Identity-based Digital Signature Scheme Without Bilinear Pairings

He Debiao*, Chen Jianhua, Hu Jin
School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei, China, 430072

Abstract*:* Many identity-based digital signature schemes using bilinear pairings have been proposed. But the relative computation cost of the pairing is approximately twenty times higher than that of the scalar multiplication over elliptic curve group. In order to save the running time and the size of the signature, in this letter, we propose an identity based signature scheme without bilinear pairings. With both the running time and the size of the signature being saved greatly, our scheme is more practical than the previous related schemes for practical application.
Key words: Digital signature, Identity-based cryptography, Bilinear pairings, Elliptic curve

1. Introduction

The concept of identity-based (ID-based) cryptography was first formulated by Shamir [1]. In ID-based cryptography, a user's unique identifier acts as the user's public key; the corresponding private key generated by a trusted Key Generation Center (KGC) acts as the user's implicit certificate, thereby removing the requirement of public key certificate.

Shamir [1] was the first to propose the ID-based signature scheme in which the signature has 2048 bits when one uses a 1024-bit RSA modulus. In 1988, Guillou and Quisquater [2] improved Shamir's scheme and shortened the signature size to 1184 bits when one uses a 1024-b RSA modulus and a 160-b hash function, e.g., Secure Hash Standard. However, the computation of modular exponentiation required by the above schemes make unavailable the application of the schemes in some environment, such as mobile devices, where the computation ability and battery capacity of mobile devices are limited. Fortunately, Elliptic curve cryptosystem (ECC) [3,4] has significant advantages like smaller key sizes, faster computations compared with other public-key cryptography. Many IBS schemes using the elliptic curve pairings have been proposed [5-7]. In spite of the significant improvements in the computation speed, the pairing is still regarded as the most expensive cryptography primitive. The relative computation cost of a pairing is approximately twenty times higher than that of the scalar multiplication over elliptic curve group [8]. Therefore, IBS schemes without bilinear pairings would be more appealing in terms of efficiency.

In this letter, we present an IBS scheme without pairings. The scheme rests on the elliptic curve discrete logarithm problem (ECDLP).With the pairing-free realization, the scheme's overhead is lower than that of previous schemes [5-7] in both computation and communication.

2. Background of elliptic curve group

We will just give a simple introduction of elliptic curve defined on prime field F_p in this part,

*Corresponding author.
E-mail: hedebiao@163.com, *Tel*:+0086015307184927

while the knowledge of elliptic curve defined on binary field can be found in [3,4].

Let the symbol E/F_p denote an elliptic curve E over a prime finite field F_p, defined by an equation

$$y^2 = x^3 + ax + b, \quad a, b \in F_p \qquad (1)$$

and with the discriminant

$$\Delta = 4a^3 + 27b^2 \neq 0. \qquad (2)$$

The points on E/F_p together with an extra point O called the point at infinity form a group

$$G = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \cup \{O\}. \quad (3)$$

Let the order of G be n. G is a cyclic additive group under the point addition “+” defined as follows: Let $P, Q \in G$, l be the line containing P and Q (tangent line to E/F_p if $P = Q$), and R, the third point of intersection of l with E/F_p. Let l' be the line connecting R and O. Then P “+” Q is the point such that l' intersects E/F_p at R and O and P “+” Q. Scalar multiplication over E/F_p can be computed as follows:

$$tP = P + P + \cdots + P(t \ times) \qquad (4).$$

The following problems defined over G are assumed to be intractable within polynomial time.

Eliptic curve discrete logarithm problem: For $x \in_R Z_n^*$ and P the generator of G, given $Q = x \cdot P$ compute a.

3. Our scheme

3.1.Scheme description

In this section, we present an ID-based signature scheme without pairing. Our scheme consists of four algorithms: *Setup*, *Extract*, *Sign*, and *Verify*.

Setup: Takes a security parameter k, returns system parameters and a master key. Given k, KGC does as follows.

1) Choose a k-bit prime p and determine the tuple $\{F_p, E/F_p, G, P\}$ as defined in Secttion 2.

2) Choose the master private key $x \in Z_n^*$ and compute the master public key

$P_{pub} = x \cdot P$.

3) Choose two cryptographic secure hash functions $H_1 : \{0,1\}^* \to Z_n^*$ and

$H_2 : \{0,1\}^* \times G \to Z_p^*$.

