A Secure Proxy Signature Scheme with Delegation by Warrant

Constantin Popescu

Department of Mathematics and Computer Science, University of Oradea,
Oradea 410087, Romania,
cpopescu@uoradea.ro

Abstract: A proxy signature scheme is a variation of the ordinary digital signature schemes which enables a proxy signer to generate signatures on behalf of an original signer. In this paper, we present a secure proxy signature scheme. Our proxy signature scheme is based on the difficulty of solving the discrete logarithm problem. We prove that our proxy signature scheme meets all the security requirements for a proxy signature scheme.

Keywords: Cryptography, proxy signature, proxy signer, delegation, discrete logarithm problem, warrant.

1. Introduction

The concept of the proxy signature scheme was first introduced by Mambo et al. [9] in 1996. Their proxy signature scheme allows the original signer to delegate his/her signing right to the proxy signer to sign a message on behalf of the original signer. Afterwards, a verifier, which knows the public keys of the original signer and the proxy signer, can verify the validity of the proxy signature issued by the proxy signer.

The proxy signature scheme is classified in two criteria [9]: the delegation technique and generating the proxy signature. There are three types in the first criterion: full delegation, partial delegation and delegation by warrant. In a full delegation proxy signature scheme, a proxy signer uses the same private key as the original signer and generates a proxy signature as the original signer does. The disadvantage of the full delegation comes from the difficulty of distinguishing between the original signer and the proxy signer. In the partial delegation proxy signature scheme, an original signer derives a proxy key from his private key and sends it to a proxy signer in a secure channel. In a proxy signature scheme with delegation by warrant, the original signer gives a proxy signer a special message, namely, warrant. A warrant certifies that the proxy signer is legal and consists of signers' identity, delegation period and the types of the message on which the proxy signer can sign.

Also, there are two types in the second criterion: protected and unprotected proxy signature schemes. In an unprotected proxy signature scheme, the proxy signature is generated by the both the proxy signer and the original signer. In this case, the verifier cannot distinguish the identity of the signer. In a protected proxy signature scheme, the proxy signature is generated with the proxy signature key of the original signer and also with the private key of the proxy signer. Afterwards, a verifier validates the proxy signature with the public keys of both the original signer and the proxy signer.

Proxy signature schemes are useful in many applications such as electronic payment systems [3], [15], [17], [18] and wireless networks [7], [26].

A lot of proxy signature schemes and some ID-based proxy signature schemes with special features were proposed, such as identity-based multi-proxy signature [1], [2], identity-based strong designated verifier proxy signature [28], [30].

Okamoto et al. [14] proposed a proxy unprotected signature scheme based on the RSA assumption. Also, in 2001, Lee et al. [7] proposed a proxy protected signature scheme based on the RSA assumption. Unfortunately, Wang et al. [27] point out that Lee et al.'s [7] proxy signature scheme is insecure. The first proxy signature scheme based on the factoring integer problem is proposed by Shao [22], in 2003. Recently, Zhou et al. [31] proposed two efficient proxy protected signature schemes. Their first scheme is based on RSA [20] assumption and the second scheme is based on the integer factorization problem. Zhou et al. [31] claim that their schemes are more efficient than other schemes. However, Park et al. [16] point out their schemes are insecure. Moreover, Liu et al. [8] point out that Zhou et al.'s [31] schemes are vulnerable to the undelegated proxy signature attack: any attacker without the delegation of the original signer can generate a valid proxy signature. Xue et al. [29] proposed two proxy signature schemes based on the difficulty of factorings of large integers without formal security proofs. Recently, Shao [24] proposed proxy protected signature scheme based on RSA. Also, most proxy signature schemes are based on the difficulty of discrete logarithm problem [4] or elliptic curve discrete logarithm problem [6], [11], [19], [25].

Mambo et al. [9], [10] proposed three proxy signature schemes based on ElGamal's signature scheme [5], Schnorr's signature scheme [21], and Okamoto's signature scheme [13].

