Computer Science Journal of Moldova, vol.**21**, no.2(62), 2013

# Cryptoschemes Based on Difficulty of Simultaneous Solving Two Different Difficult Problems *

A.N. Berezin, N.A. Moldovyan, V.A. Shcherbacov

**Abstract**

The paper proposes a general method for construction cryptoschemes based on difficulty of simultaneous solving factoring (FP) and discrete logarithm modulo prime problem (DLpP). The proposed approach is applicable for construction digital signatures (usual, blind, collective), public key encryption algorithms, public key distribution protocols, and cryptoschemes of other types. Moreover, the proposed approach provides reducing the signature size and increasing the rate of the cryptoschemes, while comparing with the known designs of the digital signature protocols based on the FP and DLpP.

**Keywords.** Cryptosystem, cryptographic protocol, digital signature, public encryption, factorization problem, discrete logarithm problem, integrated security parameter.

## 1 Introduction

Cryptographic protocols with public key are widely used to provide information security of the information technologies. Usually the public key cryptoschemes are based on difficulty of one computational problem [1]. The problems of factoring and discrete logarithm (DL) modulo a prime are two widely used difficult problems for designing different types of the cryptographic protocols [2]. Their security is based on



* The first author is supported by St.Petersburg Government grant.

the following two facts: i) the used difficult problem is computationally infeasible while applying the best known algorithm, i.e. its difficulty $W$ is sufficiently large; ii) the probability $P$ of the discovery in near future an computationally efficient algorithm for solving the hard problem is negligibly small. Thus, the ratio $W/P$ can be introduced as some integrated security parameter [3]. Therefore increasing the difficulty of the hard problem or reducing the value $P$ leads to increasing the integrated security parameter. The value $P$ can be significantly reduced in the case when breaking the public key cryptoscheme requires solving simultaneously two independent computationally hard problems. The factoring problem (FP) and the problem of finding the discrete logarithm (DL) modulo prime are two widely used problems for constructing the public key cryptoschemes.

A number of signature schemes based on difficulty of the simultaneous solving the factoring problem (FP) and the DL modulo prime problem (DLpP) are proposed in the papers [4], [5]. In these cryptoschemes there is used the DLpP with the prime $p'$ having the following structure $p' = en + 1$, where the composite number $n$ is difficult for factoring. Such signature schemes are composed so that forging a signature requires solving the 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 cryptoschemes of other type based on difficulty of simultaneous solving the DLpP and FP.

The present paper introduces a new method to design the cryptoschemes, breaking of which requires to solve simultaneously both the FP and the DLpP. The proposed method is free of the lacks of the known one, while designing the signature algorithms, and provides possibility to construct cryptoschemes of different other types. Section 2 describes the idea of the proposed method that consists in applying the difficulty of finding the discrete logarithm modulo a composite number that is difficult for factoring. Section 3 describes a new public key agreement scheme. Section 4 describes the public encryption algorithm. Sections 5 and 6 present new signature schemes. Discussion and conclusion are presented in section 7.

## 2 The Proposed Method

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 each prime factor [2]. 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 simultaneously two different difficult problems, the FP and the DLpP.

Difficulty of the factoring number $n = pq$ that is equal to product of two large strong [6] primes $p$ and $q$ is defined by the size of the smaller one, for example $q < p$. Let $|q|$ denote the bit size of the number $q$. The difficulty of finding DL modulo prime $p$ is approximately equal to the difficulty of factoring number $n$, if $|p| = 2|q|$. Suppose one selects $n = pq$ such that $|p| = 2|q|$. Then, the difficulty of finding the DL modulo $n$, the difficulty of finding the DL modulo $p$, and the difficulty of factoring $n$ are values of the same order. Therefore the difficulty of breaking a cryptosystem based on the DL modulo $n$ problem does not change its order in the case when a breakthrough algorithm for solving the FP or for solving the DLpP will be invented. Since the probability of the last two events are sufficiently small, then the probability $P$ is significantly reduced for the cryptoschemes breaking of which requires solving the DL modulo $n$ problem (DLnP).

Thus, constructing the public key cryptoschemes based on the DLnP like the known cryptoschemes based on the DLpP represents an interesting and general approach to get the cryptoschemes based on the difficulty of solving simultaneously the FP and the DLpP. Applying this design method one can extend the type set of algorithms and protocols breaking of which implies simultaneous solving the FP and the DLpP.

