ADOPTATION OF WEIL PAIRING IBE FOR SECURE FILE SHARING

Cheong Hyeon Choi
Security Lab., MIS Dept. School of Business
Kwangwoon University
Seoul, Korea
e-mail: chchoi@kw.ac.kr

***Abstract*—Competitive enterprise has lots of secret files in digital form, which is vulnerable to illegal online activity. Our target network accepts authorized users on registered machines. In general it is known that the traditional public-key scheme is suitable to strong authentication and encryption, but it is inferior to the IBE (Identity Based Encryption) scheme in terms of performance and certificate management. Thus we improve the WP (Weil Pairing) IBE scheme so as to be suitable to our SFS (secure file sharing) system with certificate-less key revocation as normalizing public identity. In addition, the man-in-the-middle attack is negligible because the security is based on hard ECDLP (elliptic curve discrete logarithm problem). However, in the perspective of performance, our WP IBE is bound to the complexity of modular-ADD. In the perspective of security, our SFS network is close with single authorized server and multiple registered clients. In addition, the SFS architecture is fortunately useful in DRM network and P2P file sharing network.**

Keywords-IBE; Weil Pairing; Diffie-Hellman; DLP.

I. INTRODUCTION

In general, corporate secrets are accessible inside a restrictive area proofed against exposure in the building. In real, secrets such as manufacturing blueprints are confined so as to be accessible to only CEO or authorized technicians through registered machines in a restrictive area. Such network is composed of a single server and multiple clients, where confidential files are stored in the server and disseminated to the registered clients. Its cryptographical scheme must satisfy stronger security constraints than traditional schemes. Communication between the server and the clients is asymmetric with respect to the cryptographical functions and the exchanged messages [12]. Our target network called SFS (Secure File Sharing) satisfies these properties.

There are a couple of commercial networks similar to our SFS architecture. The first is P2P (Peer-to-Peer) network, which is formed with asymmetric connections, where its file server knows the client keeping a registered file, and redirects a file request to the corresponding client [19]. The second is DRM (Digital Right Management) network, which consists of single DRM server with copyrighted digital materials and multiple purchasers (clients). The DRM server must verify both a purchaser and its digital player with so-called two-factor signature, then package the purchased materials using encryption algorithm [22]. In this network, the important security issue is authenticity [15].

It is desirable to use the public-key scheme for such security issue. Nevertheless, we note that the traditional public-key scheme is inappropriate in the perspectives of performance and certificate management. In 1984, Shamir suggested Identity Based Encryption (IBE) scheme to lessen a burden of certificate management [1]. The IBE needs no more public-key certificate since any string {0,1}* can be a public key. In the early stage, it was proposed to use an e-mail address as a public key. The early IBE, however, might expose key material to adversary since the public ID and crypto-functions are open without authentication. In order to avoid such exposure risk, we combine a public identity and a period as normalizing to temporal public key [2]. As the result, the normalized public key will be ineffective after its valid period, which means the key revocation [15], called as *certificate-less key revocation*.

We modify the WP (Weil Pairing) IBE scheme for better performance and stronger security so as to be appropriate to our SFS network, which is characterized as follows:

① One-to-many: Data transmission between single file server and multiple file consumers

② CS model: Asymmetric communication between the file server and consumers (clients).

③ Closedness: All links of the network formed by secure channels.

④ Strong authenticity: Adoption of multi-factor authentication (signature).

In Section II, we review the related work on the IBE scheme, the crypto-functions of PKG (Private Key Generator) and the complexity of crypto-functions for performance analysis. Section III addresses the SFS system architecture and its core protocols. Specifically we mention the system setup and the client registration. Section IV is concerned with how to improve our SFS WP IBE. Section V concludes the performance issue, unsolved problems of our works and its future.

