Algorithm 5:

(1) Generate a random prime π having size $\lambda - 2\rho$ and compute the integer $p = 2\pi r^2 + 1$.

(2) Set counter $i = 0$.

(3) Generate a random integer $\mu < p$.

(4) If the following four conditions hold: $\mu^{2\pi r^2} = 1 \bmod p$, $\mu^{\pi r^2} \neq 1 \bmod p$, $\mu^{2\pi r} \neq 1 \bmod p$ and $\mu^{2r^2} \neq 1 \bmod p$, then go to step 6, otherwise go to step 5.

(5) If $i < 20$, then go to step 3, otherwise go to step 1.

(6) Output the prime number $p = 2\pi r + 1$.

Algorithms 4 and 5 work correctly, since fulfilment of the conditions indicated in step 4 means that the number μ has order $\omega_p = p - 1$ modulo p. Indeed, due to Euler theorem for any composite number n there exist no numbers having order $n - 1$ modulo n.

For the case $\lambda = 2464$ and $\rho = 128$, the simulation of Algorithm 4 as well as the simulation of Algorithm 5 produce the prime p after performing on average about 2000 rounds of the cycle including the passes 2–5 (the number of different checked primes π).

We have composed a computer program implementing algorithm. To generate 2464-bit prime number, the program processed several minutes, while using the computer Core i5 (3.2 GHz, RAM 4 Gbyte).

One can easily modify the described algorithm to reduce significantly the time required for generating primes having large size; however, Algorithm 4 is sufficiently practical, since primes of the form $p = 2\pi r + 1$ are to be generated only at the stage of creating private and public keys.

3 New Blind Signature Protocol Based on the DL Problem

3.1 Underlying DS Scheme

(a) Key Generation

Initially, a 240-bit signature scheme (the underlying DS scheme) is constructed and used as the base DS algorithm while designing the blind DS protocol.

In the base DS scheme, it is assumed that the parameters n, α, β and r are generated by a trusted party using randomly selected strong primes p and q having the size that provides the required security level. After computing the parameters n, α, β and r, the secret values p and q are destroyed.

The user generates his private key as a pair of random integers x and w ($1 < x < r$; $1 < w < r$) and computes the public key y in accordance with the following formula: $y = \alpha^x \beta^w \bmod n$.

When generating a digital signature, the signer uses only the secret values x and w; thus, to forge the signature of the signer, it is sufficient for a potential attacker to calculate x and w from the known public key y and basic α and β. This problem is known as the DLP on a multi-dimensional basis, in the considered case, on a two-dimensional base. The detailed DLP modulo n is given below, where it is shown that this problem has the same order of complexity as the factoring problem. If there is an effective polynomial algorithm for computing the two-dimensional logarithm, it can be converted into a polynomial algorithm for factoring modulo n. Because the IFP and DLP modulo prime are independent difficult problems, the logarithm modulo prime and factoring modulo composite are significantly different problems.

3.1.1 The Special Case of the Discrete Logarithm Modulo a Composite Number

It is interesting to consider the case of solving the discrete logarithm modulo a composite number with a special structure. Choose the modulus n that is equal to the product of two large prime numbers p and q (i.e. $n = pq$), such that $p - 1$ and $q - 1$ contain the same large prime factor r. In this case, a finite group of all invertible numbers modulo n is generated by a basis containing two elements α and β, the orders of which contain divisor r. Suppose that the value of y contained in this group is given and it is required to compute values x and w such that $y = \alpha^x \beta^w \bmod n$ holds. This problem can be called a two-dimensional DLP or DLP with a two-dimensional base. Let us show that this problem can be reduced to the usual discrete logarithm problem. From the last formula, we can derive the following system of congruencies

$$\begin{cases} y = \alpha^x \beta^w \equiv g^c \bmod p \\ y = \alpha^x \beta^w \equiv g'^{c'} \bmod q \end{cases}, \tag{6}$$

where g and g' are primitive roots modulo p and q, respectively. Solving the usual discrete logarithm problem several times, we obtain the values c, c', u, v, u' and v', such that

$$a = g^u \bmod p, b = g^v \bmod p,$$

$$a = g'^u \bmod q, b = g'^v \bmod q.$$

Using the computed values c, c', u, v, u' and v', the following system of congruencies can be written:

$$\begin{cases} g^{ux+vw} \equiv g^c \bmod p \\ g'^{u'x+v'w} \equiv g'^{c'} \bmod q \end{cases}. \tag{7}$$

From the last system, we obtain the following system:

$$\begin{cases} ux + vw \equiv c \mod (p-1) \\ u'x + v'w \equiv c' \mod (q-1) \end{cases}. \tag{8}$$

From this system of two linear congruencies, we compute the unknowns x and w that determine the "coordinates" of the required two-dimensional discrete logarithm values.