The DLnP is defined relatively some known numbers $\alpha$ and $y$, and

consists in finding $x$ such that the following expression holds:

$$y = \alpha^x \bmod n.$$

As it is shown in paper [7] the value $\alpha$ should be selected properly in order to prevent some new factoring methods connected with using the value $\alpha$. The following two variants are possible for selecting secure $\alpha$.

1. In the first variant there are generated primes $p$ and $q$ such that sufficiently large prime $\gamma$ divides both the number $p-1$ and the number $q-1$. A number having order $\gamma$ modulo $n$ is selected as the value $\alpha$. In this case the value $\gamma$ is not secret.

2. In the second variant there are generated primes $p$ and $q$ such that the prime $\gamma'$ divides the number $p-1$ ($\gamma'|p-1$), the prime $\gamma''$ divides the number $q-1$ ($\gamma''|q-1$), $\gamma'$ does not divide $q-1$, and $\gamma''$ does not divide $p-1$. A number having order $\gamma = \gamma'\gamma''$ modulo $n$ is selected as the value $\alpha$. In this case the values $\gamma$, $\gamma'$, and $\gamma''$ are secret.

In the case of construction the cryptoscheme with 128-bit security the values $\gamma$, $\gamma'$, and $\gamma''$ should have the following size: $|\gamma| \geq 256$ bits, $|\gamma'| \geq 128$ bits, and $|\gamma''| \geq 128$ bits. Below in the proposed cryptoschemes there is used modulus $n$ that is the product of the two strong primes $q$ and $p$ having the length $|q| \approx 1232$ bits and the length $|p| \approx 2464$ bits. For such size of the used numbers the difficulty of factoring $n$ and the difficulty of finding DL modulo $p$ are equal to $O(2^{128})$ modulo multiplication operations, where $O(*)$ is the order notation.

## 3 The Public Key Agreement Protocol

The $i$th user generates the secrete prime values $p_i$, $q_i$, and a random 256-bit number $x_i$. Then he computes his public key $(n_i, \alpha_i, y_i)$, where $n_i = p_i q_i$; $y_i = \alpha_i^{x_i} \bmod n_i$; $\alpha_i$ is a number having 256-bit prime order modulo $n_i$. The public keys $(n_i, \alpha_i, y_i)$ are registered at some authority center that publishes all registered public keys in some reference book. Some $i$th user and some $j$th user can compute their common secret key $Z_{ij}$ using the following protocol:

1. The $i$th user generates a random 256-bit number $u_i$, computes the value $R_i = \alpha_j^{u_i} \bmod n_j$, and sends $R_i$ to the $j$th user.

2. The $j$th user generates a random 256-bit number $u_j$, computes the value $R_j = \alpha_i^{u_j} \bmod n_i$, and sends $R_j$ to the $i$th user.

3. The $i$th user selects from the reference book the public key $y_j$ and computes the following values: $Z'_{ij} = y_j^{u_i} \bmod n_j$, $Z''_{ij} = R_j^{x_i} \bmod n_i$, and $Z_{ij} = Z'_{ij} Z''_{ij}$.

4. The $j$th user selects from the reference book the public key $y_i$ and computes the following values: $Z'_{ji} = y_i^{u_j} \bmod n_i$, $Z''_{ji} = R_i^{x_j} \bmod n_j$, and $Z_{ji} = Z'_{ji} Z''_{ji}$.

The secret shared by the $i$th and $j$th users is the value $Z_{ji} = Z_{ij}$.

The protocol correctness proof is as follows:

$$\left.\begin{array}{l} Z'_{ij} = y_j^{u_i} \bmod n_j = \alpha_j^{x_j u_i} \bmod n_j \\ Z''_{ji} = R_i^{x_j} \bmod n_j = \alpha_j^{u_i x_j} \bmod n_j \end{array}\right\} \Rightarrow \quad Z''_{ji} = Z'_{ij} \, ,$$

$$\left.\begin{array}{l} Z'_{ji} = y_i^{u_j} \bmod n_i = \alpha_i^{x_i u_j} \bmod n_i \\ Z''_{ij} = R_j^{x_i} \bmod n_i = \alpha_i^{u_j x_i} \bmod n_i \end{array}\right\} \Rightarrow \quad Z''_{ij} = Z'_{ji} \, ,$$

$$\left\{Z''_{ji} = Z'_{ij}; Z''_{ij} = Z'_{ji}; Z_{ij} = Z'_{ij} Z''_{ij}; Z_{ji} = Z'_{ji} Z''_{ji}\right\} \Rightarrow$$
$$\Rightarrow Z_{ij} = Z_{ji}.$$