4) Publish $\{F_p, E/F_p, G, P, P_{pub}, H_1, H_2\}$ as system parameters

and keep the master key x secretly.

Extract: Takes as input system parameters, master key and a user's identifier, returns the user's ID-based private key. With this algorithm, KGC works as follows for each user U with identifier ID_U.

1) Choose at random $r_U \in Z_n^*$, compute $R_U = r_U \cdot P$ and $h_U = H_1(ID_U, R_U)$.

2) Compute $s_U = r_U + h_U x$.

U 's private key is the tuple (s_U, R_U) and is transmitted to U via a secure out-of-band channel. U can validate her private key by checking whether the equation

$$s_U \cdot P = R_U + h_U \cdot P_{pub}$$

holds. The private key is valid if the equation holds and vice versa.

Sign: Takes as input system parameters, user's private key (s_U, R_U) and a message m, returns a signature of the message m. The user U does as the follows.

1) Choose at random $l \in Z_n^*$ to compute $R = l \cdot P$.

2) Compute $h = H_2(m, R)$.

3) Verify whether the equation $\gcd(l + h, n) = 1$ holds: Continue if it does and return to step 1) otherwise.

4) Compute $s = (l + h)^{-1} s_U \bmod n$.

5) The resulting signature is (ID_U, R_U, R, s).

Verify: To verify the signature (ID_U, R_U, R, s) for message m and identity ID_U, the verifier first computes $h = H_2(m, R)$, $h_U = H_1(ID_U, R_U)$ and then checks whether

$$s \cdot (R + h \cdot P) = R_U + h_U \cdot P_{pub}$$

Accept if it is equal. Otherwise reject.

Since $R = l \cdot P$ and $s = (l+h)^{-1} s_U \bmod n$, we have

$$\begin{aligned} s \cdot (R + h \cdot P) &= (l+h)^{-1} \cdot s_U \cdot (l \cdot P + h \cdot P) \\ &= (l+h)^{-1} s_U \cdot (l+h) \cdot P = s_U \cdot P \\ &= R_U + h_U \cdot P_{pub} \end{aligned} \quad (5)$$

Then the correctness of our scheme is proved.

3.2.Security analysis

We prove the security of our scheme Σ in the random oracle model which treats H_1 and H_2 as two random oracles [9] using the signature security model defined in [10]. As for the security of Σ, the following theorem is provided.

Theorem 1: Consider an adaptively chosen message attack in the random oracle model against Σ. If there is an attacker A that can break Σ with at most q_{H_2} H_2-queries and q_S signature queries within time bound t and probability $\varepsilon \geq 10(q_{H_2}+1)(q_{H_2}+q_S)/2^k$, then the ECDLP can be solved within running time $t \leq 23 q_{H_2} t / \varepsilon$ and with probability $\varepsilon' \geq 1/9$.

Proof: Suppose that there is an attacker A for an adaptively chosen message attack against Σ. Then, we show how to use the ability of A to construct an algorithm S solving the ECDLP.

Suppose S is challenged with a ECDLP instance (P, Q) and is tasked to compute $x \in Z_n^*$ satisfying $Q = x \cdot P$. To do so, S sets $\{F_p, E/F_p, G, P, P_{pub} = Q, H_1, H_2\}$ as the system parameter and answers A's queries as follows.

Extract-query: A is allowed to query the extraction oracle for an identity ID_U. S simulates the oracle as follows. It chooses $a_U, b_U \in Z_n^*$ at random and sets

$$R_U = a_U \cdot P_{pub} + b_U \cdot P, \quad s_U = b_U, \quad h_U = H_1(ID_U, R_U) \leftarrow -a_U \bmod n \quad (6)$$

Note that (s_U, R_U) generated in this way satisfies the equation $s_U \cdot P = R_U + h_U \cdot P_{pub}$ in the extract algorithm. It is a valid secret key. S outputs (s_U, R_U) as the secret key of ID_U and stores the value of ($s_U, R_U, H_1(ID_U, R_U), ID_U$) in the H_1-table.

Signature-query: To answer A's signature query on m_i ($1 \le i \le q_S$) and an identity ID_U, S chooses at random $a_i, b_i \in Z_n^*$. Then, it gets $h_U = H_1(ID_U, R_U)$ from H_1-table, and computes $R_i = a_i^{-1} \cdot R_U - b_i \cdot P + a_i^{-1} \cdot h_U \cdot P_{pub}$, $s = a_i$ and sets $h_i = H_1(m_i, R_i) \leftarrow b_i$ and adds (m_i, R_i, b_i) to the H_2-list. If the pair (m_i, R_i) has been defined in the H_2-table. S outputs fail and exits. Since b_i is chosen at random, the probability of fail is no more than $1/n$ and is negligible. It is straightforward to verify that (R_U, R_i, s_i) is a perfect simulation. A will not be able to tell the difference between the simulation and the reality if S does not abort.