II. RELATED WORK

A. IBE (Identity Based Encryption) scheme

In the IBE [7], the public key is generated from the public identity combined with an unique ID and a time

 ISBN: 978-1-61208-245-5

period which may have an effect on key revocation [1]. Boneh and Franklin proposed the WP ECC (elliptic curve cryptography) scheme among efficient pairing-based ECC [8]. The WP IBE scheme is much more efficient because its key size is much shorter and is based on the bilinearity over the Field F_p , which is the Group law to replace multiplicative Group with additive Group [8]. In spite of the shorter key size, the ECC 256-bit public-key provides the same level of security as the 3024-bit RSA-based public key [2].

Nevertheless, security flaw of the WP IBE is relevant to the public identity which is literally open to the public including adversaries. In addition, it is difficult to hide the crypto-functions like *encryption*, *decryption and key generation* from adversaries. Thus, the WP IBE must be based on the random oracle model, which prevents the adversary from successful guessing key-related things using queries to the oracle with a public identity. In order to have the scheme secured against chosen ciphertext attacks (CCA), the oracle's responses to the quesries must satisfy onewayness and randomness to hide details of the crypto-functions [17][20].

B. Weil Pairing (WP) ECC

The WP is one of the implemented ECC schemes, being formed with the Group of points over an elliptic curve. Let p be a prime and E be the elliptic curve of Weiser Strass equation $y^2 = x^3 + 1$ over the Filed F_p . The set of rational points $E(F_p) = \{(x, y) \in F_p \times F_p : (x, y) \in E\}$ forms a cyclic Group of order $p - 1$. The set of points of order q defined as $p = 12q - 1$ forms the cyclic sub-Group G_1 of which the generator is ρ . Let G_2 be the sub-Group of F_{p^2} of order q^2 [6].

The WP is based on the bilinear map $\hat{e} : G_1 \times G_1 \to G_2$ between two cyclic Groups G_1 , G_2 of order q , q^2 respectively with following properties [6]:

① *Bilinearity:* $P, Q \in G_1$ and $a, b \in \mathbb{Z}_q$, then $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$

② *Computability:* $P, Q \in G_1$, then the efficient algorithm for $\hat{e}(P, Q) \in G_2$ exists.

③ *Non-degenerate:* $\hat{e}(\rho, \rho) \in F_{p^2}^*$ is a generator of G_2 .

The WP satisfies Diffie-Hellman problem (DHP) assumption [21]: even if an eavesdropper observed g^x, g^y in the key exchange protocol, the eavesdropper cannot compute the secret key g^{xy} with easy means, while legitimate parties can do. Simultaneously, the decisional DHP(DDHP) to decide whether g^{xy} comes from g^x and g^y , given g^x, g^y and g^{xy} , is hard. But, in the WP, DDHP becomes easy with the bilinear map as follows:

$\hat{e}(g^x, g^y) = \hat{e}(xg, yg) = \hat{e}(g, xyg) = \hat{e}(g, g^{xy})$.

It is proved that DHP is equivalent to the hard discrete logarithm problem (DLP) to compute $a = Log_\rho P$ with $P(= a\rho, a \in \mathbb{Z}_p)$ and ρ [21]. Here, we insist that the WP is enough secure to protect secret values because its security is based on ECDLP (elliptic curve discrete logarithm problem) [21].

C. WP IBE crypto-functions

The IBE scheme is characterized by four randomized algorithms: *setup(), extract(), encrypt(), decrypt()* [2][9], defined as follows:

① $setup(k) \to (s, parm)$

② $key\_extract(parm, s, ID) \to d$

③ $encrypt(parm, ID, M) \to C$

④ $decrypt(parm, C, d) \to M$

Here, $k, parm, s$ are the seed value, the security parameter like a prime order, and the system wide master key $s \in \mathbb{Z}_q$ respectively. In addition, ID, d, M and C are: $ID \in \{0,1\}^*$ acting as a public identity, d as the corresponding private key, $M = \{0,1\}^n$ as a plain message and C as its ciphertext respectively.