(b) DS Generation

- Select at random values $1 < k < r$ and $1 < t < r$ and calculate $R = \alpha^k \beta^t \bmod n$.
- Using some specified 2ρ-bit hash function $F_H(M)$ [1], calculate the first ρ-bit element E of the signature: $E = F_H(M, R) \bmod r$.
- Calculate the second ρ-bit element S of the signature: $S = (k + xE) \bmod r$.
- Calculate the third ρ-bit element U of the signature: $U = (t + xE) \bmod r$.

The triplet of numbers (E, S, U) is the signature to the document M. The signature size is fixed and equals 3ρ.

The value ρ should be consistent with the size of the modulus n, and both the value ρ and the value λ are chosen depending on the required security of the protocol. To provide 80-bit (128-bit) security, use the parameters $\rho \geq 80$ ($\rho \geq 128$) and $\lambda \geq 512$ ($\lambda \geq 1232$).

(c) DS Verification

- Compute $\tilde{R} = y^{-E} \alpha^S \beta^U \bmod n$ and $\tilde{E} = F_H(M \| \tilde{R}) \bmod r$.
- Compare values $\tilde{E}$ and E. If $\tilde{E} = E$, then the signature is valid. Otherwise, the signature is false.

3.2 New Blind DS Protocol

Our blind digital signature protocol consists of three phases and two parties (the user **A** (requester) and the signer **B**).

The new blind DS protocol works as follows:

(a) Key Generation

The signer **B** generates his private key as a pair of the random integers x and w ($1 < x < r$; $1 < w < r$) and computes the public key y in accordance with the following formula: $y = \alpha^x \beta^w \bmod n$.

(b) Blind DS Generation

There are four rounds in the blind DS protocol. The signer signs an unknown message M blindly as follows.

- **Signer B** Round 1: Select at random values $1 < k < r$ and $1 < t < r$, and calculate $\bar{R} = \alpha^k \beta^t \bmod n$. Then, send $\bar{R}$ to the user **A**.
- **User A** Round 2: Generate three random values ε, μ and τ such that 1 < ε, μ, τ < *r*, and compute

$$R = \bar{R}^{\varepsilon} y^{\mu} \alpha^{\tau} \bmod n,\ E = F_H(R\|M) \bmod r$$
$$and\ \bar{E} = \varepsilon^{-1}(E + \mu) \bmod r.$$

If $\bar{E} = 0$, then repeat step 2 with new random values of blind parameters.

Otherwise, send $\bar{E}$ to the signer **B.** (The value E is the first element of the signature to a message M.) (Fig. 1).

- **Signer B** Round 3: Using his individual values t, k and his secret keys x, w, compute $\bar{S} = (k + x\bar{E}) \bmod r$ and $\bar{U} = (t + w\bar{E}) \bmod r$.
  Then, send $\bar{U}$, $\bar{S}$ to the user **A**.
- **User A** Round 4: Compute the second and third parameters of the blind DS $S = \varepsilon\bar{S} + \tau \bmod r$ *and* $U = \varepsilon\bar{U} \bmod r$.

The triplet of numbers (E, S, U) is a blind DS to the message M, and the signature size is $|E| + |S| + |U| \approx 3|r| \approx 240\ bit$.

(c) Blind DS Verification

- *Step 1* Using the blind DS (E, S, U), compute the following values:
  $\tilde{R} = y^{-E} \alpha^S \beta^U \bmod n\ and\ \tilde{E} = F_H(\tilde{R}\|M) \bmod r\,.$
- *Step 2* Compare values $\tilde{E}$ and E. If $\tilde{E} = E$, then the signature is valid. Otherwise, the signature is false.

3.3 New Blind CDS Protocol

The blind DS protocol proposed in Sect. 3.2 can be used to design the blind collective DS (CDS) protocol.

In this section, we propose a blind CDS protocol for a broadcasting structure. The protocol consists of three phases: the key generation phase, the blind CDS generation phase and the blind CDS verification phase.

Suppose that the signing group {$\mathbf{B}_1$, $\mathbf{B}_2$, …, $\mathbf{B}_m$} wants to generate a blind CDS for a message M proposed for blind signing by a user **A**.

