International Journal of Innovative
Computing, Information and Control
Volume 10, Number 2, April 2014

ICIC International ©2014 ISSN 1349-4198
pp. 535–544

AN IMPROVED FAIL-STOP SIGNATURE SCHEME BASED ON DUAL COMPLEXITIES

Kai Chain[1], Jonathan Jen-Rong Chen[2], Jar-Ferr Yang[1] and Kuei Hu Chang[3,*]

[1]Institute of Computer and Communication Engineering
Department of Electrical Engineering
National Cheng Kung University
No. 1, University Rd., Tainan 701, Taiwan

[2]Department of Information Management
Vanung University
No. 1, Van-Nung Rd., Chungli, Taoyuan 32061, Taiwan

[3]Department of Management Sciences
R.O.C. Military Academy
No. 1, Wei-Wu Rd., Fengshan Dist., Kaohsiung 830, Taiwan
*Corresponding author: evenken2002@yahoo.com.tw

Received February 2013; revised June 2013

Abstract. *The basic design supposition for digital signatures in the cryptology domain is that the attacking and victimized computers have comparable resources. The operation of electronic commerce is based on this assumption, but the advent of accumulated networked resources and the changing computing landscape have elevated this risk. However, if an attacker has powerful computing capabilities compared with the victim, the attacker will, in given time, crack his password and gain the ability to fraudulently use the victim's identity. To avoid this threat, this study presents a plan that is based on the complexity of the fail-stop signature (FSS) scheme and the discrete logarithm and factorization of 2 mathematical problems of the digital signature algorithm. The scheme can be implemented in e-commerce information security environments and provides the user with the possibility of preventing attacks and enhancing system safety. This fail-stop scheme can assert a victim's innocence without exposing the $n = p \times q$ secret and guards against malicious behavior.*

Keywords: Digital signature, Fail-stop signature scheme, Dual complexities, E-commerce, Cryptology

1. **Introduction.** The application of modern cryptology is pervasive in such areas as the military, business, science, and technology. For example, in electronic commerce, digital signatures are used in business for contracts and acquisitions, commercial trade, transactions, and document transmissions. They are also applied in business models for network banking and online shopping. Thus, cryptology is a cornerstone of technology. Today, the rise in network applications, accompanied by rampant Internet crime, has increased the value of cryptology. A longstanding tenet of cryptology states that the strength of a password is based on the time that it can withstand guesses. When the resources of an attacker and victim are comparable, a longer password increases the calculation time to bypass [1].

The basis of traditional digital signatures generally assumes that the attacker and victim have the same level of computing resources. In reality, the popularity of maliciously applied distributed computing (e.g., botnets) has given attackers access to many more resources than electronic commerce environments, allowing resource-rich malicious groups to attack and gain access to a system and masquerade as legitimate users for illicit financial or commercial gain. This lapse in security affects the victim's finances, causes irreparable damage to his reputation with regard to credit rating, and affects the overall system that was attacked. The impact and loss of the attack are extensive and difficult to estimate [11,12]. Currently, a victim must prove his innocence, and the system's owner must ensure its security and that user rights are unaffected – only then can a business resume normal activities. To protect against this type of attack, fail-stop signature (FSS) has been proposed [16]. An FSS protects a signer against a forger with even unlimited computational power, because the likelihood of determining the signer's private key in the FSS is negligible [25]. The research on FSS has been extensive [3,6,10,19,20-26].

This study focused on FSS schemes in which the underlying issue is related to problems regarding integer factorization and the discrete logarithm. In [3], FSS schemes existed only if the computing discrete logarithms or factoring large integers were hard. In [17], an efficient FSS scheme was proposed to protect clients in an online payment system. In [27], an efficient FSS scheme based on discrete logarithm is presented. In [28], FSS schemes using schemes "bundling homomorphism" is proposed. In [23], Susilo et al. proposed a new and efficient FSS scheme. In 2004, Schmidt-Samoa proposed an improvement of Susilo et al.'s work [23] based on the difficulty of factorization [21]. This method can prove the innocence of the victims, but also expose the secret $n = p \times q$ requiring the whole system to rebuild or replace the system parameters in order to continue operating properly and securely [2,23]. In this report, we developed a method to prove a victim's innocence while safeguarding the $n = p \times q$ secret. In addition, this method can also thwart denial-of-service attacks. To this end, we propose a plan that is based on the complexity of the fail-stop scheme, which is built on a solution of the discrete logarithm and factorization problems in the digital signature algorithm.