Specifically, the WP IBE scheme produces the security parameter $parm$ summarized as follows:

$parm =< q, G_1, G_2, \hat{e}(), n, \rho, S, H_1, H_2 >$, where q is the prime order of Group, n is the message block length, the system wide public key is $S(= s\rho)$, ρ is the generator of the cyclic Group G_1 , where s is the system wide master key kept in secret on PKG (Private Key Generator). In addition, H_1 and H_2 are the hash functions with onewayness and randomness which scramble the hash algorithm so as to behave like a random oracle, defined as follows:

$H_1 : \{0,1\}^* \to G_1,\ H_2 : G_2 \to \{0,1\}^n$

The private key d_{ID} is computed from the public identity ID as follows: the public key Q_{ID} is normalized with the public identity ID as $Q_{ID} = H_1(ID) \in G_1$, and the private key is computed as $d_{ID} = sQ_{ID}$. The encryption of a message M with Q_{ID} is done as follows:

 ISBN: 978-1-61208-245-5

① Computing $g_{ID} = \hat{e}(Q_{ID}, S) \in G_2$

② Choosing a prime number $r \in Z_q^*$,

③ Generating the ciphertext like follows:

$C = < r\rho, M \oplus H_2(g_{ID}{}^r) >$

The decryption of the cyphertext C is done as follows [2][4][11]:

① $M \oplus H_2(g_{ID}{}^r) \oplus H_2(e(d_{ID}, r\rho))$

② $M \oplus H_2(g_{ID}{}^r) \oplus H_2(g_{ID}{}^r) = M$ since

$\hat{e}(d_{ID}, r\rho) = \hat{e}(sQ_{ID}, r\rho) = \hat{e}(Q_{ID}, s\rho)^r = g_{ID}{}^r$

D. WP Signature

The WP authentication process is quite different from the traditional public-key scheme. It is much faster since the signature is based on the pairing-based additive Group and its verification belongs to DDHP, as follows [6][10]:

Signer:

① Generating a temporary key pair :

$(r, R), R = r\rho, r \in \mathbb{Z}_p$

② Generating a signed digest: $T = rS + H(M, R)d_{ID}$

③ Generating the signature: $Sign = < R, T >$

Verifier:

④ Computing $v = \hat{e}(R + H(M, R)Q_{ID}, S)$,

⑤ Computing $u = \hat{e}(T, \rho)$ as

$\hat{e}(rS + H(M, R)d_{ID}, \rho) =$

$\hat{e}(rs\rho + H(M, R)sQ_{ID}, \rho) =$

$\hat{e}(r\rho + H(M, R)Q_{ID}, s\rho) =$

$\hat{e}(R + H(M, R)Q_{ID}, S) = v$

⑥ If $u \equiv v$, then it is verified.

M and $H(.)$ are the message and the hash function respectively.

E. Performance consideration.

Table 1. Basic Operation's complexity [18]

Algorithm	Input/output	Running Time
INT-DIV	*a/N (N >0)*	$O(\|a\| \cdot \|N\|)$
MOD	*a mod N (N >0)*	$O(\|a\| \cdot \|N\|)$
EXT-GCD	*a, b ((a, b)≠(0, 0))*	$O(\|a\| \cdot \|b\|)$
MOD-ADD	*a+b mod N (a, b∈Z_N)*	$O(\|N\|)$
MOD-MULT	*ab mod N (a, b∈Z_N)*	$O(\|N\|^2)$
MOD-INV	*ab=1 (mod N) (a∈Z_N)*	$O(\|N\|^2)$
MOD-EXP	*a^n mod N (a∈Z_N)*	$O(\|n\| \cdot \|N\|^2)$
EXP_G	*a^n ∈G (a∈G)*	*2|n| G-operations*

It was known that the ECC took less time to generate the key pair and digital signature than RSA, however, the only ECC signature verification is much costly (Table 2) [13][14]. In addition, *setup()* need much time to generate E points over F_p. Supposed that participants already had the security parameter produced by *setup(),* generation of the private key d_{ID} is bound to WP-time (Weil pairing time-complexity) $O(a^{(p+1)/4} \pmod p)$ of a small prime p, which is the Euclid-criteria (Section IV.B) algorithm complexity in [6].