(a) Key Generation

The group signers generate parameters:

- $x_1, x_2, \ldots, x_m$ and $w_1, w_2, \ldots, w_m$: group signers' secret keys such that $1 < x_i < p$ and $1 < w_i < p$, x_i, w_i (i = 1, 2,…, m) are selected randomly and known only to the signer $\mathbf{B}_i$.
- $y_1, y_2, \ldots, y_m$: group signers' public keys such that $y_i = \alpha^{x_i} \beta^{w_i} \bmod n$ are computed and published by the group signers $\mathbf{B}_i$.

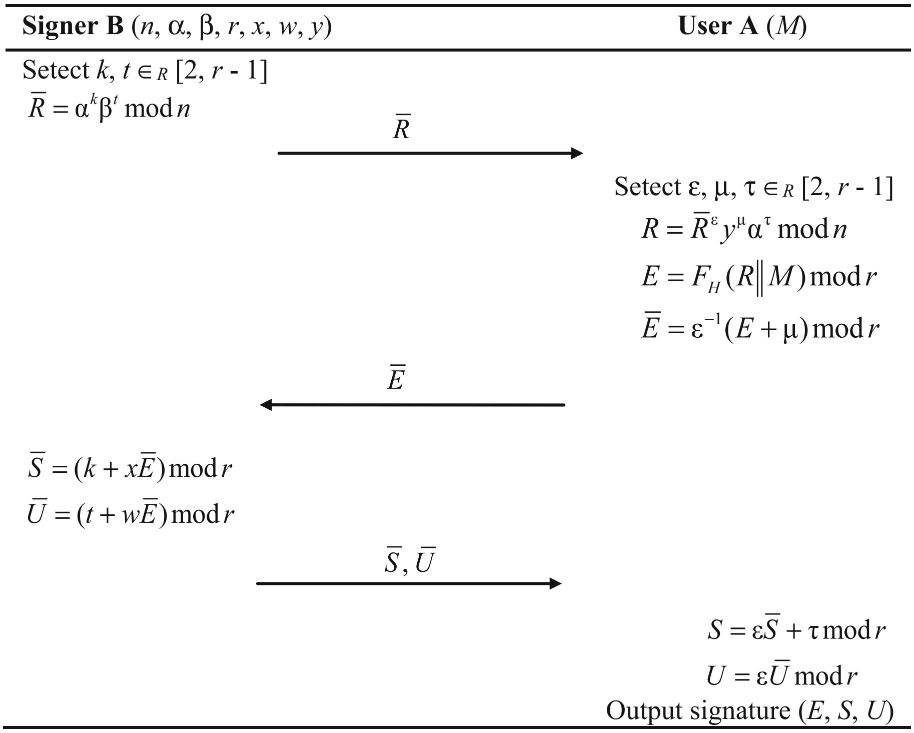
Signer B (n, α, β, r, x, w, y)
User A (M)
Setect $k, t \in_R [2, r - 1]$
$\bar{R} = \alpha^k \beta^t \bmod n$
$\bar{R}$
Setect $\varepsilon, \mu, \tau \in_R [2, r - 1]$
$R = \bar{R}^{\varepsilon} y^{\mu} \alpha^{\tau} \bmod n$
$E = F_H(R\|M) \bmod r$
$\bar{E} = \varepsilon^{-1}(E + \mu) \bmod r$
$\bar{E}$
$\bar{S} = (k + x\bar{E}) \bmod r$
$\bar{U} = (t + w\bar{E}) \bmod r$
$\bar{S}, \bar{U}$
$S = \varepsilon\bar{S} + \tau \bmod r$
$U = \varepsilon\bar{U} \bmod r$
Output signature (E, S, U)

Fig. 1 Our blind digital signature protocol

- The collective public key y is computed as a convolution of the set of individual public keys y_i of all signers: $y = \prod_{i=1}^{m} y_i \bmod n$.
- Blind CDS Generation

There are four rounds in the blind CDS protocol in which each signer signs an unknown message M blindly.

- **Signers** Round 1: Each signer generates a random value $1 < k_i < r$ and $1 < t_i < r$ and computes $r_i = \alpha^{k_i}\beta^{t_i} \bmod n$, then sends r_i to all signers. The common randomization parameter is computed as the product $\bar{R} = \prod_{i=1}^{m} r_i \bmod n$, and the value $\bar{R}$ is sent to the user **A**.
- **User A** Round 2: Generates three random values ε, μ and τ such that $1 < \varepsilon,\ \mu,\ \tau < r$, and computes

$$R = \bar{R}^{\varepsilon} y^{\mu} \alpha^{\tau} \bmod n,\ E = F_H(R\|M) \bmod r$$
$$\text{and } \bar{E} = \varepsilon^{-1}(E + \mu) \bmod r.$$

If $\bar{E} = 0$, then repeat step 2 with new random values of blinding parameters. Otherwise, send $\bar{E}$ to the signers $\mathbf{B}_i$ (the value E is the first element of the signature to a message M).

