Blind Signature Schemes Based on the Elliptic Curve Discrete Logarithm Problem

Constantin Popescu

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

Abstract: A blind signature is a form of digital signature in which the content of a message is blinded before it is signed. The first blind digital signature scheme was proposed by Chaum in 1982. Chaum constructed the blind signatures as a key tool for developing anonymous electronic cash system. In this paper we propose two blind signature schemes and an elliptic curve version of Shao's signature scheme. Our schemes are based on the difficulty of solving the elliptic curve discrete logarithm problem.

Keywords: Blind signatures, elliptic curve cryptography, digital signatures, cryptosystems.

1. Introduction

Diffie and Hellman proposed in 1976 the public key cryptography [3]. They have invented the concept of the encryption scheme and the digital signature scheme based on the public key. Since then, several digital signature schemes have been constructed. The most popular signature schemes are the RSA signature scheme [18] and the ElGamal signature scheme [4]. The RSA signature scheme is based on the difficulty of factoring a large composite number and the ElGamal signature scheme is based on the difficulty of solving discrete logarithms. Schnorr proposed in 1991 a variant [19] of the ElGamal signature scheme. Also, NIST proposed the DSA signature scheme [12]. The Schnorr's signature and the DSA signature were shortened to 320 bits.

David Chaum proposed first in 1982 blind signatures [1] in order to construct an electronic version of money (electronic cash system). Blind signatures allow a user to obtain signatures from a signer on any document, in such a way that the signer learns nothing about the message that is being signed. Blind signature schemes have been widely used to protect customers' right to privacy in the untraceable electronic cash (e-cash) systems [2]. However, it is easy to make multiple copies of the electronic coin, which is in the form of number strings. Therefore, blind signature schemes are used in order to eliminate the possible abuse of unlinkability. A number of blind signature schemes have been proposed to date [5], [10], [11], [14], [22]. The blind signature schemes are useful in some applications [21] where the anonymity is a big issue. Examples include the online voting systems and the electronic cash systems [13], [15], [16], [17].

In this paper we propose two blind signature schemes and an elliptic curve version of Shao's signature scheme.

The rest of this paper is organized as follows. In the next section we review the model of a blind signature scheme, the elliptic curves cryptography and the Shao's signature scheme. Then we present our signature schemes in the section 3. Furthermore, we discuss some aspects of security in the section 4. The section 5 concludes the work of our paper.

2. Preliminaries

In this section we review the model of a blind signature scheme, the basic knowledges of the elliptic curve cryptography and the Shao's signature scheme.

2.1 The model of a blind signature scheme

In this section we review the definition of a blind signature scheme and its security [6]. The blind signatures are treated as an interactive protocols between two players:

- A *Signer*, who blindly signs a document *m*.
- A *User*, who obtains the signature of her document *m*.

Definition 1. A blind signature scheme **(Signer, User, Gen, Ver)** is defined by the two Interactive Turing machines **(Signer, User)** and the following algorithms:

- The key generation algorithm **Gen**: is a probabilistic polynomial time key generation which takes as an input a security parameter 1^k and outputs a pair (pk,sk) of public and secret keys.
- The verification algorithm **Ver**: is a deterministic polynomial time algorithm which takes as input the tuple $(pk,m,\sigma(m))$ and outputs *accept/reject*. If both Signer and User follow the protocol then the Signer always outputs *completed* and the output of user is always *accepted* by the verification algorithm.

The *Signer*(*pk,sk*) and the *User*(*pk,m*) are two polynomially bounded probabilistic Interactive Turing machines. The both machines have the separate tapes: read-only input tape, write-only output tape, a read/write work tape, a read-only random tape, and two communication tapes, a read-only and a write-only tape. The *Signer* and the *User* follow an interactive protocol of some polynomial number of rounds. The *Signer* outputs either *completed* or *non-completed* and the *User* outputs either *fail* or $\sigma(m)$.

The security of a blind signature scheme consists of two properties: blindness and non-forgeability.