In 1996, Mambo, Usuda and Okamoto, first defined the basic security properties of a proxy signature scheme as follows [9], [10]:

Verifiability: From a proxy signature, a verifier can be convinced of the original signer's agreement on the signed message.

Strong unforgeability: A proxy signer can create a valid proxy signature on behalf of the original signer. However, the original signer and any third party cannot generate a valid proxy signature with the name of proxy signers.

Strong identifiability: From a proxy signature, anyone can determine the identity of the corresponding proxy signer.

Strong undeniability: Once a proxy signer generates a valid proxy signature on behalf of the original signer, the proxy signer cannot deny his signature generation against anyone.

Prevention of misuse: It should be confident that the proxy key pair cannot be used for other purposes. In the case of misuse, the responsibility of proxy signers should be determined explicitly.

In this paper we propose a secure proxy signature scheme based on the discrete logarithm problem. The proposed proxy signature scheme is derived from the Shao's signature scheme [23]. Our proxy signature scheme inherits the strength security properties of the signature scheme proposed in [23]. Also, we give an elliptic curve version of our proxy signature scheme.

The rest of this paper is organized as follows. In the next section we review the model of a proxy signature scheme. Then we present our proxy signature scheme in the section 3 and in the section 4 we give an elliptic curve version of our proposed proxy signature scheme. Furthermore, we discuss some aspects of security in the section 5. The section 6 concludes the work of our paper.

2. The Model of a Proxy Signature Scheme

In this section, we describe a formal definition for a proxy signature scheme.

A proxy signature scheme has three entities: an original signer, a proxy signer and a verifier.

Definition 1. A proxy signature scheme is comprised of four algorithms: Key Generation, Proxy Key Generation, Proxy Signature Generation, Verify:

Key Generation: Given a security parameter k as input, a random algorithm outputs the system parameters. The original signer chooses his/her key pair (SK, PK).

Proxy Key Generation: The original signer creates a warrant w, which records the delegation policy, limits of authority, valid periods of delegation and proxy signatures, and the identities of the proxy signer and the original signer. Then the original signer generates a signature of the warrant w with his/her private key and sends this signature to the proxy signer. This algorithm outputs the proxy private key s_{PS}.

Proxy Signature Generation: For a message $m \in \{0,1\}^*$, the proxy signer computes the proxy signature σ by using his/her proxy private key s_{PS}.

Verify: This is a deterministic algorithm. Given a proxy signature σ, a verifier uses his/her private key to check its validity and outputs *1* if σ is valid, otherwise outputs *0*.

3. The Proposed Proxy Signature Scheme

In this section, we present a secure proxy signature scheme derived from the Shao's signature scheme [23]. Our proxy signature scheme inherits the strength security properties of the signature scheme proposed in [23]. The participants of our proxy signature scheme are: an original signer, a proxy signer and a verifier. The details are described as follows:

3.1 Key generation

In this section the original signer creates his/her private key and the corresponding public key.

Let p and q be large prime numbers such that $q \mid (p-1)$. Let H be collision-resistant hash function where:

$$H : \{0,1\}^* \to \{0,1\}^{|q|/2} \quad (1)$$

The original signer chooses a random $x \in Z_q^*$ and computes $y = g^x \bmod p$. The original signer has a private key x and the corresponding public key is the tuple (p, q, g, y).

3.2 Proxy key generation

The proxy signer creates his/her private key and the corresponding public key. The delegation warrant w contains the delegation policy, including limits of authority, the message type to be signed, valid periods of delegation and proxy signatures, and the identities of the proxy signer and the original signer.

a) The original signer signs the delegation warrant w as follows:

The original signer chooses a random number $k_{OS} \in Z_q^*$ and computes:

$$r_{OS} = g^{k_{OS}} H(w) \bmod p$$

$$h = H(w \,\|\, r_{OS})$$

$$s_{OS} = k_{OS} - xh \bmod q\,.$$

The symbol || denotes the concatenation of two strings.