- **Signers** Round 3: Each signer using his individual values t_i, k_i and his secret keys x_i, w_i computes $s_i = (k_i + x_i\bar{E}) \bmod r$ and $u_i = (t_i + w_i\bar{E}) \bmod r$,

then sends s_i and u_i to all signers.

Compute the common parameters

$$\bar{S} = \sum_{i=1}^{m} s_i = (\sum_{i=1}^{m} k_i + \bar{E}\sum_{i=1}^{m} x_i) \bmod r$$

and $\bar{U} = \sum_{i=1}^{m} u_i = (\sum_{i=1}^{m} t_i + \bar{E}\sum_{i=1}^{m} w_i) \bmod r$.

Then, send $\bar{U}$, $\bar{S}$ to the user **A**.

- **User A** Round 4: Compute the second and third parameters of the blind CDS

$S = \varepsilon\bar{S} + \tau \bmod r$ and $U = \varepsilon\bar{U} \bmod r$.

The triplet of numbers (E, S, U) is a blind CDS to the message M, and the signature size is $|E| + |S| + |U| \approx 3|r| \approx 240$ bit.

The blind CDS size does not depend on the number of signers and its size is equal to 3ρ.

(b) Blind CDS Verification

The blind CDS verification procedure uses the collective public key Y.

- *Step 1* Using the blind CDS (E, S, U), compute the following values:

$$\tilde{R} = y^{-E}\alpha^{S}\beta^{U} \bmod n \textit{ and } \tilde{E} = F_H(\tilde{R}\|M) \bmod r.$$

- *Step 2* Compare values $\tilde{E}$ and E.

If $\tilde{E} = E$, then the signature is valid. Otherwise, the signature is false.

4 Analysis of Our Protocols

In this section, we analyse the security and efficiency of our proposed blind signature protocols.

4.1 Correctness

Theorem 1 (DS) *The triplet of numbers* (E, S, U) *is a valid DS corresponding to the message M.*

Proof Substituting the values y, U and S in the right side of the verification equation $\tilde{R} = y^{-E}\alpha^{S}\beta^{U} \bmod n$, we obtain the following:

$$\begin{aligned}\tilde{R} &= y^{-E}\alpha^{S}\beta^{U} \bmod n = (\alpha^{x}\beta^{w})^{-E}\alpha^{S}\beta^{U} \bmod n \\ &= \alpha^{-Ex}\beta^{-Ew}\alpha^{(k+xE)}\beta^{(t+wE)} \bmod n = \\ &= \alpha^{-Ex}\beta^{-Ew}\alpha^{k}\alpha^{xE}\beta^{t}\beta^{wE} \bmod n = \alpha^{k}\beta^{t} \bmod n = R. \\ &\Rightarrow \tilde{E} = F_H(\tilde{R}\|M) = F_H(R\|M) = E.\end{aligned}$$

The last equality proves the correctness of the DS scheme (Fig. 2). □

Theorem 2 (blind DS) *The triplet of numbers* (E, S, U) *is a valid blind DS corresponding to the message M.*

Proof Substituting the values E, U and S in the right side of the verification equation $\tilde{R} = y^{-E}\alpha^{S}\beta^{U} \bmod n$, we obtain the following:

$$\begin{aligned}\tilde{R} &= y^{-E}\alpha^{S}\beta^{U} \bmod n = y^{-\varepsilon\bar{E}+\mu}\alpha^{\varepsilon\bar{S}+\tau}\beta^{\varepsilon\bar{U}} \bmod n \\ &= y^{-\varepsilon\bar{E}}y^{\mu}\alpha^{\varepsilon\bar{S}}\alpha^{\tau}\beta^{\varepsilon\bar{U}} \bmod n = (y^{-\bar{E}}\alpha^{\bar{S}}\beta^{\bar{U}})^{\varepsilon}y^{\mu}\alpha^{\tau} \bmod n \\ &= \bar{R}^{\varepsilon}y^{\mu}\alpha^{\tau} \bmod n = R. \\ &\Rightarrow \tilde{E} = F_H(\tilde{R}\|M) = F_H(R\|M) = E.\end{aligned}$$

Thus, the protocol works correctly, and the described procedure results in the DS (E, S, U) that is known to user **A** and unknown to signer **B**. □

Theorem 3 (blind CDS) *The triplet of numbers* (E, S, U) *is a valid CDS corresponding to the message M.*

Proof We use the collective public key y.