III. SFS ARCHITECTURE AND PROTOCOL

A. Architecture

Our SFS system architecture is shown in Figure 1.

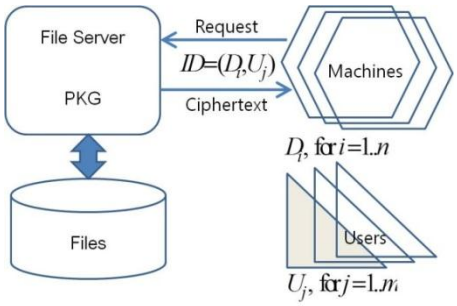


Figure 1. Architecture for Secure File Server with PKG.

As seen in Figure 1, the SFS architecture is composed of the single file server playing the role in PKG and the secret file manager, and the multiple clients of which a client means a pair of a legal user and a registered machine. A secret file can be exchanged only between the file server and a client. After completing manipulation of the secret file, the machine must eliminate the secret file from the machine, and only the file server can store the secret file.

P2P file sharing system using Gnutella protocol [19] is similar to this architecture except that the files stay at the clients registered in the network and the server keeps only file location information [19]. The architecture for DRM system is also similar to Figure 1, except that the clients are purchasers and the server is a distributor of copyrighted material.

In our SFS system, the file server is the supplier of a secret file and the clients are consumers identified by authorized user-ID U_j on registered machine-ID D_i. The public identity ID is formed by the ID pair (D_i, U_j).

B. Protocol

The SFS system takes three stages for the lifetime; the *setup stage* is first along with *initial registration* of all the machines and users, the next is the *file dissemination stage*, where transfers an encrypted file with digital signature from

the server to the client against the man-in-the-middle attack. The last is the *evolution stage* for joining and releasing users and machines in time. For the lifetime, there is the *setup stage* once initially, and the *file dissemination stage* and the *evolution stage* are repetitive.

A secret file is *created*, and then is *registered* at the file server. The file can be *distributed* on the SFS network and *modified* by the owner. This is the lifecycle of a secret file, depicted in Figure 2.

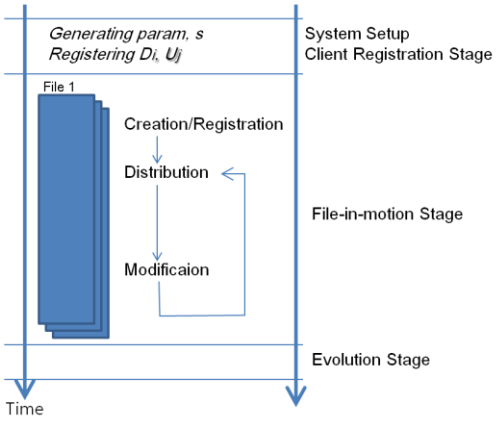


Figure 2. Life Cycle of File sharing system

Setup

The *setup(.)* generates the security parameter and the master key as follows (refer to section II):

① $param =< q, n, G_1, G_2, \hat{e}(), \rho, S, H_1, H_2 >$

② $s \in \mathbb{Z}_q$

param is distributed to authorized machines. The system wide private-key s is kept at a safe place in the server. S is the system wide public-key, ρ is the generator of the cyclic Group G_1 of order q relevant to prime number in $\mathbb{Z}_q^*$. n is the signature size. Two hash functions provide randomized hashing defined as follows:

$H_1 : \{0,1\}^* \rightarrow G_1,\ H_2 : G_2 \rightarrow \{0,1\}^n$.

Registration

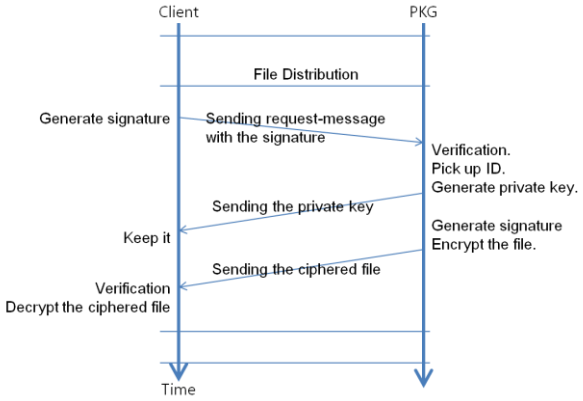
The SFS network limits the members to the machines and the users registered prior to the *file-in-motion* stage of Figure 2. Member is identified with its public identity ID_{entity}. In the *file distribution*, ID_{client} in the *request-message* is formed as combining two members' ID D_i and U_j.