The pair (h, s_{OS}) is the signature of the delegation warrant w. The original signer sends w and the pair (h, s_{OS}) to the proxy signer.

b) The proxy signer computes:

$$r' = y^h g^{s_{OS}} H(w) \bmod p$$

$$h' = H(w \,\|\, r')$$

The proxy signer checks whether the following equation holds:

$$h = h'\,.$$

c) If the above equation holds, the proxy signer computes the proxy private key s_{PS} as follows:

The proxy signer selects a random number $k_{PS} \in Z_q^*$ and computes:

$$g_{PS} = g^{k_{PS}} \bmod p$$

$$s_{PS} = s_{OS} k_{PS}^{-1} \bmod q\,.$$

3.3 Proxy signature generation

The proxy signer generates a proxy signature of a message $m \in \{0,1\}^*$ as follows:

d) The proxy signer picks a random number $k \in Z_q^*$ and computes:

$$r = g_{PS}^k H(m \,\|\, w) \bmod p$$

$$h_{PS} = H(m \,\|\, w \,\|\, r)$$

$$s = k - s_{PS} h_{PS} \bmod q\,.$$

e) The proxy signature of the message m is the tuple $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$.

3.4 Verify

After receiving the proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$, a verifier processes as following:

a) Verifies whether the valid limits authority, the message type to be signed by the proxy signer and the public keys of the original signer and the proxy signer meet the restrictions in the delegation warrant w. If not, the verifier stops the verify algorithm.

b) Computes the following values:

$$r'' = g^{s_{OS} h_{PS}} g_{PS}^s H(m \,\|\, w) \bmod p \quad (2)$$

$$h'' = H(m \,\|\, w \,\|\, r'') \quad (3)$$

c) The verifier checks whether the following equation holds:

$$h_{PS} = h'' \quad (4)$$

d) If the above equation holds, the verifier accepts the proxy signature, otherwise, the verifier rejects the proxy signature.

4. The Elliptic Curve Version of the Proposed Scheme

In this section we propose the elliptic curve version of our proxy signature scheme.

4.1 Key generation

Assume (q,a,b,P,n,h) is a set of the elliptic curve domain parameters [11]:

- q is a field size;
- a,b are two field elements in Z_q^*, which define the equation of the elliptic curve E over Z_q^*;
- A finite point $P=(x,y)$ of prime order in $E(Z_q^*)$, where $x,y \in Z_q^*$;
- The order n of the point P, with $n > 2^{160}$ and $n > 4\sqrt{q}$;
- The cofactor $h = \# E(Z_q^*)/n$.

Let H be collision-resistant hash function where:

$$H : \{0,1\}^* \rightarrow \{0,1\}^{|q|/2}$$

The original signer chooses a random $x \in Z_q^*$ and computes the point $Q = xP$. The original signer has a private key x and the corresponding public key is the tuple (q,a,b,P,Q,n,h).

4.2 Proxy key generation

The proxy signer creates his/her private key and the corresponding public key. The original signer signs the delegation warrant w as follows:

a) Chooses a random number $k'_{OS} \in Z_q^*$ and computes:

$$R_{OS} = k'_{OS} P H(w)$$

$$h_{OS} = H(w \,||\, x_{R_{OS}})$$

$$s'_{OS} = k'_{OS} - x h_{OS} \bmod q \,,$$

where $x_{R_{OS}}$ is x-coordinate of the point R_{OS}.

The pair (h_{OS}, s'_{OS}) is the signature of the delegation warrant w. The original signer sends w and the pair (h_{OS}, s'_{OS}) to the proxy signer.

b) The proxy signer computes:

$$R' = (h_{OS} Q + s'_{OS} P) H(w)$$

$$h' = H(w \,||\, x_{R'}),$$

where $x_{R'}$ is x-coordinate of the point R'.