Substituting the values E, U and S in the right side of the verification equation $\tilde{R} = y^{-E}\alpha^{S}\beta^{U} \bmod n$, we obtain the following:

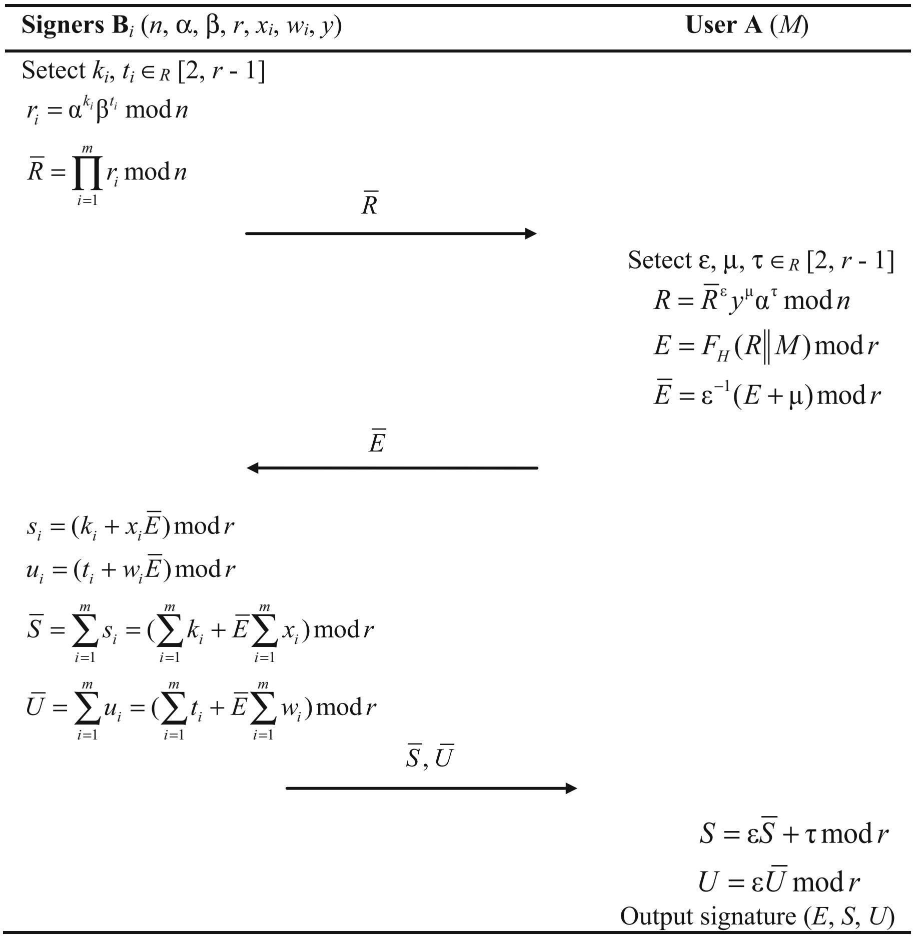

Fig. 2 Our blind collective digital signature protocol

Signers B$_i$ (n, α, β, r, x_i, w_i, y) **User A** (M)

Setect $k_i, t_i \in_R [2, r - 1]$

$$r_i = \alpha^{k_i}\beta^{t_i} \bmod n$$

$$\bar{R} = \prod_{i=1}^{m} r_i \bmod n$$

$$\xrightarrow{\bar{R}}$$

Setect $\varepsilon, \mu, \tau \in_R [2, r - 1]$

$$R = \bar{R}^{\varepsilon} y^{\mu} \alpha^{\tau} \bmod n$$

$$E = F_H(R \| M) \bmod r$$

$$\bar{E} = \varepsilon^{-1}(E + \mu) \bmod r$$

$$\xleftarrow{\bar{E}}$$

$$s_i = (k_i + x_i\bar{E}) \bmod r$$

$$u_i = (t_i + w_i\bar{E}) \bmod r$$

$$\bar{S} = \sum_{i=1}^{m} s_i = (\sum_{i=1}^{m} k_i + \bar{E}\sum_{i=1}^{m} x_i) \bmod r$$

$$\bar{U} = \sum_{i=1}^{m} u_i = (\sum_{i=1}^{m} t_i + \bar{E}\sum_{i=1}^{m} w_i) \bmod r$$

$$\xrightarrow{\bar{S}, \bar{U}}$$

$$S = \varepsilon\bar{S} + \tau \bmod r$$

$$U = \varepsilon\bar{U} \bmod r$$

Output signature (E, S, U)