① $ID_{entity} = [D_i \mid U_j]$

② $ID_{client} =< D_i \parallel U_j >$

Distribution

In the *file distribution*, two messages are exchanged between the server and the client. One is the *request-message* requesting a secret file from the server, and another is the *file-message* conveying the encrypted file to the client. Always the sender must authenticate the signature on the messages. ID_{client} is signed with a temporal private key. The sender attaches the signature on the *request-message*; refer to Figure 3.



Figurte 3. File Distribution

With ECC key pair (P, x) where $P = xQ$, $1 \le x \le n-1$, the early ECC signature scheme is [14]:

① Choosing a random number $1 \le k \le n-1$.

② Computing point $kQ = (x_1, y_1) = X$.

③ Computing $r = x_1 \pmod n$

④ Computing $k^{-1} \pmod n$

⑤ Computing $e \leftarrow SHA(m)$

⑥ Signing as $s = k^{-1}(e + xr) \pmod n$.

⑦ Generating the signature tuple $< r, s >$.

Our digital signature scheme is improved as follows:

sign()

① Choosing a prime number $r \in \mathbb{Z}_q$ as a temporal secret key.

② Computing $R = r\rho \in G_1$ as the public key.

③ Generating $ID = D_i \parallel U_j \in G_1$

④ Signing as $\Sigma_{ID} = rID \in G_1$.

⑤ Generating the signature tuple $< R, ID, \Sigma_{ID} >$.

⑥ Adding the tuple to the *request-message*

 ISBN: 978-1-61208-245-5

We remind that ECC arithmetic is based on the modular operation, and given $< R, ID, \Sigma_{ID} >$, finding r is ECDLP (Elliptic Curve Discrete Logarithm Problem).

The ECC verification is [14]:

① Computing $e \leftarrow SHA(m)$

② Computing $w = s^{-1} (\text{mod n})$

③ Computing $u_1 = ew \ (\text{mod n})$, $u_2 = rw \ (\text{mod n})$

④ Computing $X = u_1 Q + u_2 P$.

⑤ Computing point $v = x_1 \ (\text{mod } n)$.

⑥ Accepting if $r = v$

verify()

① If $\hat{e}(R, ID) \ (= \hat{e}(r\rho, ID) = \hat{e}(\rho, rID)) \equiv \hat{e}(\rho, \Sigma_{ID})$ is true, the verification is successful. Otherwise, it is failed. Here ρ is the generator of the security parameter.

Here, remember that $\hat{e}(R, xID) \neq \hat{e}(\rho, x\Sigma_{ID})$ because $xID \ (\text{mod } q) \neq ID$. Finding such x is ECDLP

IV. IMPROVEMENT OF WP IBE

The reason that our WP scheme is much better than RSA or the early ECC is that the WP is based on the Bilinearity. The IBE security is relevant to the public identity open to the public. We assume the selective-ID security [5] is satisfied in SFS, in which the target ID is specified in advance before the master public-key is published [11].

Improvement of our scheme is threefold: the selective ID normalization, the two-factor signature, and the certificate-less key revocation.