If A can forge a valid signature on message m with the probability $\varepsilon \ge 10(q_{H_2}+1)(q_{H_2}+q_S)/2^k$, where m has not been queried to the signature oracle, then a replay of S with the same random tape but different choice of H_2 will output two valid signatures (m, R_U, R_i, h_i, s_i) and (m, R_U, R_i, h_i', s_i'). Then we have

$$s_i \cdot (R + h_i \cdot P) = R_U + h_U \cdot P_{pub}, \tag{7}$$

and

$$s_i' \cdot (R + h_i' \cdot P) = R_U + h_U \cdot P_{pub}. \tag{8}$$

Let $R = r \cdot P$, $R_U = a_U \cdot P_{pub} + b_U \cdot P$, $P_{pub} = Q = x \cdot P$, then we have

$$s_i \cdot (r \cdot P + h_i \cdot P) = a_U \cdot x \cdot P + a_U \cdot P + h_U \cdot x \cdot P, \tag{9}$$

and

$$s_i' \cdot (r \cdot P + h_i' \cdot P) = a_U \cdot x \cdot P + a_U \cdot P + h_U \cdot x \cdot P. \tag{10}$$

then we have

$$s_i' \cdot s_i \cdot (r \cdot P + h_i \cdot P) = s_i' \cdot a_U \cdot x \cdot P + s_i' \cdot a_U \cdot P + s_i' \cdot h_U \cdot x \cdot P, \tag{11}$$

and

$$s_i \cdot s_i' \cdot (r \cdot P + h_i' \cdot P) = s_i \cdot a_U \cdot x \cdot P + s_i \cdot a_U \cdot P + s_i \cdot h_U \cdot x \cdot P. \tag{12}$$

Hence, we have

$$(s_i' \cdot a_U + s_i' \cdot h_U - s_i \cdot a_U - s_i \cdot h_U) \cdot x \cdot P = (s_i' \cdot s_i \cdot h_i - s_i' \cdot a_U - s_i \cdot s_i' \cdot h_i' + s_i \cdot a_U) \cdot P. \tag{13}$$

Let $u = (s_i' \cdot a_U + s_i' \cdot h_U - s_i \cdot a_U - s_i \cdot h_U)^{-1} \bmod n$ and $v = (s_i' \cdot s_i \cdot h_i - s_i' \cdot a_U - s_i \cdot s_i' \cdot h_i' + s_i \cdot a_U) \bmod n$, then, we get $x = uv \bmod n$.

According to [10, Lemma 4], the ECDLP can be solved with probability $\varepsilon' \geq 1/9$ and time $t' \leq 23 q_{H_2} t / \varepsilon$.

4. Comparison with previous scheme

In this section, we will compare the efficiency of our new scheme with Cha et al.'s scheme [5], Yi's scheme [6] and Hess's scheme [7]. In the computation efficiency comparison, we obtain the running time for cryptographic operations using MIRACAL [11], a standard cryptographic library.

The hardware platform is a PIV 3-GHZ processor with 512-MB memory and a Windows XP operation system. For the pairing-based scheme, to achieve the 1024-bit RSA level security, we use the Tate pairing defined over the supersingular elliptic curve $E / F_p : y^2 = x^3 + x$ with embedding degree $2 \cdot q$ is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p a 512-bit prime satisfying $p + 1 = 12qr$. For the ECC-based schemes, to achieve the same security level, we employed the parameter secp160r1[12], recommended by the Certicom Corporation, where $p = 2^{160} - 2^{31} - 1$. The running times are listed in Table 1 where sca.mul. stands for scalar multiplication.