$$\begin{aligned}
\tilde{R} &= y^{-E}\alpha^{S}\beta^{U} \bmod n = y^{-\varepsilon\bar{E}+\mu}\alpha^{\varepsilon\bar{S}+\tau}\beta^{\varepsilon\bar{U}} \bmod n \\
&= y^{-\varepsilon\bar{E}}y^{\mu}\alpha^{\varepsilon\bar{S}}\alpha^{\tau}\beta^{\varepsilon\bar{U}} \bmod n = (y^{-\bar{E}}\alpha^{\bar{S}}\beta^{\bar{U}})^{\varepsilon}y^{\mu}\alpha^{\tau} \bmod n \\
&= (y^{-\bar{E}}\alpha^{\sum_{i=1}^{m} s_i}\beta^{\sum_{i=1}^{m} u_i})^{\varepsilon}y^{\mu}\alpha^{\tau} \bmod n \\
&= \left(\prod_{i=1}^{m} y_i^{-\bar{E}}\alpha^{s_i}\beta^{u_i}\right)^{\varepsilon} y^{\mu}\alpha^{\tau} \bmod n \\
&= \left(\prod_{i=}^{m} r_i\right)^{\varepsilon} y^{\mu}\alpha^{\tau} \bmod n = \bar{R}^{\varepsilon}y^{\mu}\alpha^{\tau} \bmod n = R. \\
&\Rightarrow \tilde{E} = F_H(\tilde{R}\|M) = F_H(R\|M) = E.
\end{aligned}$$

Thus, the protocol works correctly, and the described procedure results in the CDS (E, S, U) that is known to user **A** and unknown to each of the signers. □

4.2 Unlinkability

Unlinkability In a blind signature scheme, the unlinkability property makes it impossible for the signer to derive the link between a given signature and the instance of the signing protocol that produces the blinded form of that signature.

Theorem 4 (blind DS) *The protocol provides unlinkability in the case in which the message M and signature (E, S, U) will be presented to the signer.*

Proof Let $(\bar{E}_1, \bar{S}_1, \bar{U}_1)$ and $(\bar{E}_2, \bar{S}_2, \bar{U}_2)$, be two different signatures produced blindly and stored by some signer **B**.

In accordance with the equation of the signature generation procedure, we obtain the following relations:

$$\varepsilon = U/\bar{U} \bmod r;\ \tau = S - U\bar{S}/U \bmod r$$
$$\text{and } U = U\bar{E}/U - E \bmod r.$$

These relations show that the signature (E, S, U) could be produced by user $\mathbf{A}_1$ from the triplet $(\bar{E}_1, \bar{S}_1, \bar{U}_1)$. (In this case, the supposed user $\mathbf{A}_1$ used the values ε_1, τ_1 and σ_1.) The same signature can also be produced by the user $\mathbf{A}_2$ with some signer **B** from the triplet $(\bar{E}_2, \bar{S}_2, \bar{U}_2)$. (In this case, the supposed user $\mathbf{A}_2$ had used the values ε_2, τ_2 and σ_2.) Since the values (ε, τ and σ) are selected at random, the signature could be produced from each of two considered triplets as well as from each of the triplets in the database, i.e. the unlinkability property (or blindness property) is provided by the protocol. □

Theorem 5 (blind CDS) *The protocol provides the unlinkability property in the case in which the message M and signature (E, S, U) will be presented to all or to one of the signers.*

Proof We suppose that many different users present electronic messages to some given set of signers for blind signing. Suppose that the signers have saved all triplets $(\bar{E},\ \bar{S},\ \bar{U})$ that appeared in the blind CDS procedures.

Let $(\bar{E}_1, \bar{S}_1, \bar{U}_1)$ and $(\bar{E}_2, \bar{S}_2, \bar{U}_2)$, be two triplets. According to the blind CDS protocol construction, the elements of the first triplet satisfy the following expression:

$$\varepsilon = U/\bar{U} \bmod r;\ \ \tau = S - U\bar{S}/U \bmod r$$
$$\text{and}\ \ U = U\bar{E}/U - E \bmod r.$$

These relations show that the signature (E, S, U) could be produced by user $\mathbf{A}_1$ from the triplet $(\bar{E}_1, \bar{S}_1, \bar{U}_1)$. (In this case, the supposed user $\mathbf{A}_1$ had used the values ε_1, τ_1 and σ_1.) The same signature can also be produced by the user $\mathbf{A}_2$ with the same signers $\mathbf{B}_i$ from the triplet $(\bar{E}_2, \bar{S}_2, \bar{U}_2)$. (In this case, the supposed user $\mathbf{A}_2$ used the values ε_2, τ_2 and σ_2.) Since the values (ε, τ and σ) are selected at random, the signature could be produced from each of two considered triples as well as from each of the triplets in the database, i.e. the unlinkability property (or blindness property) is provided by the protocol. □

4.3 Unforgeability

Unforgeability implies that only the signer(s) can generate valid signatures.

