BULETINUL ACADEMIEI DE ŞTIINŢE
A REPUBLICII MOLDOVA. MATEMATICA
Numbers 2(72)−3(73), 2013, Pages 27–36
ISSN 1024–7696

Short signatures from the difficulty of factoring problem

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

Abstract. For some practical applications there is a need of digital signature schemes (DSSes) with short signatures. The paper presents some new DSSes based on the difficulty of the factorization problem, the signature size of them being equal to 160 bits. The signature size is significantly reduced against the known DSS. The proposed DSSes are based on the multilevel exponentiation procedures. Three type of the exponentiation operations are used in the DSSes characterized in performing multiplication modulo different large numbers. As modulus prime and composite numbers are used. The latters are difficult for factoring and have relation with the prime modulus.

Mathematics subject classification: 11T71, 94A60.
Keywords and phrases: Information authentication, digital signature scheme, short signature, factorization problem, discrete logarithm problem.

1 Introduction

There are digital signature schemes (DSSes) which are based on different hard mathematical problems. The difficulty of factorizing a composite number $n = qr$, which is the product of two large unknown primes q and r, is used in the design of the first DSSes which gained practical importance, the RSA cryptosystem [1], Rabin's DSS [2], and Fiat-Shamir's DSS [3]. In the mentioned DSS the digital signature length depends on the required security level estimated as the number of group operations that should be performed to forge a signature, while using sufficiently large but reasonable memory (for example up to 2^{50} bits). At present the 2^{80} operations security level can be accepted as the minimum one. The RSA and Rabin's DSS provide the minimum security level with the 1024-bit signature length [4].

An important practical problem is the develop DSS with short signature length [5]. A DSS based on the difficulty of discrete logarithm problem in the multiplicative group [6] or in group of elliptic curve points allows the reduce the signature size [7]. The DSA standard and Schnorr's DSS based on difficulty of finding discrete logarithm modulo large prime number provide comparatively short signatures which have the 320-bit length and the security level of the 1024-bit RSA. The ECDSA standard also requires the use of the 320-bit signature size [8] to get the same security.

In the present paper we consider some ways to reduce the signature length in DSS based on the difficulty of the factorization problem. In Section 2 we present

DSSes with 320-bit and 240-bit signature lengths proposed in our previous works [9, 10] using the factoring problem difficulty. In Section 3 we describe some new signature formation mechanisms which are used to reduce the signature length to the 160 bits value and present two DSSes with short signature. Section 4 presents the comparison with the known signature algorithms. Section 5 concludes the paper.

2 Randomized signature schemes based on factorization problem

The well known cryptosystem RSA [1] is based on the calculations modulo n which is the product of two randomly chosen strong prime numbers r and q [11]. The public key is represented by a pair of numbers $(n,\ e)$, where e is a random number that is relatively prime with the Euler phi function $\varphi(n) = (p-1)(q-1)$. The triple $(p,\ q,\ d)$, where $d = e^{-1} \mod \varphi(n)$, is the private key. Data ciphering with RSA is described as follows: $C = M^e \mod n$ (public-key encryption) and $M = C^d \mod n$ (decryption), where $M < n$ is a plaintext and C is a ciphertext. The RSA signature (S) generation and verification are performed as follows: $S = M^d \mod n$ and $M = S^e \mod n$, correspondingly.

Usually the signed documents are comparatively long. In such cases instead of sign the document M we sign the hash function value $H = F_H(M)$ which corresponds to $M : S = H^d \mod n$. The RSA security is based on the difficulty of factoring modulus n, which depends on the structure of primes p and q. At present the requirements on the primes p and q are well clarified [4, 12, 14]. The RSA signature size is sufficiently large, 1024 bit in the case of 80-bit security. The factoring problem is well studied, therefore it is a trusted one for designing secure DSSes. However, other DSSes based on this problem, for example Rabin's DSS [2] and Fiat-Shamir's DSS [3], also define long signatures. Using the fact that computing discrete logarithm modulo n is at least as difficult as factoring n the papers [9, 10] propose the DSS with the 320-bit randomized signature. The paper [11] proposes the randomized DSS with the 240-bit signature.