A. Public ID Normalization

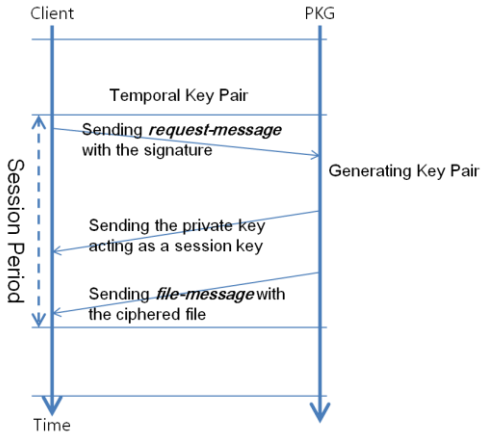


Figure 4. Session Period

In the IBE, public ID is identical for the lifetime and the adversary can obtain key pairs to guess the target key adaptively. For IBE security, we modify constant IBE public ID into temporal ID for *certificate-less key revocation.* The SFS system generates the public key after normalizing public identity ID as follows:

① Client: generating $ID_{client} = D_i \ \| \ U_j$

② PKG: normalizing the public identity ID as $ID_{client}^{time} = ID_{client} \ \| \ time_{dur}$.

③ PKG: hashing ID into $a \in \mathbb{Z}_q$, as $a = H(ID)$

The format of $time_{dur}$ is denoted as *year:mon:day:hour:min*. Wild * in $time_{dur}$ means an entire period of public identity ID. For instance if $time_{dur}$ is *year:mon:day:hour:***, the public identity ID is invalid at the next hour. This ID is called a *temporal client identity*, denoted as ID_{client}^{time}.

B. Key Generation

We mentioned that the private key generation is triggered by the *request-message* of a client (Figure 3). The key pair is similar to the session key valid for the session from the *request-message* through its *file-message* (Figure 4).

The key generation algorithm of WP ECC is generated from the well-known function *MapToPoint*, as follows [3][6][20]:

① $x \leftarrow H(ID_{client}^{time} \ \| \ j) \ (\text{mod } p)$ at $p = 12q - 1$, $j = 0$

② $a \leftarrow x^3 + 1 \ (\text{mod } p)$

③ If $a^{(p-1)/2} = 1 \ \text{mod } p$, $y \leftarrow a^{(p+1)/4} \ (\text{mod } p)$; $Q \leftarrow 12(x, y)$ -- Euler Criteria

④ Otherwise, $j = j + 1$, do the first step ①.

We modify the *MapToPoint* algorithm [6] for the key generation of the SFS system. In the SFS, the public key is denoted as Q_{ID} relevant to the client's public identity ID. The private key d_{ID} is generated as $d_{ID} = sQ_{ID}$. It is common that the private key d_{ID} is distributed to the client using secure channels like TLS or VPN. Therefore, note that there is no difficulty in SFS network since all connections in SFS are protected by the IBE based VPN.

In the performance perspective, the mean number of loops is: $E[n] = \sum_{n=1}^{\infty} \frac{n}{2^n}$ since the probability satisfying the Euler Criteria in *MapToPoint* algorithm is 1/2. Thus the loop mean $E[n]$ is approximately less than 2. Therefore, the key generation is bound to $O(E[n] \cdot a^{(p+1)/4}) = O(a^{(p+1)/4})$,

 ISBN: 978-1-61208-245-5

called "*WP-operation*" complexity [6]. However, since $p(=12q-1)$ is a small odd prime where Group G_1 is of order q, the key generation is approximate to the complexity *MOD-EXP* $O(p^2)$ (Table 1).

C. Encryption and Decryption including Authentication

If the CCA-proof encryption scheme of Canneti, Halevi and Katz [11] is considered, the SFS suggests the hybrid WP IBE combining the following schemes. The SFS system uses two sets of key pairs: $< R, r >$ for $R = r\rho$ $(r \in \mathbb{Z}_q^*)$ and $< Q_{ID}, d_{ID} >$ for $d_{ID} = sQ_{ID}$. The reason is that *Encrypt()* satisfies both integrity and authenticity as well. Let M and C be the plain-message and the cipher-message.