2.2 Elliptic curve cryptography

In 1985, Miller[9] and Koblitz[7] introduced Elliptic Curve Cryptography (ECC) which has attracted increasing attention in recent years due to its shorter key length requirement in comparison with other public key cryptosystems such as DSA [12], ElGamal [4] and RSA[18]. For example, 160-bit elliptic curve version of DSA signature algorithm (ECDSA) has a security level equivalent to 1024-bit DSA signature algorithm. Such advantages make elliptic curve cryptography a better choice for public key cryptography.

The elliptic curve cryptosystems are based on the elliptic curve logarithm problem over a finite field. Unlike other popular cryptosystems such as DSA, RSA or ElGamal, the elliptic curve cryptosystem is much more difficult to break at equivalent key lengths.

Table 1 compares the key sizes for different cryptosystems to encryption for comparable levels of security against brute-force attacks. ECC is especially well suited for constrained environments such as smart cards, mobile phones, PDAs, digital postage marks.

Table 1. NIST Guidelines for Public-Key Sizes (Key size in bits)

Symmetric Encryption (3DES, AES)	DSA, RSA and Diffie-Hellman	Elliptic Curve
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

An elliptic curve over a finite field F_p of characteristic greater than three can be constructed by choosing of two variables a and b within the field F_p.

Definition 2. *The elliptic curve is the set of points* (x,y) *which satisfy the elliptic curve equation* $y^2 = x^3 + ax + b (\text{mod } p)$, *where* $x,y \in F_p$, *together with a special point ("point at infinity") denoted* O *and* $4a^3 + 27b^2 \neq 0 (\text{mod}\, p)$.

The elliptic curve group is an additive abelian group with the point O which is the identity element. The formulas for addition of two points on an elliptic curve over a finite field F_p of characteristic greater than three are given as follows. Let $P(x_1,y_1)$ and $Q(x_2,y_2)$ be elements of the elliptic curve group. Then $P+Q=(x_3,y_3)$, where

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases} \tag{1}$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if} \quad P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if} \quad P = Q \end{cases} \tag{2}$$

Next, we give a definition of the elliptic curve discrete logarithm problem [8].

Definition 3. *Let* $\boldsymbol{E}$ *be an elliptic curve defined over a finite field* $\boldsymbol{F_p}$ *and let* $\boldsymbol{P \in E(F_p)}$ *be a point of order* $\boldsymbol{n}$*. Given* $Q \in E(F_p)$, *the elliptic curve discrete* *logarithm problem is to find the integer* $k, 0 \le k \le n-1$, *such that* $Q = k \cdot P$.

Elliptic curve cryptography is particularly useful in applications where the memory, the bandwidth and/or the computational power is limited (e.g., smartcards, wireless communications).

2.3 Shao's signature scheme

We shortly describe Shao's signature scheme [20]. The parameters of this scheme are as follows. Let p and q be large prime numbers such that $q|(p\text{-}1)$. Let H and H' be ideal hash functions where $H: \{0,1\}^* \times Z_p^* \to \{0,1\}^{|q|/2}$, $F: \{0,1\}^* \to Z_p^*$.

The key generation algorithm: Picks a random $x \in Z_q^*$ as the private key. The corresponding public key is $y=g^x$.

The signing algorithm: Picks a random $k \in Z_q^*$ and computes $f=F(m)$, $r=g^k f$, $h=H(m,r)$ and $s=k\text{-}hx$, where $m \in \{0,1\}^*$. The signature of the message m is *sigma*$=(h,s)$.

The verification algorithm: The inputs are the public key y and the message m and the signature *sigma*$=(h,s)$. Computes $f=F(m)$, $r'=fy^h g^s$ and $h'=H(m,r')$. If $h=h'$, the verification algorithm outputs *valid*, otherwise it outputs *invalid*.