2.1 The 320-bit signature algorithm

In the DSS from [9] such strong primes r and q are used that the numbers $p-1$ and $q-1$ contain different prime divisors γ' and γ'', respectively. The values γ' and γ'' should have the lengths at least equal to 80 bits. The secret key is the triple $(p,\ q,\ \gamma)$. The public key is a pair of numbers $(n,\ \alpha)$, where α is generated as follows. Select a random number β that is simultaneously a primitive element modulo p and a primitive element modulo q, compute $t = \gamma'^{-1}\ \gamma''^{-1}\varphi(n) = \gamma'^{-1}\ \gamma''^{-1}(p-1)(q-1)$ and $\alpha = \beta^t \mod n$. The number α is generator of the γ-order group $\{\alpha \mod n,\ \alpha^2 \mod n, \ldots, \alpha^\gamma \mod n\}$, i. e. $\alpha^\gamma \mod n = 1$.

The generation of the α parameter can be performed also in the following way:

1. Choose random $\beta < n$ and calculate $\sigma = \beta^t \mod n$.

2. If $\sigma \neq 1$ and $gcd(\sigma - 1,\ n) = 1$, then $\alpha \leftarrow \sigma$, otherwise go to step 1.

A document or the hash value corresponding to it are interpreted as integers M and H, correspondingly.

Therefore, the required g-bit sequence $H = (h_{g-1},\ h_{g-2},\ \ldots,\ h_1,\ h_0)$ is taken as the number

$$H = h_{g-1}2^{g-1} + h_{g-2}2^{g-2} + \ldots + h_2 2^2 + h_1 2^1 + h_0 2^0.$$

This scheme is described by the following verification equation:

$$g + k = (\alpha^{kgH} \mod n) \mod \delta,$$

where $(g,\ k)$ is the signature and $\delta > \gamma$ is a prime number which has, for example, the length of $|\delta| = |\gamma| + 4$ bits, where $|\delta|$ denotes the bit size of the value δ. The signature size is $|k| + |g| = |\delta| + |\gamma| \approx 2|\gamma| = 2(|\gamma'| + |\gamma''|) \geq 320$ bits.

The signature generation is performed as follows:

1. Given the M document calculate the hash value $H = F_H(M)$.

2. Check whether $H \neq 0$ and $\gcd(H,\ \gamma) = 1$. If H = 0 or $\gcd(H,\ \gamma) \neq 1$, then modify the document M and go back to step 1.

3. Select a random $U < \gamma$ and calculate $Z = (\alpha^U \mod n) \mod \delta$ and $D = (Z^2/4 - U/H) \mod \gamma$.

4. Check whether D is a quadratic residue modulo γ. If not, then go back to step 3.

5. Solve the following system which contains one congruence and one equation relative to the unknowns g and k:

$$\begin{cases} kgH \equiv U \mod \gamma, \\ g + k = Z. \end{cases}$$

The solution gives the following signature generation formulas:

6. Calculate the signature using the formulas $g = Z/2 \pm \sqrt{D} \mod \gamma$ and $k = Z - g$.

The signature verification is performed as follows:

1. Calculate the hash function $H = F_H(M)$.

2. Check whether the following signature verification equation

$$g + k = (\alpha^{kgH} \mod n) \mod \delta$$

is satisfied. If $g + k \neq (\alpha^{kgH} \mod n) \mod \delta$, then reject the signature.

Proof that the signature verification works:

The left side of the signature verification equation is equal to:

$$g + k = Z = (\alpha^U \mod n) \mod \delta.$$

The right side of the signature verification equation is equal to:

$$(\alpha^{kgH} \mod n) \mod \delta = (\alpha^U \mod n) \mod \delta = g + k,$$

since we have

$$kgH \equiv (Z - Z/2 \mp sqrtD)(Z/2 \pm \sqrt{D})H \equiv (Z^2/4 - D)H \equiv$$
$$\left[Z^2/4 - Z^2/4 - U/H\right] H \equiv U \mod \gamma.$$