Encryption - Encrypt()

① $g_{ID} = \hat{e}(Q_{ID}, S) \in G_2$

② Modified message $M' =< R, M >$

③ Intermediate cipher message $C' = M' \oplus H_2(g_{ID}{}^r)$

④ Signature $\Sigma_{C'} = rH_1(C')$

⑤ Cipher message $C =< R, C', \Sigma_{C'} >$

Decryption - Decrypt()

① $C' \oplus H_2(\hat{e}(d_{ID}, R)) =$
$M' \oplus H_2(g_{ID}{}^r) \oplus H_2(\hat{e}(d_{ID}, R)) =$
$M' \oplus H_2(g_{ID}{}^r) \oplus H_2(g_{ID}{}^r) = M' < R', M >$ since
$\hat{e}(d_{ID}, R) = \hat{e}(sQ_{ID}, r\rho) = \hat{e}(Q_{ID}, s\rho)^r = \hat{e}(Q_{ID}, S)^r$
$= g_{ID}{}^r$

② If verification $\hat{e}(\Sigma_{C'}, \rho) \equiv \hat{e}(H_1(C'), R)$ is true, C is clean.

③ If $R' = R$ in $M' =< R', M >$, it means that the message integrity is verified and the plain message M is restored.

The *encryption* and *decryption* satisfy confidentiality, integrity, and authenticity as well. However, adversary cannot obtain a hint on any relation between C and M in CCA [16] since randomized hash is applied to C and M. In the performance perspective, this encryption and decryption are bound to $O(g_{ID}{}^r)$, EXP_G, r *WP-operation* (Table 1). $O(g_{ID}{}^r)$ is $O(g_{ID}{}^r \pmod p) = O(rp^2)$. Since $1 \le r \le p$, therefore, the encryption scheme is bound to $O(p^3)$.

V. PERFORMANCE

The early ECC is known to be better than RSA in terms of key size, signature and encryption. Of course the verification is worse than RSA (Table 2) [13][14].

Table 2. Comparison of the early ECC and RSA [13].

Key Length		Key Generation		Signature		Verification	
ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
163	1024	0.08	0.16	0.15	0.01	0.23	0.01
233	2240	0.18	7.47	0.34	0.15	0.51	0.01
283	3072	0.27	9.80	0.59	0.21	0.86	0.01
409	7680	0.64	133.9	1.18	1.53	1.80	0.01
571	15360	1.44	679.06	3.07	9.20	4.53	0.03

Among ECCs, the WP is known to be more efficient with respect to security and performance. Comparing with RSA, the ECC key length 163-bit is the same level of security as RSA's 1024-bit [6][13][14]. In Table 2, we know that the early ECC is better than RSA, except the verification. The early ECC is compared with our SFS WP scheme in Table 3. Our improved WP scheme is much better.

In the security level, therefore, the WP IBE turned out to be the better public-key cryptography than the early ECC [2].

Table 3. Comparison of SFS WP and ECC

Operation	Our WP		ECC	
	sign	verify	sign	verify
MOD-EXP			1	1
MOD-MULT	2	2	4	4
MOD-ADD			2	2
Hash()			1	1
total	2	2	8	8

We note that the signature scheme of our SFS WP is similar to the verification one in term of the number of operations (Table 3). In the ECC, it is known that the signature takes more time than the encryption in term of the running time. Therefore, we expect that our SFS WP is improved in the perspective of performance.

VI. CONCLUSION AND FUTURE WORK

In the future, we will make a real time test-bed to assess their performance. We think that the WP is difficult to substitute for the conventional public-key scheme in the crypto-process structure. The IBE hash function is assumed to satisfy both onewayness and randomness, however, it is not straightforward to implement such hash function. The IBE random oracle model is not practical and the model is speculative for security proof.

As the pairing-based ECC limits keys over the field, it surprisingly reduces key computation time. Nevertheless, adaptive attacks like IND-ID-CCA [16] are feasible, however, the WP neglects such attacks. In future work, we will improve the signature and the encryption collaboratively in a simple fashion.

REFERENCES

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," Advances in Cryptology, Crypto '84, Lecture Notes in Computer Science, vol. 196, pp. 47-53, Springer-Verlag, 1984.