## 4 The Public Encryption Algorithm

Suppose some sender wishes to send a secret message $M < n$ to some user using the public communication channel. This can be done using the following public encryption algorithm, like the ElGamal's encryption algorithm [8].

1. The sender selects from the public key reference book the receiver's public key $(n, \alpha, y)$, generates a 256-bit random number $u$, computes the value $R = \alpha^u \bmod n$, and the value $Z = y^u \bmod n$.

2. Then he encrypts the message $M$ as follows: $C = MZ \bmod n$.

3. The cryptogram $(R, C)$ is send to the receiver.

The decrypting message $M$ procedure is as follows:

1. The receiver, using his secret key $x$ such that $y = \alpha^x \bmod n$, computes the value $W = R^{-x} \bmod n$.

2. Then he computes the message $M$ : $M = CW \bmod n$. The decryption correctness proof is as follows:

$\{CW \equiv MZR^{-x} \equiv My^u(\alpha^u)^{-x} \equiv M\alpha^{xu}\alpha^{-ux} \equiv M \bmod n\} \Rightarrow$
$\Rightarrow CW \bmod n = M.$

## 5 The Digital Signature Protocol

Suppose some user has registered his public key $(n, \alpha, y)$ in the authority center. The value $y$ is computed as follows $y = \alpha^x \bmod n$, where $x$ is the 256-bit secret key; $n = pq$, and the value $\alpha$ has 256-bit prime order $\gamma$ modulo $n$ ($\gamma$ is not secret). The user can sign the message $M$ using the following digital signature scheme, that is like the Schnorr's signature algorithm [9].

1. Generate a random 256-bit number $k$ and compute the value $R = \alpha^k \bmod n$.

2. Using some specified 256-bit hash function $F_H$ compute the first element of the signature: $E = F_H(R||M)$.

3. Compute the second element of the signature $(E, S)$ : $S = k + xE \bmod \gamma$.

The signature verification procedure includes the following steps:

1. Compute the values $\tilde{R} = y^{-E}\alpha^S \bmod n$ and $\tilde{E} = F_H\left(\tilde{R}||M\right)$.

2. Compare the values $E$ and $\tilde{E}$. If $E = \tilde{E}$, then the signature to message $M$ is valid, otherwise the signature is rejected as the false one.

The signature scheme correctness proof is as follows:

$$\left\{\tilde{R} \equiv y^{-E}\alpha^S \equiv \alpha^{-xE}\alpha^{k+xE} \equiv \alpha^k \equiv R \bmod n\right\} \Rightarrow$$
$$\Rightarrow \left\{\tilde{R} = R;\ \tilde{E} = F_H\left(\tilde{R}||M\right);\ E = F_H\left(R||M\right)\right\} \Rightarrow \tilde{E} = E\,.$$

## 6 The Blind Signature Protocol

Blind signature protocol is characterized by the fact that the person signing some electronic message $M$ prepared by some other person (called requester) doesn't know the content of the message $M$ and later after the signature generation the signer can't link the signed

message to its author. The signature scheme described in section 5 can be extended to the following blind signature protocol.

1. The signer generates a random 256-bit number $k$ and value $\bar{R} = \alpha^k \bmod n$, then he sends the value $\overline{R}$ to the requester.

2. The requester generates two 256-bit random values $\epsilon$ and $\mu$ called the "blinding" parameters and calculates the value $R = \bar{R} y^{\mu} \alpha^{\varepsilon} \bmod n$. Then, using some specified 256-bit hash function $F_H$, he calculates $H = F_H(M)$ and $E = F_H(R||M)$ and $\bar{E} = E + \mu \bmod \gamma$. The value $\overline{E}$ is the first element of the blind signature. The requester sends $\overline{E}$ to the signer.