The proxy signer checks whether the following equation holds:

$$h' = h_{OS} \qquad (5)$$

c) If the equation (5) holds, the proxy signer calculates his/her private key s'_{PS} as follows:

- The proxy signer selects a random number $k_{PS} \in Z_q^*$ and computes:

$$Q_{PS} = k_{PS} P$$

$$s'_{PS} = s'_{OS} k_{PS}^{-1} \bmod q \,.$$

4.3 Proxy signature generation

The proxy signer generates a proxy signature of a message $m \in \{0,1\}^*$ as follows:

a) The proxy signer picks a random number $k \in Z_q^*$ and computes:

$$R = k Q_{PS} H(m \,||\, w)$$

$$h'_{PS} = H(m \,||\, w \,||\, x_R)$$

$$s' = k - s'_{PS} h'_{PS} \bmod q \,,$$

where x_R is x-coordinate of the point R.

b) The proxy signature of the message m is the tuple $(m, w, Q_{PS}, s'_{OS}, h'_{PS}, s')$.

4.4 Verify

A verifier has to verify the proxy signature $(m, w, Q_{PS}, s'_{OS}, h'_{PS}, s')$:

a) Checks whether the valid limits authority, the message type to be signed by the proxy signer and the public keys of the original signer and the proxy signer meet the restrictions in the delegation warrant w. If not, the verifier stops this algorithm.

b) Calculates the following values:

$$R'' = \left(s'_{OS}h'_{PS}P + s'Q_{PS}\right)H\left(m \,||\, w\right) \tag{6}$$

$$h'' = H\left(m \,||\, w \,||\, x_{R''}\right) \tag{7}$$

where $x_{R''}$ is x-coordinate of the point R''.

c) The verifier checks whether the following equation holds:

$h'_{PS} = h''$. (8)

If the equation (8) holds, the verifier accepts the proxy signature, otherwise, the verifier rejects it.

5. Security Analysis of our Proxy Signature Scheme

In this section we discuss aspects of security of the proposed proxy signature scheme. The hardness of forgery in our proxy signature scheme is determined by security parameters p and q. We let p be at least 512 bits and q be 160 bits. The Secure Hash Algorithm

(SHA) [12] is used in our scheme. The security of our proposed signature schemes is based on the difficulty of solving the discrete logarithm problem.

5.1 Correctness

We have to prove the correctness of the signature (h, s_{OS}) of the delegation warrant w.

Theorem 1. The pair (h, s_{OS}) is a valid signature of the delegation warrant w.

Proof. In order to prove that (h, s_{OS}) is a valid signature of the delegation warrant w, we have to check that $h = h'$, which is equivalent with $r_{OS} = r'$. We have:

$$r' = y^h g^{s_{OS}} H(w) \bmod p =$$

$$= g^{xh} g^{k_{OS}-xh} H(w) \bmod p$$

$$= g^{xh+k_{OS}-xh} H(w) \bmod p$$

$$= g^{k_{OS}} H(w) \bmod p$$

$$= r_{OS}$$

Also, we will prove the correctness of the proposed proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ of the message m.

Theorem 2. The tuple $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ is a valid proxy signature of the message *m*.

Proof. We have to prove that

$$h_{PS} = h''$$

which is equivalent with $r'' = r$. Obviously, the relation follows from:

$$r'' = g^{s_{OS}h_{PS}} g_{PS}^{s} H(m \,||\, w) \bmod p =$$

$$= g^{s_{OS}h_{PS}} g^{k_{PS}s} H(m \,||\, w) \bmod p$$

$$= g^{s_{OS}h_{PS}} g^{k_{PS}(k-s_{PS}h_{PS})} H(m \,||\, w) \bmod p$$

$$= g^{s_{OS}h_{PS}} g^{k_{PS}k-k_{PS}s_{PS}h_{PS})} H(m \,||\, w) \bmod p$$

$$= g^{s_{OS}h_{PS}} g^{k_{PS}k-k_{PS}s_{OS}k_{PS}^{-1}h_{PS})} H(m \,||\, w) \bmod p$$

$$= g^{s_{OS}h_{PS}} g^{k_{PS}k-s_{OS}h_{PS})} H(m \,||\, w) \bmod p$$

$$= g^{s_{OS}h_{PS}+k_{PS}k-s_{OS}h_{PS}} H(m \,||\, w) \bmod p$$

$$= g^{k_{PS}k} H(m \,||\, w) \bmod p$$

$$= (g^{k_{PS}})^{k} H(m \,||\, w) \bmod p$$

$$= g_{PS}^{k} H(m \,||\, w) \bmod p$$

$$= r$$

Theorem 3. The pair (h_{OS}, s'_{OS}) is a valid signature of the warrant w.