To explain the requirements imposed on parameters n and α it is useful to consider the case of prime value γ (for example: $\gamma \mid p - 1$ and $\gamma \nmid q - 1$), for which we have

$$\alpha = \beta^{\varphi(n)/\gamma} = (\beta^{(q-1)})^{(p-1)/\gamma} \mod n \Rightarrow$$
$$\alpha \equiv (\beta^{(q-1)})^{(p-1)/\gamma} \equiv 1^{(p-1)/\gamma} \equiv 1 \mod q \Rightarrow$$
$$\alpha - 1 \equiv \mod q \Rightarrow q \mid \alpha - 1 \Rightarrow gcd(\alpha - 1, n) = q$$

where $\gcd(a, b)$ denotes the greatest common divisor of the numbers a and b. Thus, in the considered case it is possible to factorize the modulus using the extended Euclidean algorithm. Therefore some restrictions imposed on generating the public key are necessary. We can prevent this attack using a prime γ that divides both $p - 1$ and $q - 1$, but γ^2 does not divide $p - 1$ nor $q - 1$. In this case we have:

$$\alpha \equiv \beta^{\frac{(p-1)(q-1)}{\gamma^2}} \equiv \beta^{u'u''}(\mod n),$$

where γ does not divide each of the numbers $u' = (p - 1)/\gamma$ and $u'' = (q - 1)/\gamma$. If β is simultaneously a primitive element modulo p and a primitive element modulo q, then we have $\alpha \mod p \neq 1$ and $\alpha \mod q \neq 1$, i. e. $gcd(\alpha - 1, n) = 1$. Unfortunately, in the case of the prime secret element γ can be calculated factorizing the $n - 1$ value. Indeed, we have: $p = u'\gamma + 1$, $q = u''\gamma + 1$, and $n = u'u''\gamma^2 + (u' + u'')\gamma + 1$, hence $\gamma \mid (n - 1)$. Therefore the composite value $\gamma = \gamma'\gamma''$, where γ' and $\gamma' \neq \gamma''$ are different divisors of $p - 1$ and $q - 1$, should be used. If β is a "double primitive element", then we have

$$\alpha \equiv \beta^{\frac{(p-1)(q-1)}{\gamma'\gamma''}} \equiv \beta^{u'u''}(\mod n),$$

where $u' = (p - 1)/\gamma'$ and $u'' = (q - 1)/\gamma''$. Thus, in such way of the public key formation we also have $gcd(\alpha - 1, n) = 1$. If one of the primes γ' and γ'', for example γ', is small, then one can factorize n trying different values γ' and verifying the relation $\gcd(\alpha^{\gamma'} - 1, n) = p$. Therefore both values γ' and γ'' should be sufficiently large. To define 80-bit security one has to use the 80-bit prime numbers γ' and γ''.

2.2 The 240-bit signature algorithm

The paper [11] proposes some modification of the DSS described in Subsection 2.1. The main feature of the DSS introduced in [11] is to apply the "two-level" exponentiation procedure that provides the possibility to use a prime secret value γ. In the DSS from [11] the value γ is the order of the value α modulo the secret

value q. Due to hiding the modulus q it becomes possible to use prime secret order γ. In this DSS the following verification equations are used:

$$R \equiv \beta^{\alpha^{kgH} \mod n} \mod p;\ k = (R^{\alpha^g \mod n} \mod p) \mod \delta,$$

where $p = 2n + 1$ is a prime; n is the product of two 512-bit primes q and r ($n = rq$); β is a number which has the order q modulo p; and α is a number which has the order γ modulo q. The modulus δ is a 80-bit prime. The private key is represented by the pair (q, γ), where γ is the 160-bit prime number such that $\gamma \,|\, q - 1$. The public key is the triple (α, β, p). The 240-bit signature (k, g), where $|k| = 80$ bits and $|g| = 160$ bits, corresponds to the 160-bit hash-function value H and provides the 80-bit security.

The signature generation procedure includes the following steps:

1. Generate a random value $U < \gamma$.
2. Compute the value k using the formula $k = (\beta^{\alpha^U \mod q} \mod p) \mod \delta$.
3. Compute the value g using the formula $g = U/(kH + 1) \mod \gamma$.

The last formula is derived from the following system of two congruences:

$$\begin{cases} t + g = U \mod \gamma \\ t = kgH \mod \gamma, \end{cases}$$

where t is an auxiliary unknown.

Now prove that the signature verification works. Suppose a valid signature (k, g) corresponding to the hash value H is given. Taking into account that α has the order γ modulo q and substituting the values k and g in the verification equations we get

$$R \equiv \beta^{\alpha^{kgH} \mod n} \equiv \beta^{\alpha^{kgH} \mod q} \equiv \beta^{\alpha^{\left(\frac{kHU}{kH+1}\right)} \mod q} \mod p;$$

$$\left(R^{\alpha^g \mod n} \mod p\right) \mod \delta = \left(R^{\alpha^g \mod q} \mod p\right) \mod \delta =$$

$$= \left(\left(\beta^{\alpha^{\left(\frac{kHU}{kH+1}\right)} \mod q} \mod p\right)^{\alpha^g \mod q} \mod p\right) \mod \delta =$$

$$= \left(\beta^{\alpha^{\frac{kHU}{kH+1} + \frac{U}{kH+1}} \mod q} \mod p\right) \mod \delta = \left(\beta^{\alpha^U \mod q} \mod p\right) \mod \delta = k,$$

i. e. the signature verification result is positive, which means the DSS works correctly.