4.3.1 Attack 1 (Outsider attack)

The attacker tries to derive the signature (E, S, U), where $R = y^{-E}\alpha^{S}\beta^{U} \bmod n$ and $E = F_H(R\|M) \bmod r$, for a given message M by fixing one of the values R, E, S and U and finding the other ones. For example, the attacker selects R and tries to figure out the values of E, S and U satisfying $R = y^{-E}\alpha^{S}\beta^{U} \bmod n$ and vice versa. The attacker first chooses at random the value R and then computes the values S and U only if the difficult computational problem of the DLP modulo a composite number is breakable. The attacker first chooses at random value E and then computes R. It is supposed that a secure hash function is used in the protocol; therefore, the attacker is not able to select the value R producing some specially chosen value E. Similar to the case, the attacker first chooses at random the value S (or U) and then computes the values E and U (or S) only if the difficult computational problem of the DLP modulo a composite number is breakable.

4.3.2 Attack 2 (User attack)

The user can know individual signatures, but this does not damage the security of the protocols. If he cannot compute the blind DS (CDS) correctly from the individual signatures, the verification equation of the blind CDS is not satisfied. This type of attack can be detected by the verifier.

4.4 Attack 3 (Signer(s) attack)

Suppose that $m - 1$ signers that share some collective signature $(\bar{E},\ \bar{S},\ \bar{U})$ with the m signer are attackers trying to calculate the secret key of the m signer. The attackers know the values $(r_m,\ s_m,\ u_m)$ generated by the mth signer.

These values satisfy the equation $r_m = y_m^{-\bar{E}}\alpha^{s_m}\beta^{u_m} \bmod n$, where the value $\bar{E}$ is out of the attackers' control. Therefore, computing the secret key of the m signer requires solving the DLP modulo a composite number.

4.5 Performance

Next, we investigate the performance of our protocols using the number of modular multiplications, number of hashing operations, number of random number generations, number of inverse computations and number of modular exponentiations.

Note that the time for computing modular addition and subtraction is ignored since it is much smaller than the time for computing modular exponentiation, modular multiplication and modular inverse.

The comparisons of computational costs performed by the user, signer and verifier between the proposed blind signature protocol and the scheme of (Minh et al. 2012; Tahat et al. 2009) are summarized in Tables 1 and 2.

The comparisons of the numbers of computations performed by a user between the proposed blind CDS protocol and the protocols of (Hieu et al. 2017; Moldovyan and Moldovyan 2011) are summarized in Table 3.

The performance of our proposed blind DS (CDS) protocols is almost equivalent to the protocol (Minh et al. 2012; Hieu et al. 2017; Tahat et al. 2009; Moldovyan and Moldovyan 2011). However, the proposed protocols have signature sizes much shorter than the protocols (Minh et al. 2012; Hieu et al. 2017; Tahat et al. 2009; Moldovyan and Moldovyan 2011) and are thus superior in practice (Table 4).

In most of the applications based on blind signatures, the signer(s) usually possesses much more computational capability than a user, while the computational capability of the users may be limited in certain situations such as mobile clients. To guarantee the quality of these growing popular communication services based on blind signatures,

Table 1 Computational costs of the proposed blind DS protocol and the protocols of (Minh et al. 2012; Tahat et al. 2009)

Type of operation	Performed by the user			Performed by the signer		
	Our protocol	(Minh et al. 2012)	(Tahat et al. 2009)	Our protocol	(Minh et al. 2012)	(Tahat et al. 2009)
Exponentiation	3	3	7	2	2	2
Inversion	1	1	4	0	0	0
Hashing	1	1	3	0	0	1
Multiplication	5	4	11	3	1	2
Random number generation	3	2	2	2	1	1

Table 2 Computational costs of the proposed blind DS protocol and the protocols of (Minh et al. 2012; Tahat et al. 2009)

Type of operation	Performed by the verifier		
	Our protocol	(Minh et al. 2012)	(Tahat et al. 2009)
Exponentiation	3	3	4
Hashing	1	1	1
Multiplication	2	1	1
Numbers of inverses	1	0	0

Table 3 Computational costs of the proposed blind CDS protocol and the protocols of (Hieu et al. 2017; Moldovyan and Moldovyan 2011)