3. Proposed Signature Schemes

3.1 Parameters of our signature schemes

Let p and q be large prime numbers such that $q|(p\text{-}1)$. Let H and H' be collision-resistant hash functions where:

$$H: \{0,1\}^* \times Z_p^* \to \{0,1\}^{|q|/2} \tag{3}$$

and

$$H': \{0,1\}^* \to Z_p^*. \tag{4}$$

Choose an elliptic curve E defined over a finite field F_p of characteristic greater than three and calculate the order of the elliptic curve $\#E(F_p)$. Let P be a point of order q on the elliptic curve E, where $q|\#E(F_p)$. Let $m \in \{0,1\}^*$ be a message.

3.2 Blind signature version of Shao's signature scheme

In this subsection we construct a blind signature scheme from the Shao's signature scheme.

Key Generation: Picks a random $x_2 \in Z_q^*$ as the private key. The corresponding public key is $y_2 = g^{x_2} \bmod p$.

Signature Generation (Blind Protocol 1): In order to get the signature of a secret message m, the user asks the signer to initiate a communication:

- The signer selects $\bar{k} \in [2, q-1]$, computes $\bar{r} = g^{\bar{k}} \bmod p$ and sends $\bar{r}$ to the user.
- The user randomly selects two blinding factors $\alpha, \beta \in [2, q-1]$ and computes:

$$r_2 = g^\alpha \bar{r}^\beta H'(m) \bmod p.$$

The user also computes $h_2 = H(m, r_2)$ $\bar{m} = h_2 \beta^{-1} \bmod q$ and sends the value $\bar{m}$ to the signer.

- The signer computes:

$$\bar{s} = (\bar{k} - \bar{m} x_2) \bmod q \tag{5}$$

and sends it to the user.

- The user computes:

$$s_2 = (\bar{s}\beta + \alpha) \bmod q. \tag{6}$$

The blind signature of the message m is $S_2 = (h_2, s_2)$.

Signature Verification: Computes $r_2' = y_2^{h_2} g^{s_2} H'(m) \bmod p$ and $h_2' = H(m, r_2')$. If $h_2 = h_2'$, the blind signature $S_2 = (h_2, s_2)$ of the message m is valid, otherwise it is invalid.

3.3 Elliptic curve version of Shao's signature scheme

In this subsection we propose the elliptic curve version of Shao's signature scheme.

Key Generation: Picks a random $x_3 \in Z_q^*$ as the private key. The corresponding public key is the point Q_3, where $Q_3 = x_3 P$.

Signature Generation: Picks a random $k_3 \in Z_q^*$ and computes $R_3 = k_3 PH'(m)$, $h_3 = H(m, x_{R_3})$ and $s_3 = (k_3 - h_3 x_3) \bmod q$, where $m \in \{0,1\}^*$ and x_{R_3} is x-coordinate of the point R_3. The signature of the message m is $S_3 = (h_3, s_3)$.

Signature Verification: The inputs are the public key Q_3 and the message m and the signature $S_3 = (h_3, s_3)$. Computes $R_3' = (h_3 Q_3 + s_3 P) H'(m)$ and $h_3' = H(m, x_{R_3'})$, where $x_{R_3'}$ is x-coordinate of the point R_3'. If $h_3 = h_3'$, the verification algorithm outputs *valid*, otherwise it outputs *invalid*.

3.4 Elliptic Curve Blind Signature Scheme

In this subsection we describe an elliptic curve version of the blind signature scheme from subsection 3.2.

Key Generation: Picks a random $x_4 \in Z_q^*$ as the private key. The corresponding public key is the point Q_4, where $Q_4 = x_4 P$.

In order to get the signature of a secret message *m*, the user asks the signer to initiate a communication.

Signature Generation (Blind Protocol 2): In order to get a blind signature of a secret message *m*, the user asks the signer to initiate a communication:

- The signer selects $k_4 \in [2, q-1]$, computes the point $R_4' = k_4 P$ and sends R_4' to the user.
- The user randomly selects two blinding factors $\alpha, \beta \in [2, q-1]$ and computes:

$R_4 = \alpha P + \beta R_4'$ and $h_4 = H(m, x_{R_4})$,

where x_{R_4} is x-coordinate of the point R_4. The user also computes $m' = h_4 \beta^{-1} \bmod q$ and sends the value m' to the signer.