Let us consider some possible attacks. The first one includes finding the value $X = \log_\beta R \mod p$ and calculating q as a divisor of the value $\left(\alpha^{kgH} \mod n\right) - X$. Then the value γ can be determined as one of divisors of the value $q - 1$. Due to the large values $|p|$ and $|q|$ this attack is computationally infeasible.

The second attack is to find the value $X' = \log_\alpha \alpha^{kgH} \mod n$ and then to calculate γ as one of divisors of the value $kgH - X'$. The second attack defines the following requirement: the value α should have a large order λ modulo n. Taking into account the comments for selection of the value α in Subsection 2.1 the value λ should be equal to the product of two large primes: $\lambda = \gamma\mu$, where μ divides

$p - 1$ and does not divide $q - 1$; $|\mu| \geq 160$ bits. If this requirement is satisfied, then the second attack is also computationally infeasible.

The most efficient is the third attack implementing a modification of the Baby-Step-Giant-Step algorithm to compute the value x from the known value $y = \beta^{\alpha^x \bmod n} \bmod p$.

The algorithm is described as follows [10]:

1. Select a random value $U > 2^{|\gamma|+10}$ and calculate $y = \beta^{\alpha^U \bmod n} \bmod p$.
2. For $i = 0$ to $D = \left[\sqrt{\gamma}\right] + 1$ calculate $z' = \beta^{\alpha^{iD} \bmod n} \bmod p$. Save the values z' in the table containing pairs $z'(i)$.
3. Order the table of pairs $(i,\ z'(i))$ according to the value $z'(i)$ and set $j = 0$.
4. Calculate $z''(j) = y^{1/\alpha^j \bmod n} \bmod p$.
5. Check if in the table there exists $z'(i_0)$ such that $z'(i_0) = z''(j)$. If $z''(j) \neq z'(i)$ for $i = 0$ to D, then increment the counter $j : j = j + 1$ and go to step 4.
6. Calculate the value $U' = i_0 + j$ and factorize the value $U - U'$.
7. Select a divisor γ such that $\beta^{\alpha^\gamma \bmod n} \bmod p = \beta$.

The difficulty of this algorithm is equal to $W \approx 3\sqrt{\gamma}$ exponentiation operations. To provide the 80-bit security one should use the prime order γ which has the size equal to 160-bits.

3 Proposed 160-bit signature schemes

The proposed DSS is based on the three-level exponentiation procedure that defines the following function

$$y = \Omega^{\beta^{\alpha^x} \bmod n \bmod N} \bmod p$$

where $p = eN + 1$; $N = PQ$; $P = e'n + 1$; $n = qr$; e is a 16-bit even integer; e' is a 100-bit even integer; $Q = 2Q' + 1$ is a prime; $r = 2r' + 1$ is a prime; $q = e''\gamma + 1$ is a prime; Q', q, and r' are 512-bit primes; γ is a 80-bit prime; P is a 1124-bit prime. The value Ω has order P modulo p; the value β has order q modulo P; the value α has order γ modulo q. The values P, Q, r, and γ are elements of the private key.

The three-level exponentiation procedure makes the Baby-Step-Giant-Step algorithm inefficient to compute the value x that defines the given value y. Indeed, we have

$$\begin{aligned} y &\equiv \Omega^{\beta^{\alpha^x} \bmod n \bmod N} \bmod p \equiv \\ &\equiv \Omega^{\beta^{\alpha^{iD+j}} \bmod n \bmod N} \bmod p \equiv \\ &\equiv \Omega^{\beta^{\alpha^{iD}\alpha^j} \bmod n \bmod N} \bmod p, \end{aligned}$$

i. e. it is not possible to transform the formula defining the function $y(x)$ in the formula the right side of which is free from the integer j and the left side is free from the integer i. Below we propose two variants of the 160-bit signature scheme using an additional 80-bit prime δ as a specified parameter of the signature algorithm.