Type of operation	Performed by the user			Performed by the verifier		
	Our protocol	(Hieu et al. 2017)	(Moldovyan and Moldovyan 2011)	Our protocol	(Hieu et al. 2017)	(Moldovyan and Moldovyan 2011)
Exponentiation	3	2	2	3	2	2
Inversion	1	0	0	1	1	1
Hashing	1	1	1	1	1	1
Multiplication	5	3	2	2	1	1
Random number generation	3	2	2			

Table 4 Signature size of the proposed protocols and the protocols of (Minh et al. 2012; Hieu et al. 2017; Tahat et al. 2009; Moldovyan and Moldovyan 2011)

Size of signature (bits)	Our protocols	Minh et al. (2012)	Tahat et al. (2009)	Hieu et al. (2017)	Moldovyan and Moldovyan (2011)
	240	1184	2048	1184	320

it is more urgent to reduce the computational load for users than for signer(s).

5 Conclusions

This paper proposed two new blind DS protocols with a 3ρ-bit signature size that provide ρ-bit security. These protocols are the first ones based on the computational difficulty of the DLP modulo a composite number $n = pq$.

The protocols use finite groups possessing two-dimensional cyclicity. When selecting parameters to provide 80-bit security, the size of the proposed blind DS protocols is 240 bit (and are not dependent on the number of signers).

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable suggestions and comments.

References

Abe M (2001) A secure three-move blind signature scheme for polynomials many signatures. Advances in cryptology—EUROCRYPT 2001, vol 2045. Lecture notes in computer science. Springer, Heidelberg, pp 136–151

Boldyreva A (2003) Threshold signatures, multisignatures and blind signatures based on the gap-Diffe-Hetlman-group signature scheme. PKC 2003: 6th international workshop on theory and practice in public key cryptography, vol 2567. Lecture notes in computer science. Springer, Heidelberg, pp 31–46

Chaum D (1982) Blind signatures for untraceable payments. Advances in Cryptology—CRYPTO'82. Plenum Press, New York, pp 199–203

Chaum D (1983) Blind signature system. Advances in cryptology—CRYPTO'83. Plenum Press, New York, p 153

Chaum D (1989) Privacy protected payment. In: SMART CARD 2000. Elsevier Science Publishers B.V., North-Holland, pp 69–93

Chaum D, Fiat A, Noar M (1988) Untraceable electronic cash. In: Advances in Cryptology (Crypto'88), LNCS, vol 403. Springer, Heidelberg, pp 319–327

Fischlin M, Schroder D (2009) Security of blind signatures under aborts. PKC 2009: 12th international conference on theory and practice of public key cryptography, vol 5443. Lecture notes in computer science. Springer, Heidelberg, pp 297–316

Fuchsbauer G, Hanser C, Slamanig D (2015) Practical round-optimal blind signatures in the standard model. Advances in cryptology—CRYPTO 2015, Part II, vol 9216. Lecture notes in computer science. Springer, Heidelberg, pp 233–253

Gordon J (1985) Strong primes are easy to find. In: Advances in Cryptology—EUROCRYPT'84, LNCS, vol 209. Springer, Heidelberg, pp 216–223

Hanzlik L, Kluczniak K (2016) A short paper on blind signatures from knowledge assumptions. In: Financial Cryptography and Data Security—FC 2016, Springer LNCS

Hieu M, Nam H, Moldovyan N, Tien G (2017) New blind signature protocols based on a new hard problem. Int Arab J Inf Technol 14(3):307–313

Menezes J, Van Oorschot PC, Vanstone SA (1996) Handbook of Applied Cryptography. CRC Press, Boca Raton, p 816

Minh NH, Binh DV, Giang NT, Moldovyan NA (2012) Blind signature protocol based on difficulty of simultaneous solving two difficult problems. Appl Math Sci 6(139):6903–6910

Moldovyan NA, Moldovyan AA (2011) Blind collective signature protocol based on discrete logarithm problem. Int J Netw Secur 12(1):44–51

Moldovyan AA, Moldovyan NA, Novikova ES (2012) Blind 384-bit Digital Signature Scheme. In: MMM-ACNS'12 Proceedings of the 6th international conference on mathematical methods, models and architectures for computer network security: computer network security, St. Petersburg, Russia, Springer, Berlin, Heidelberg, pp 77–83

Rivest R, Shamir A, Adleman A (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126

Tahat NMF, Shatnawi SMA, Ismail ES (2008) New partially blind signature based on factoring and discrete logarithms. J Math Stat 4(2):124–129

Tahat NMF, Ismail ES, Ahmad RR (2009) A new blind signature scheme based on factoring and discrete logarithms. Int J Cryptol Res 1(1):1–9