- The signer computes

$s_4' = (k_4 - m' x_4) \bmod q$ and sends it to the user.

- The user computes:

$$s_4 = (s_4' \beta + \alpha) \bmod q. \qquad (7)$$

The elliptic curve blind signature of the message m is $S_4 = (h_4, s_4)$.

Signature Verification: Computes $R_4'' = h_4 Q_4 + s_4 P$ and $h_4' = H(m, x_{R_4''})$, where $x_{R_4''}$ is x-coordinate of the point R_4''. If $h_4 = h_4'$, the elliptic curve blind signature $S_4 = (h_4, s_4)$ of the message m is valid, otherwise it is invalid.

4. Security Analysis

In this section we discuss aspects of security of our signature schemes.

4.1 Blindness

Theorem 1. The both protocols **Blind Protocol 1** and **Blind Protocol 2** are two blind signature schemes.

Proof. Our protocols used the blinding factors $\alpha, \beta \in [2, q-1]$ and these values are selected at random. Also, the user sends only the values $\overline{m}$ and m' to the signer.

Since H is a collision-resistant hash function, the signer can't recover the original message m from the following two equations $h_4 = H(m, x_{R_4})$ and $h_2 = H(m, r_2)$. If the signature $S_2 = (h_2, s_2)$ is valid with the values $\overline{k}$, $\overline{r} = g^{\overline{k}} \bmod p$, $\overline{m}$ and $\overline{s} = (\overline{k} - \overline{m} x_2) \bmod q$, then he following equations must hold for α and β:

$$\overline{m} = h_2 \beta^{-1} \bmod q$$
$$s_2 = (\overline{s} \beta + \alpha) \bmod q$$
$$r_2 = g^{\alpha} \overline{r}^{\beta} H'(m) \bmod p.$$

We have that the blinding factors α and β are uniquely computed by the above equations:

$$\beta = h_2 \overline{m}^{-1} \bmod q$$
$$\alpha = (s_2 - \overline{s} h_2 \overline{m}^{-1}) \bmod q.$$

We obtain:

$$\overline{k} \beta + \alpha = \overline{k} h_2 \overline{m}^{-1} \bmod q + s_2 - \overline{s} h_2 \overline{m}^{-1} =$$
$$s_2 + h_2 x_2 \pmod q.$$

So, we have:

$$g^{\alpha}\overline{r}^{\beta}H'(m)=g^{\alpha}g^{\overline{k}\beta}H'(m)=$$
$$=g^{\alpha+\overline{k}\beta}H'(m)=g^{s_2+h_2x_2}H'(m)=$$
$=y_2^{h_2}g^{s_2}H'(m)=r_2'$, which implies the equality $r_2=r_2'$. Therefore, the **Blind Protocol 1** is a blind signature scheme.

We follow the same steps for the **Blind Protocol 2** in order to show that the **Blind Protocol 2** is a blind signature scheme. □

4.2. The correctness of the signatures

We have to prove that the signatures *S*₂=(*h*₂,*s*₂), *S*₃=(*h*₃,*s*₃), *S*₄=(*h*₄,*s*₄) are correct (valid).

Theorem 2. The signature $S_2=(h_2,s_2)$ is a valid blind signature of the message *m*.