3.1 The first scheme

The public key includes the values p, N, n, Ω, β, and α.

The modulus p is generated as follows:

1. Generate 512-bit primes Q, q, and r.

2. Generate a random 100-bit even integer e' such that the value $P = e'rq + 1$ is prime.

3. Select such a 16-bit even integer e that the value $p = ePQ + 1$ is prime.

The value Ω is generated as follows:

1. Generate a random number $\rho < p$ and compute $\Omega' = \rho^{eQ} \mod p$.

2. If $\Omega' \neq 1$, then output $\Omega = \Omega'$.

The value β is generated as follows:

1. Generate a random number $\rho < N$ and compute $\beta' = \rho^{e'r} \mod N$.

2. If β' is a primitive element modulo Q and $\beta' \neq 1 \mod q$, then output $\beta = \beta'$.

The value α is generated as follows:

1. Generate a random number $\rho < n$ and compute $\alpha' = \rho^{e''} \mod n$.

2. If α' is a primitive element modulo r and $\alpha' \neq 1 \mod q$, then output $\alpha = \alpha'$.

The first variant of the 160-bit DSS includes the following signature generation procedure.

1. Compute the hash function value H from the message M to be signed $H = F_H(M)$.

2. Generate a random value $t < \gamma$ and compute the value R:

$$R = \Omega^{\beta^{H\alpha^t} \mod q \mod P} \mod p.$$

3. Compute the first signature element k: $k = RH \mod \delta$.

4. Compute the second signature element g: $g = k^{-1}t \mod \gamma$.

The corresponding signature verification procedure is as follows:

1. Compute the hash function value $H = F_H(M)$ from the message M to which the signature $(k,\ g)$ is appended.

2. Compute the value $\tilde{R}$: $\tilde{R} = \Omega^{\beta^{H\alpha^{kg} \mod n} \mod N} \mod p$.

3. Compute the value $\tilde{k}$: $\tilde{k} = \tilde{R}H \mod \delta$.

If $\tilde{k} = k$ the signature is valid. Otherwise the signature is rejected.

Correctness proof.

$$\begin{aligned}
&\tilde{R} = \Omega^{\beta^{H\alpha^{kg} \mod n} \mod N} \mod p = \\
&= \Omega^{\beta^{H\alpha^{kg} \mod q} \mod P} \mod p = \\
&= \Omega^{\beta^{H\alpha^{t} \mod q} \mod P} \mod p = R \Longrightarrow \\
&\Longrightarrow \tilde{k} = \tilde{R}H \mod \delta = RH \mod \delta = k.
\end{aligned}$$

3.2 The second scheme

The second variant of the 160-bit signature scheme uses an additional element of the public key $y = \alpha^x \mod q$, where x is the additional 80-bit element of the private key, and includes the following signature generation procedure.

1. Compute the hash function value H from the message M to be signed $H = F_H(M)$.

2. Generate a random value $t < \gamma$ and compute the value R:

$$R = \Omega^{H\beta^{\alpha^t \mod q} \mod P} \mod p.$$

3. Compute the first signature element k: $k = RH \mod \delta$.

4. Compute the second signature element g: $g = t - xk \mod \gamma$.

The corresponding signature verification procedure is as follows:

1. Compute the hash function value $H = F_H(M)$ from the message M to which the signature $(k,\ g)$ is appended.

2. Compute the value $\tilde{R}$: $\tilde{R} = \Omega^{H\beta y^k \alpha^g \mod n \mod N} \mod p$.

3. Compute the value $\tilde{k}$: $\tilde{k} = \tilde{R}H \mod \delta$.

If $\tilde{k} = k$, the signature is valid. Otherwise the signature is rejected.

Correctness proof.

$$\begin{aligned}
&\tilde{R} = \Omega^{H\beta y^k \alpha^g \mod n \mod N} \mod p = \\
&= \Omega^{H\beta y^k \alpha^g \mod q \mod P} \mod p = \\
&= \Omega^{H\beta \alpha^{xk} \alpha^g \mod q \mod P} \mod p = \\
&= \Omega^{H\beta \alpha^{xk+t-xk} \mod q \mod P} \mod p = \\
&= \Omega^{H\beta \alpha^t \mod q \mod P} \mod p = R \Longrightarrow \\
&\Longrightarrow \tilde{k} = \tilde{R}H \mod \delta = RH \mod \delta = k.
\end{aligned}$$

4 Comparison with the known signature algorithms

The attacks against the DSS [10] which are described in Section 2 can be also considered as attacks against the proposed in this section signature algorithm, which are based on the three-level exponentiation procedure. Actually due to one added exponentiation level it becomes impossible to apply the Baby-Step-Giant-Step algorithm and it becomes possible to reduce the size of the order of the value α, which leads to shortening the digital size to 160 bits. Table 1 illustrates the comparison of some signature schemes related to Baby-Step-Giant-Step algorithm.