Proof. We have to check that $h' = h_{OS}$, which is equivalent with $R' = R_{OS}$. We have:

$$R' = \left(h_{OS}Q + s'_{OS}P\right)H(w) =$$

$$= \left(h_{OS}xP + \left(k'_{OS} - xh_{OS}\right)P\right)H(w)$$

$$= \left(h_{OS}xP + k'_{OS}P - xh_{OS}P\right)H(w)$$

$$= k'_{OS}PH(w)$$

$$= R_{OS}$$

Theorem 4. The tuple $(m, w, Q_{PS}, s'_{OS}, h'_{PS}, s')$ is a valid proxy signature of the message m.

Proof. We have to prove that $h'_{PS} = h''$, which is equivalent with $R'' = R$. We have:

$$R'' = \left(s'_{OS}h'_{PS}P + s'Q_{PS}\right)H(m \,||\, w) =$$

$$= \left(s'_{OS}h'_{PS}P + \left(k - s'_{PS}h'_{PS}\right)k_{PS}P\right)H(m \,||\, w)$$

$$= \left(s'_{OS}h'_{PS}P + kk_{PS}P - s'_{PS}h'_{PS}k_{PS}P\right)H(m \,||\, w)$$

$$= \left(s'_{OS} h'_{PS} P + kk_{PS} P - s'_{OS} k_{PS}^{-1} h'_{PS} k_{PS} P\right) H(m \| w)$$

$$= \left(s'_{OS} h'_{PS} P + kk_{PS} P - s'_{OS} h'_{PS} P\right) H(m \| w)$$

$$= kk_{PS} PH(m \| w)$$

$$= kQ_{PS} H(m \| w)$$

$$= R$$

5.2 Verifiability

The verifier obtains the proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ and then he/she can gain the identities of the proxy signer and the original signer, valid periods of delegation from the delegation warrant w. Afterwards, the verifier can check the proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ by using the equations (2), (3) and (4). Also, since the proxy private key is created interactively between the original signer and the proxy signer (see section 3.2), a verifier can be aware of the agreement upon signing a message.

5.3 Strong unforgeability

In this section we have to prove that our proxy signature scheme is secure against forgery attacks.

Theorem 5. The proposed proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ of the message *m* are secure against existential forgery.

Proof. Because the signature scheme [23] is secure against existential forgery, this allows only the proxy signer to generate the proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ for a message *m*. If an attacker wants to forge a proxy signature, he/she needs the private key s_{PS} of the proxy signer. Because $s_{PS} = s_{OS} k_{PS}^{-1} \bmod q$ and $g_{PS} = g^{k_{PS}} \bmod p$ contains a secret k_{PS} chosen by the proxy signer, an attacker can not get it from $g_{PS} = g^{k_{PS}} \bmod p$. It is not feasible according to the discrete logarithm problem. Also, the hash function H has the feature that it is infeasible to generate two distinct inputs with matching outputs. So, an adversary cannot find a value $m' \neq m$ with $H(m \| w \| r) = H(m' \| w \| r)$ and $H(m \| w \| r'') = H(m' \| w \| r'')$.

5.4 Strong identifiability

The delegation warrant *w* contains the identities and the public keys of the original signer and the proxy signer, the valid period, limits authority, the type of the message to be signed. Therefore, anyone can determine the identities of the original signer and the proxy signer from the proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$.

5.5 prevention of misuse

The delegation warrant w (which contains the information about the type of the message can be signed by the proxy signer) is signed by the original signer. Therefore, the proxy signer can not use the proxy key to generate a proxy signature for other purposes, other than the original signer delegated.

5.6 Efficiency

We compare in Table 1 the computation time of our proxy signature scheme with the proxy signature schemes [4], [7], [14], [22], [24].