3. The signer computes the second element of the blind signature: $\overline{S} = (k + x\overline{E}) \bmod \gamma$, then he sends the value $\overline{S}$ to the requester.

4. The requester "unblinds" parameter $\overline{S}$, i.e. computes the second element of the signature $(E, S)$ to message $M : S = \overline{S} + \epsilon \bmod \gamma$.

The signature verification is performed as it is described in Section 5.

The blind signature protocol correctness proof is as follows:

$$\left.\begin{array}{l} \tilde{R} \equiv y^{-E}\alpha^{S} \equiv \alpha^{-xE}\alpha^{\bar{S}+\varepsilon} \equiv \alpha^{-x(\bar{E}-\mu)}\alpha^{k+x\bar{E}+\varepsilon} \equiv (\alpha^{x})^{\mu}\alpha^{k}\alpha^{\varepsilon} \equiv \\ \equiv \bar{R}y^{\mu}\alpha^{\varepsilon} \equiv R \bmod n \end{array}\right\} \Rightarrow$$
$$\Rightarrow \ \left\{\tilde{R} = R;\ \tilde{E} = F_H\left(\tilde{R}||M\right)\ ;\ E = F_H(R||M)\right\} \ \Rightarrow \ \tilde{E} = E\ .$$

## 7 Discussion and conclusion

Earlier the signature and blind signature protocols based on FP and DLpP have been proposed in papers [3], [4], [5]. Comparing these signature protocols with the signature scheme proposed in this paper one can find that the proposed scheme provides significantly shorter signatures (512 bit against at least 2500 bits for the case of 128-bit security) and significantly faster generation and verification procedures. Detailed estimation of the hardware implementation cost, like the one, performed in [10], represent interest for an additional research. One can expect that the indicated advantages should yield more efficient implementation in hardware though.

Except the blind signature protocol, in all described cryptoschemes it is possible to use the value $\alpha$ with the non-secret 256-bit prime value $\gamma$ (such that $\gamma|p-1$ and $\gamma|q-1$) or with secret $\gamma=\gamma'\gamma''$, where the numbers $\gamma'$ and $\gamma''$ are some 128-bit primes (such that $\gamma'|p-1; \gamma''|q-1; \gamma'$ does not divide $q-1$; and $\gamma''$ does not divide $p-1$). In the last case one can easily prove that the algorithm solving the DLnP can be used to solve both the factoring $n$ problem and the DL modulo $p$ problem. Indeed, using such hypothetic algorithm we can perform the following computations:

1. Select a random 300-bit number $X$ and compute the value $y=\alpha^X \bmod n$.

2. Using the hypothetic algorithm for solving the DLnP compute the value $x$ such that $y=\alpha^x \bmod n$.

3. Factor the value $X-x$ (this is the sufficiently easy computational procedure, since the size of the largest factors $\gamma'$ and $\gamma''$ is equal to 128 bits) and find the values $\gamma'$ and $\gamma''$.

4. Compute the value $D=\alpha^{\gamma'} \bmod n$ (note that the value $p$ divides the value $D-1$, since $D\equiv\alpha^{\gamma'}\equiv 1 \bmod p$).

5. Using the Euclidian algorithm compute the greatest common divisor $\gcd(n, D-1)=p$ (i.e. the factorization of the value $n$ is computed).

Suppose one should find the value $u$ from the given values $\beta$ and $z$, where $z=\beta^u \bmod p$ and $\beta$ has order $\gamma'$ modulo $p$. The hypothetic algorithm for finding the DL modulo $n$ can be used to compute the value $u$ as follows:

1. Generate a value $\lambda$ having prime order $\gamma''$ modulo $q$.

2. Compute the integer $\alpha' < n$ from the following system of two congruences

$$\begin{cases} \alpha' \equiv \beta \bmod p \\ \alpha' \equiv \lambda \bmod q \end{cases}$$

.

3. Compute the integer $y < n$ from the following system of two congruences

$$\begin{cases} y \equiv z \bmod p \\ y \equiv \lambda \bmod q \end{cases}$$

.