Table 1. Transformation of the base function using the representation of the unknown x as $x = iD + j$ where $D = \left[(\sqrt{\gamma})\right] + 1,\ i,\ j = 0, 1, 2, \ldots, D$; γ is order of the value α modulo prime p.

DSS	*Base function of DSS*	*Representation of the base formula*
[7]	$y = \alpha^x \mod p$	$y\alpha^{-iD} = \alpha^j \mod p$
[10]	$y = \beta^{\alpha^x \mod n} \mod p$	$y^{\alpha^{-iD} \mod n} = = \beta^{\alpha^j \mod n} \mod p$
Proposed	$y = \Omega^{\beta^{\alpha^x \mod n} \mod N} \mod p$	?

The proposed DSSes are oriented to applications that require using short signatures and the performance of the signature generation and verification procedures is not of high significance. In the described signature generation and verification procedures more exponentiation operations are used than in the known DSSes with 240-bit and 320-bit signatures. For comparison see Table 2.

Table 2. The performance comparison of the proposed signature scheme with the DSSes of [7] and [10] for the case of the 80-bit security.

Signature properties	DSS			
	The 1st proposed	The 2nd proposed	[10]	[7]
Signature generation performance, arbitrary units	5	5	30	100
Signature verification performance, arb. un.	2	2	11	50
Signature size, bits	160	160	240	320

5 Conclusion

This paper introduces an approach to design 160-bit signature schemes based on the difficulty of factorization problem. Different variants of implementing the approach are possible applying the proposed three-level exponentiation procedure. We estimate that the signature generation and verification performance can be increased by factor ≈ 3, however such implementations of the 160-bit signature algorithms represent an additional problem.

References

[1] Rivest R. L., Shamir A., Adleman L. M. *A Method for Obtaining Digital Signatures and Public Key Cryptosystems.* Communications of the ACM, 1978, **21**, No. 2, 120–126.

[2] Rabin M. O. *Digitalized signatures and public key functions as intractable as factorization.* Technical report MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[3] Fiat A., Shamir A. *How to prove yourself: Practical solutions to identification and signature problems.* Advances in cryptology – CRYPTO'86, Springer-Verlag LNCS, 1987, vol. 263, 186–194.

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

[5] Pintsov L., Vanstone S. *Postal revenue collection in the digital age.* Proceedings of Financial Cryptography (ed. by Y. Frankel), Springer-Verlag LNCS, 2000, vol. 1962, 105–120.

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

[7] Schnorr C. P. *Efficient signature generation by smart cards.* J. Cryptology. 1991, **4,** 161–174.

[8] ANSI X9.62 and FIPS 186-2. Elliptic curve signature algorithm, 1998.

[9] Moldovyan N. A. *New Public Key Cryptosystems Based on Difficulty of Factorization and Discrete Logarithm problems.* 3d Int. Workshop IF&GIS'07 Proc. St.Petersburg, May 28–29, 2007, Springer LNGC, 2007, vol. XIV, 160–172.

[10] Moldovyan N. A. *Short Signatures from Difficulty of Factorization Problem.* International Journal of Network Security, 2009, **8,** No. 1, 90–95 (http://ijns.femto.com.tw).

[11] Moldovyan N. A. *An approach to shorten digital signature length.* Computer Science Journal of Moldova, 2006, **14,** No. 3(42), 390–396.

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

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

Received November 29, 2012

N. A. Moldovyan, A. A. Moldovyan
St. Petersburg Institute for Informatics
and Automation of Russian Academy of Sciences
14 Liniya, 39, St. Petersburg 199178
Russia
E-mail: *nmold@mail.ru*; Web: *www.spiiras.nw.ru*

V. A. Shcherbacov
Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
Academiei str. 5, MD−2028 Chişinău
Moldova
E-mail: *scerb@math.md*; Web: *www.scerb.com*