In our proxy signature scheme and the proxy signature schemes in [4], [7], [14], [22], [24], the size of *q* is 160 bits and the size of *p* is at least 512 bits (in [22] and [24] we have n_0 instead of *p*). For the security reason, a 512-bit prime provides marginal security, such that, we suppose that *p* and n_0 are two 768-bit integers. In this case, one modular exponentiation takes on 240 modular multiplications. In the proxy key generation algorithm we have one modular exponentiation and one modular multiplications (denote by MM in Table 1), in total 241 modular multiplications. The proxy signer needs 480 modular multiplications to verify the signature of the delegation warrant *w*. In the proxy signature generation algorithm, the proxy signer needs 241 modular multiplications. A verifier needs 480 modular multiplications to verify the proxy signature $(m, w, g_{PS}, s_{OS}, h_{PS}, s)$ of the message *m*. We neglect the time complexity of the hash function. From the Table 1 above, we conclude that our proposed proxy signature scheme is efficient in the proxy key generation phase, the proxy key verification phase, the signature generation phase and the signature verification phase. Also, our proposed scheme does not need pairing computation which is the most expensive cost.

Table 1. Computation time of the proxy signature schemes

Proxy Signature Schemes	**Proxy Key Generation**	**Proxy Key Verification**	**Signature Generation**	**Signature Verification**
Chen et al. [4]	241 MM	721 MM	242 MM	725 MM
Okamoto et al. [14]	482 MM	242 MM	485 MM	483 MM
Lee et al. [7]	241 MM	241 MM	723 MM	723 MM
Shao [22]	241 MM	242 MM	483 MM	482 MM
Shao [24]	241 MM	240 MM	720 MM	720 MM
Our Proxy Scheme	241 MM	240 MM	241 MM	480 MM

6. Conclusion

In this paper we constructed a proxy signature scheme based on the discrete logarithm problem. We analyzed its security and showed that our proxy signature scheme meets all the security requirements for a proxy signature scheme. Also, we presented an elliptic curve version of our proxy signature scheme.

REFERENCES

1. CAO, F., Z. F. CAO, **A Secure Identity-based Proxy Multi-signature Scheme**, Information Sciences, 2009, vol. 179, pp. 192-302.
2. CAO, F., Z. F. CAO, **A Secure Identity-based Multi-proxy Signature Scheme**, Computers and Electrical Engineering, vol. 35, 2009, pp. 86-95.
3. CHAUM, D., A. FIAT, M. NAOR, **Untraceable Electronic Cash**, Proc. of the Crypto'88, 1990, pp. 319-327.
4. CHEN, I., M. CHANG, Y.-S. YEH, **Design of Proxy Signature in the Digital Signature Algorithm (DSA)**, Journal of Information Science And Engineering, vol. 22, 2006, pp. 965-973.
5. ELGAMAL, T., **A Public Key Cryptosystem and Signature Scheme Based on Discrete Logarithms**, IEEE Transactions IT-31(4), 1985, pp. 469-472.
6. KOBLITZ, N., **Elliptic Curve Cryptosystems**, Mathematics of Computation, vol. 48, 1987, pp. 203-209.
7. LEE, B., H. KIM, K. KIM, **Secure Mobile Agent using Strong Nondesignated Proxy Signature**, Proc. of the Australasian Conference on Information Security and Privacy, LNCS 2119, 2001, pp. 474-486.
8. LIU, Y., H. WEN, C. LIN, **Proxy-Protected Signature Secure Against the Undelegated Proxy Signature Attack**, Computers and Electrical Engineering, vol. 33(3), 2007, pp. 177–185.
9. MAMBO, M., K. USUDA, E. OKAMOTO, **Proxy Signatures: Delegation of the Power to Sign Messages**, IEICE Transactions on Fundamentals, vol. E79-A, 1996, pp. 1338-1354.
10. MAMBO, M., K. USUDA, E. OKAMOTO, **Proxy Signatures for Delegating Signing Operation**, Proc. Third ACM Conference on Computer and Communications Security, ACM press, 1996, pp. 48-57.
11. MENEZES, A., D. JOHNSON, S. VANSTONE, **The Elliptic Curve Digital Signature Algorithm (ECDSA)**, International Journal of Information Security, vol. 1(1), 2001, pp. 36-63.
12. NIST, **Secure Hash Signature Standard (SHS)**, National Institute of Standards and Technology, FIPSP 180-2, 2002.
13. OKAMOTO, T., **Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes**, Advances in Cryptology – CRYPTO’92, Springer-Verlag, 1983, pp. 31-53.
14. OKAMOTO, T., M. TADA, E. OKAMOTO, **Extended Proxy Signatures**