4. Applying the hypothetic algorithm compute the unknown $x$ from the equation $y = \alpha'^x \bmod n$.

5. Output the value $u = x \bmod \gamma'$.

It is sufficiently evident that the values $\alpha'$ and $y$ computed at steps 2 and 3, correspondingly, have order $\gamma'\gamma''$. From the equation $y = \alpha'^x \bmod n$ one can get

$$y \equiv \alpha'^x \bmod p \;\Rightarrow (y \bmod p) \equiv (\alpha' \bmod p)^x \bmod p \;\Rightarrow$$
$$\Rightarrow z = \beta^x \bmod p \;\Rightarrow\; u \equiv x \bmod \gamma' \,.$$

This means the last described algorithm outputs the correct value $u$.

It has been demonstrated that for the case of the value $\alpha$ having secret order $\gamma = \gamma'\gamma''$, in certain sense, solving the DLnP is equivalent to solving both the DLpP and FP. Analogous proof for the case of the value $\alpha$ having non-secret prime order $\gamma$ represents an open problem.

To reduce the signature size (from 512 bits to 384 bits in the case of 128-bit security) in the schemes based on difficulty of simultaneous solving the FP and the DLpP one can apply the signature schemes based on the difficulty of FP and proposed earlier in paper [11]. For this purpose in the design of the signature and blind signature protocols described in [11] only small modification is required, which consists in using the composite modulus described in this paper instead of the composite modulus equal to the product of two primes having almost the same size.

Thus, the present paper shows that using the difficulty of the DLnP is an alternative and efficient approach to the design of the cryptoschemes based on difficulty of solving simultaneously both the FP and the DLpP.

## References

[1] J. Pieprzyk, Th. Hardjono, J. Seberry. *Fundumentals of Computer Security*, Springer-Verlag Berlin, 2003, 677 p.

[2] A.J. Menezes, S.A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1996, 780 p.

[3] N.H. Minh, D.V. Binh, N.T. Giang, N.A. Moldovyan. *Blind Signature Protocol Based on Difficulty of Simultaneous Solving Two Difficult Problems*, Applied Mathematical Sciences. 2012. V.6. No 139. pp. 6903–6910.

[4] N.M.F. Tahat, S.M.A. Shatnawi, E.S. Ismail. *A New Partially Blind Signature Based on Factoring and Discrete Logarithms*, J.of Mathematics and Statistics, 4(2), pp. 124–129 (2008).

[5] N.M.F. Tahat, E.S. Ismai, R.R. Ahmad. *A New Blind Signature Scheme Based On Factoring and Discrete Logarithms*, International Journal of Cryptology Research 1 (1): pp. 1–9, (2009).

[6] J. Gordon. *Strong primes are easy to find*, Advances in cryptology – EUROCRYPT'84, Springer-Verlag LNCS, 1985, vol. 209, pp. 216–223.

[7] A.A. Moldovyan, D.N. Moldovyan, L.V. Gortinskaya. *Cryptoschemes based on new signature formation mechanism*, Computer Science Journal of Moldova. 2006. Vol. 14. No 3(42). pp.397–411.

[8] T. ElGamal. *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory. 1985, Vol. IT-31, No. 4. pp. 469–472.

[9] C.P. Schnorr. *Efficient signature generation by smart cards*, Journal of Cryptology. 1991. Vol. 4. pp. 161–174.

[10] N. Sklavos. *On the Hardware Implementation Cost of Crypto-Processors Architectures* , Information Systems Security. 2010. Vol. 19. No 2. pp. 53–60.

[11] A.A. Moldovyan, N.A. Moldovyan, E.S. Novikova. *Blind 384-bit Digital Signature Scheme*, Proceedings of the International workshop, Methods, Models and Architectures for Network Security

MMM ACNS 2012. October 17-20, St.Petersburg, Russia / Lect. Notes Comput. Sci., Berlin: Springer-Verlag, 2012, Vol. 7531, pp. 77–83.

A.N. Berezin, N.A. Moldovyan,
V.A. Shcherbacov

Received April 5, 2013

St. Petersburg Institute for Informatics and Automation of
Russian Academy of Sciences
14 Liniya, 39, 199178, St. Petersburg, Russia
E–mail: *a.n.berezin.ru@gmail.com*, *nmold@mail.ru*

Institute of Mathematics and Computer Science of
Academy of Sciences of Moldova
Academiei, 5, MD-2028, Chisinau, Moldova
e-mail: *scerb@math.md*