Proof. The verification equation for $S_2=(h_2,s_2)$ is $h_2=h_2'$, which is equivalent with $r_2=r_2'$. Obviously, the relation follows from:

$$\begin{aligned}
r_2' &= y_2^{h_2}g^{s_2}H'(m)\bmod p=\\
&=g^{x_2h_2}g^{\overline{s}\beta+\alpha}H'(m)\bmod p\\
&=g^{x_2h_2+(\overline{k}-\overline{m}x_2)\beta+\alpha}H'(m)\bmod p\\
&=g^{x_2h_2+\overline{k}\beta-\overline{m}x_2\beta+\alpha}H'(m)\bmod p\\
&=g^{x_2h_2+\overline{k}\beta-h_2\beta^{-1}x_2\beta+\alpha}H'(m)\bmod p\\
&=g^{\overline{k}\beta+\alpha}H'(m)\bmod p\\
&=g^{\alpha}g^{\overline{k}\beta}H'(m)\bmod p\\
&=g^{\alpha}g^{\overline{k}\beta}H'(m)\bmod p\\
&=g^{\alpha}\overline{r}^{\beta}H'(m)\bmod p\\
&=r_2.
\end{aligned}$$

Theorem 3. The signature $S_3=(h_3,s_3)$ is a valid signature of the message *m*.

Proof. The verification equation for $S_3=(h_3,s_3)$ is $h_3=h_3'$, which is equivalent with $R_3=R_3'$.

The derivation of the verification is described as follows:

$$\begin{aligned}
R_3' &=(h_3Q_3+s_3P)H'(m)=\\
&=h_3x_3P+(k_3-h_3x_3)PH'(m)\\
&=(h_3x_3P+k_3P-h_3x_3P)H'(m)\\
&=k_3PH'(m)\\
&=R_3.
\end{aligned}$$

Theorem 4. The signature $S_3=(h_3,s_3)$ is a valid blind signature of the message *m*.

Proof. The verification equation for the signature $S_4=(h_4,s_4)$ is $h_4=h_4'$, which is equivalent with $R_4=R_4''$. The validity of the signature $S_4=(h_4,s_4)$ for the message *m* follows from:

$$\begin{aligned}
R_4'' &=h_4Q_4+s_4P\\
&=h_4x_4P+(s_4'\beta+\alpha)P\\
&=h_4x_4P+(k_4-m'x_4)\beta P+\alpha P\\
&=h_4x_4P+k_4\beta P-m'x_4\beta P+\alpha P\\
&=h_4x_4P+k_4\beta P-h_4\beta^{-1}x_4\beta P+\alpha P\\
&=k_4\beta P+\alpha P\\
&=\alpha P+k_4P\beta\\
&=\alpha P+\beta R_4'\\
&=R_4.
\end{aligned}$$

4.3. Non-forgeability of the signatures

The hardness of forgery in our signature schemes is determined by security parameters p and q. We let p be at least 512 bits and q be 160 bits. The security of our proposed signature schemes is based on the difficulty of solving the elliptic curve discrete logarithm problem.

Theorem 5. The proposed signatures $S_2=(h_2,s_2)$, $S_3=(h_3,s_3)$ and $S_4=(h_4,s_4)$ of the message *m* are secure against existential forgery.

Proof. Because the signature scheme [20] is secure against existential forgery, this allows only the legal signer to generate the signature for the message *m*. Also, the hash function *H* has the feature that it is infeasible to generate two distinct inputs with matching outputs. So, the user cannot find a value *m'≠m* with *H*(*m*,*r*₂)=*H*(*m'*,*r*₂) and $H(m,x_{R_4})=H(m',x_{R_4})$, where x_{R_4} is *x*-coordinate of the point *R*₄. If an adversary has the points R_3,P,R_4' and *H'*(*m*) he cannot determine *k*₃ and *k*₄ from the equations*R*₃=*k*₃*PH'*(*m*) and $R_4'=k_4P$, because he must solve the elliptic curve discrete logarithm problem.

5. Conclusions

In this paper we proposed a digital signature scheme and two blind signature schemes based on the elliptic curve discrete logarithm problem. We proved that our signature schemes meet the security requirements such as blindness, correctness and non-forgeability of the signatures.