Table 1. Cryptographic Operation Time(in milliseconds)

Pairing	Pairing-based sca.mul	ECC-based sca.mul.	Map-to-point hash
20.04	6.38	2.21	3.04

To evaluate the computation efficiency of different schemes, we use the simple method from [13]. For example, the sign algorithm of our scheme requires one ECC-based scale multiplication; thus, the computation time of the sign algorithm is 2.21 × 1 = 2.21 ms; the verify algorithm has to carry out three ECC-based scalar multiplications, and the resulting running time is 2.21 × 3 = 6.63 ms. As another example, in Cha et al.'s scheme[5], the sign algorithm should carry out two pairing-based scalar multiplications and a map-to-point hash computation; thus, the computation time for a client is 6.38 × 2 +3.04= 15.8 ms; the verify algorithm has to carry out one pairing, and the resulting running time is 20.04 × 1 = 20.04 ms. The size of signature is evaluated by the overall size of the messages generated by the sign algorithm in a scheme. For example, in our scheme, the generated message comprises an identity, two points of elliptic curve and a number in Z_n^*. Assuming that the size of identity is 4B, the resulting signaling traffic is 4 + 40 × 2 + 20 = 104 B. As another example, in Cha et al.'s scheme, the generated message comprises an identity and two points of elliptic curve, then the resulting signaling traffic is 4 + 128 × 2 = 260 B. Table 2 shows the results of the performance comparison.

Table 2. Performance comparison of different schemes

	Running time		Size of signature
	Sign	Verify	
Cha et al.'s scheme [5]	15.8 ms	20.04 ms	260 B
Yi's scheme [6]	19.14 ms	46.46 ms	260B
Hess's scheme [7]	26.42 ms	40.08 ms	132B
Our scheme	2.21 ms	6.63 ms	104 B

According to Table 2, the running time of the sign algorithm of our scheme is 13.98% of Cha et al.'s schemes, 11.54% of Yi's et al.'s scheme and 8.36% of Hess's scheme, the running time of the verify algorithm of our scheme is 33.08% of Cha et al.'s schemes, 14.27% of Yi's et al.'s scheme and 16.54% of Hess's scheme, the size of signature of our scheme is 40% of Cha et al.'s schemes, 40% of Yi's et al.'s scheme and 78.79% of Hess's scheme. Thus our scheme is more useful and efficient than the previous schemes[5-7].

5. Conclusion

In this paper, we have proposed an efficient identity-based digital signature scheme. We also prove the security of the scheme under random oracle. Compared with previous scheme, the new scheme reduces both the running time and the size of signature. Therefore, our scheme is more practical than the previous related schemes for practical application.

6. References

[1]. A. Shamir, Identity-based cryptosystems and signature schemes, Proc. CRYPTO1984, LNCS, vol.196, pp.47–53, 1984.

[2]. L. C. Guillou and J. J. Quisquater, A'paradoxical' identity-based signature scheme resulting from zero-knowledge, in Proc. Crypto'88, Santa Barbara, CA, Aug. 1988, pp. 216–231.

[3]. V.S. Miller, Use of elliptic curves in cryptography. In: Advances in cryptology, proceedings of CRYPTO'85, vol. 218. LNCS, Springer-Verlag; 1986: 417–426.

[4]. Koblitz N. Elliptic curve cryptosystem. Mathematics of Computation 1987, 48:203–209.

[5]. J. C. Cha and J. H. Cheon, An Identity-Based Signature from Gap Diffie-Hellman Groups, PKC 2003, LNCS 2567, pp. 18－30, 2003.

[6]. X. Yi, An Identity-Based Signature Scheme From the Weil Pairing, IEEE COMMUNICATIONS LETTERS, VOL. 7, NO. 2, FEBRUARY 2003, 67-69.

[7]. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310－324. Springer, Heidelberg(2003).

[8]. L. Chen, Z. Cheng, and N.P. Smart, Identity-based key agreement protocols from pairings, Int. J. Inf. Secur, no.6, pp.213–241, 2007.

[9]. M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient schemes, in Proc. 1st ACM Conf. Comput. Commun. Security, 1993, pp. 62–73.

[10]. P. David, S. Jacque，Security Arguments for Digital Signatures and Blind Signatures, Journal

of Cryptology, Vol. 13, No. 3. p. 361-396, 2000.
[11]. Shamus Software Ltd., Miracl library, http://www.shamus. ie/index.php?page=home.
[12]. The Certicom Corporation, SEC 2: Recommended Elliptic Curve Domain Parameters, www.secg.org/collateral/sec2_final.pdf.
[13]. X. Cao, X. Zeng, W. Kou, and L. Hu, Identity-based anonymous remote authentication for value-added services in mobile networks, IEEE Transactions on Vehicular Technology, vol.58, no.7, pp.3508 - 3517, 2009.