 ISBN: 978-1-61208-245-5

[2] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” Advances in Cryptology - Crypto'01, LNCS 2139, pp. 213 - 229, Springer-Verlag, 2001

[3] V. S. Miller and A. K. Lenstra, “Weil Pairing and Its Efficient Calculation,” J. Cryptology (2004) 17: pp. 235–261, 2004.

[4] J. Callas, “Identity-based Encryption with Conventional Public-Key Infrastructure,” PGP Corporation Palo Alto, California, USA, jon@pgp.com.

[5] C. Gentry and A. Silverberg, “Hierarchical ID-Based Cryptography,” Proceedings of ASIACRYPT ’02, LNCS 2501, pp. 548-566, as <http://eprint.iacr.org/2002/056/>.

[6] X. Yi, “An Identity-Based Signature Scheme From the Weil Pairing,” IEEE Communication Letters, vol. 7, no. 2, Feb. 2003.

[7] M. Burmester and Y. Desmedt, “Identity-based key infrastructures,” Proceedings of the IFIP TC11 19th International Information Security Conference (SEC 2004), pp. 167–176, Kluwer, August 2004.

[8] A. Menezes, “An Introduction to Pairing-Based Cryptography,” Contemporary Mathematics, vol. 477, 2009.

[9] Y. Zheng, “Improved public key cryptosystems secure against chosen ciphertext attacks,” Technical Note, University of Wollongong, 1994.

[10] E. Fujisaki and T. Okamoto, “Secure integration of asymmetric and symmetric encryption schemes,” in Advances in Cryptology - Crypto'99, LNCS 1666, pp. 537-554, Springer-Verlag, 1999.

[11] R. Canneti, S. Halevi and J. Katz, “Chosen-ciphertext security from identity-based encryption,” Proc. Of Eurocrypt’04, LNCS 3027, pp. 207-222, Springer-Verlag, May 2–6, 2004, Interlaken, Switzerland.

[12] R. Lu and Z. Cao, “ID-based Encryption Scheme Secure against Chosen Ciphertext Attacks,” Cryptology ePrint Archive http://eprint.iacr.org/2005/355.

[13] N. Jansma and B. Arrendondo, “Performance Comparison of Elliptic Curve and RSA Digital Signatures,” nicj.net/files, Apr. 28, 2004.

[14] K. Gupta and S. Silakari, “ECC over RSA for Asymmetric Encryption: A Review,” International Journal of Computer Science Issues, vol. 8, issue 3, no. 2, pp. 370-375, May 2012.

[15] C. Gentry, “Certificate-Based Encryption and the Certificate Revocation Problem,” Proceedings of EUROCRYPT’03, LNCS 2656, pp.272-293, May 2003, Varsaw, Poland

[16] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, “Relations among notions of security for public-key encryption schemes,” CRYPTO '98, volume 1462 of LNCS, pp. 26-45, Aug. 23-27, 1998, Santa Barbara, USA.

[17] M. Bellare and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” First ACM Conference on Computer and Communications Security, ACM, 1993.

[18] M. Bellare and P. Rogaway, “Introduction to Modern Cryptography,” Sep. 21, 2005, citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124, [retrieved: Sep. 2012].

[19] D. Bickson and D. Malkhi, “A Study of Privacy in File Sharing Networks,” 2004, http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.7.3157, [retrieved: Sep. 2012].

[20] P. Yang, T. Kitagawa, G. Hanaoka, R. Zhang, K. Matsuura, and H. Imai, “Applying Fujisaki-Okamoto to Identity-Based Encryption,” Lecture Notes in Computer Science, 2006, Volume 3857/2006, pp. 183-192, DOI: 10.1007/11617983_18.

[21] A. Joux and K. Nguyen, “Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups,” (2001), http://eprint.iacr.org/2001/003.ps.gz, [retrieved: Sep. 2012].

[22] K. Park, H. Hwang, C. Lee, and S. Min, “DRM Technology Status and Contents Distribution Infrastructure Construction,” Journal of KIISE, vol. 23, no. 8, pp. 8-14, Sep. 2005. (in Korean).

 ISBN: 978-1-61208-245-5