REFERENCES

1. CHAUM, D., **Blind Signature for Untraceable Payments**, Proc. of Eurocrypt '82, Plenum Press, 1983, pp. 199-203.
2. CHAUM, D., A. FIAT, M. NAOR, **Untraceable Electronic Cash**, Proc. of the Crypto '88, 1990, pp. 319-327.
3. DIFFIE, W., M. E. HELLMAN, **New Directions in Cryptography**, IEEE Transactions IT-22, 1976, pp. 644-654.
4. ELGAMAL, T., **A Public Key Cryptosystem and Signature Scheme Based on Discrete Logarithms**, IEEE Transactions IT-31(4), 1985, pp. 469-472.
5. FAN, C., L. C. WU, V. HUANG, **Cryptanalysis on Chen-Qiu-Zheng Blind**
6. **Signature Scheme**, Applied Mathematical Sciences, Vol. 2, (16), 2008, pp. 787-791.
7. JUELS, A., M. LUBY, R. OSTROVSKY, **Security of Blind Digital Signatures**, Lecture Notes In Computer Science; Vol. 1294 Proc. of the 17th Ann. Intl. Cryptology Conference on Advances in Cryptology, 1997, pp: 150-164.
8. KOBLITZ, N., **Elliptic Curve Cryptosystems**, Mathematics of Computation, vol, 48, 1987, pp. 203-209.
9. MENEZES, A., **Elliptic Curve Public Key Cryptosystems**, Kluwer Academic Publishers, 1993.
10. MILLER, V., **Uses of Elliptic Curves in Cryptography**, Advances in Cryptology, Proc. of Crypto ’85, Lecture Notes in Computer Sciences, 218, Springer-Verlag, 1986, pp. 417-426.
11. MOLDOVYAN, N., A. MOLDOVYAN, **Blind Collective Signature Protocol Based on Discrete Logarithm Problem**, Intl. Journal of Network Security, vol. 11(2), 2010, pp. 106-113.
12. MOLDOVYAN N. A., **Blind Signature Protocols from Digital Signature Standards**, Intl. Journal of Network Security, Vol.12(3) , 2011, pp. 202-210.
13. NIST. **Digital Signature Standard (DSS), Publication** 186-3, Federal Information Processing Standards, 2009.
14. OROS H., C. POPESCU, **A Secure and Efficient Off-line Electronic Payment System for Wireless Networks**, Intl. J. of Computers, Comm. and Control, Suppl. Issue, 2010.
15. POINTCHEVAL, D., J. STERN, **Security Arguments for Digital Signatures and Blind Signatures**, Journal of Cryptology 13, 2000, pp. 361-396.
16. POPESCU, C, **An Electronic Cash System Based on Group Blind Signatures**, Informatica 17, 2006, pp. 551-564.
17. POPESCU, C, **A Secure and Efficient Off-line Electronic Transaction Protocol**, Studies in Informatics and Control, vol. 19(1), 2010, pp. 27-34.
18. POPESCU, C., H. OROS, **An Off-line Electronic Cash System with Multiple Banks**, Intl. J. of Computers, Comm. and Control, Suppl. Issue, 2006, pp. 386-392.
19. RIVEST, R. L., A. SHAMIR, L. ADELMAN, **A Method for Obtain Digital Signatures and Public-key Cryptosystem**, Comm. on ACM 21 (2), 1978, pp. 120-126.
20. SCHNORR, C. P., **Efficient Signature Generation by Smart Cards**, Journal of Cryptology 3(3), 1991, pp, 161-174.
21. SHAO, Z., **A Provably Secure Short Signature Scheme Based on Discrete Logarithms**, Information Sciences 177, 2007, pp. 5432-5440.
22. THORSTEINSSON, G., PAGE, T., A. NICULESCU, **Using Virtual Reality for Developing Design Communication**, Studies in Informatics and Control, vol. 19(1), 2010, pp. 93-106.
23. TRIPATHY, A. C., PATRA, I., JENA, D., **Proxy Blind Signature based on ECDLP**, Intl. J. of Computer and Network Security, Vol. 2(6), 2010.