for Smart Cards, Proc. of Information Security Workshop (ISW'99), LNCS 1729, Springer-Verlag, 1999, pp. 247–58.

15. OROS, H., POPESCU C., **A Secure and Efficient Off-line Electronic Payment System for Wireless Networks**, International Journal of Computers, Communications and Control, vol. 5(4), 2010, pp. 551-557.

16. PARK, J. H., B. G. KANG, J. W. HAN, **Cryptanalysis of Zhou et al.'s Proxy-protected Signature Schemes**, Applied Mathematics and Computation, vol. 169(1), 2005, pp. 192–197.

17. POPESCU, C., **An Electronic Cash System Based on Group Blind Signatures**, Informatica, vol. 17, 2006, pp. 551-564.

18. POPESCU, C., **A Secure and Efficient Off-line Electronic Transaction Protocol**, Studies in Informatics and Control, vol. 19(1), 2010, pp. 27-34.

19. POPESCU C., **Blind Signature Schemes Based on the Elliptic Curve Discrete Logarithm Problem**, Studies in Informatics and Control, vol. 19(4), 2010, pp. 397-402.

20. RIVEST, R. L., A. SHAMIR, L. ADELMAN, **A Method for Obtain Digital Signatures and Public-Key Cryptosystem**, Communication on ACM , vol. 21(2), 1978, pp. 120-126.

21. SCHNORR, C. P., **Efficient Signature Generation by Smart Cards**, Journal of Cryptology, vol. 3(3), 1991, pp. 161-174.

22. SHAO, Z., **Proxy Signature Schemes Based on Factoring**, Information Processing Letters, vol. 85, 2003, pp. 137-143.

23. SHAO, Z., **A Provably Secure Short Signature Scheme Based on Discrete Logarithms**, Information Sciences, vol. 177, 2007, pp. 5432-5440.

24. SHAO, Z., **Provably Secure Proxy-protected Signature Schemes Based on RSA**, Computers & Electrical Engineering, vol. 35, 2009, pp. 497-505.

25. TRIPATHY, A. C., I. PATRA, D. JENA, **Proxy Blind Signature based on ECDLP**, International Journal of Computer and Network Security, vol. 2(6), 2010.

26. WANG, G., **Designated-verifier Proxy Signatures for e-Commerce**, Proc. IEEE 2004 International Conference on Multimedia and Expo (ICME 2004), 2004, pp. 1731-1734.

27. WANG. G., F. BAO, J. ZHOU, R. H. DENG, **Security Analysis of Some Proxy Signatures**, Proceedings of Information Security and Cryptology, (ICISC'03), 2004, pp. 305-319.

28. WANG, Q., Z. F. CAO, **An Identity-based Strong Designated Verifier Proxy Signature Scheme**, Wuhan University Journal of Natural Sciences, vol. 11(6), 2006, pp. 1633-1635.

29. XUE, Q., Z. CAO, **Factoring Based Proxy Signature Schemes**, Journal of Computational and Applied Mathematics, vol. 195, 2006, pp. 229-241.

30. ZHANG, J., J. MAO, **A Novel ID-based Strong Designated Verifier Signature Scheme**, Information Science, vol. 178, 2008, pp. 733-766.

31. ZHOU, Y., Z. CAO, R. LU, **Provably Secure Proxy-protected Signature Schemes Based on Factoring**, Applied Mathematics and Computation, vol. 164(1), 2005, pp. 83–98.