The paper is organized as follows. The background of digital signature schemes is introduced in Section 2. Section 3 overviews the FSS. Section 4 presents a novel FSS scheme and demonstrates that it is an instance of the general construction. Section 5 provides a complete proof and analysis of the scheme's security. In Section 6, the corresponding computation of this scheme is discussed, and we compare our scheme and existing schemes. Finally, Section 7 concludes the paper.

2. Digital Signatures Based on One Assumption.

The digital signature scheme is one of the most important technological applications in modern cryptography and information security. After many years of evolution, digital signature technologies have matured and been used widely in electronic commerce. Digital signature algorithms are categorized according to their secure assumptions: One group comprises digital signature schemes that are based on discrete logarithm problems, and the other group consists of digital signatures that are based on the factorization problem. The chief characteristics of digital signatures are as follows [4,5,7,8,13]:

(1) Authenticity: Determining the source of legality of the information, i.e., that the information has been sent by the sender rather than a forgery or recycled old messages.
(2) Integrity: Ensuring that the information has not been altered intentionally or unintentionally or replaced with new or deleted text.
(3) Nonrepudiation: After sending messages, the sender is undeniable that information of transference.

2.1. Digital signature based on discrete logarithm.

The earliest digital signature scheme that was based on the discrete logarithm was proposed by El Gamal [9] in 1985.

The detailed scheme is described as follows [9].

ElGamal Signature Scheme

• *Key Generation Phase*

(1) The signer B chooses a large prime number p and a number g such that g is a primitive element of $GF(p)$. Then, the signer B publishes the 2 numbers p and g.

(2) Signer's Keys:

(a) Private Keys: $x \in Z_p^*$

(b) Public Keys: $y \equiv g^x \bmod p$

• *Signature Generation Phase*

Input m ($1 \leqq m \leqq p-1$), which is the message that is to be signed.

(1) The signer B chooses a random integer k with $\gcd(k, p-1) = 1$.

(2) Then, $r \equiv g^k \bmod p$ is computed, where $r \in (1, p)$.

(3) The signer B computes s such that $m \equiv xr + ks \bmod p-1$ (or $s \equiv k^{-1}(m - xr) \bmod p-1$).

Return (r, s), which is the signature for the message m that is signed by the signer B.

• *Signature Verification Phase*

To verify that (r, s) is a valid signature of the message m, the verifier A can check the congruence $g^m \equiv y^r \cdot r^s \bmod p$. If it holds, then (r, s) is a valid signature of the message m, and the scheme returns 1. Otherwise, it returns 0. Note that $g^m \equiv g^{xr+ks} \equiv g^{xr} \cdot g^{ks} \equiv (g^x)^r \cdot \left(g^k\right)^s \equiv y^r \cdot r^s \bmod p$.

• *Cryptanalysis*

(1) The ElGamal signature scheme claims that its security is based on the discrete logarithm problem. If this problem can be solved trivially, the attacker C will compute the private key x by y and g. Then, the signature scheme is broken.

(2) If the attacker C wants to forge a legal signature, he chooses r (or s) and calculates s (or r) to comply with the equation $g^m \equiv y^r \cdot r^s \bmod p$. The attacker C will encounter the discrete logarithm problem.

(3) If the attacker C has obtained the signature of plaintext m and (r, s), he will want to calculate x from Equation (1). However, Equation (1) has 2 unknown values, x and k; thus, C is unable to calculate x.

(4) The attacker C can forge a legal signature (r, s) from message m, but m cannot be fixed in advance.

2.2. **Digital signature based on factoring.** The earliest digital signature scheme that was based on the factoring problem was proposed by Rivest et al. [18] in 1978. The RSA scheme can be used in public key encryption and digital signatures. The security of the RSA signature scheme is based on the problem of solving the factoring of large numbers [7]. The detailed scheme is described as follows.

RSA Signature Scheme [7]

• *Key Setup Phase*

The key setup procedure is the same as that for RSA cryptosystems.

(1) The signer B computes $n = pq$, with p and q being 2 large roughly equal prime numbers size.

(2) The signer B randomly chooses an integer e such that $\gcd(e, \o(n)) = 1$, where $\o(n) = (p-1)(q-1)$.

(3) The signer B finds an integer d such that $ed \equiv 1 \bmod \o(n)$ (i.e., $d \equiv e^{-1} \bmod \o(n)$).

[Note: Sometimes, we let $d \equiv e^{-1} \bmod \operatorname{lcm}(p-1, q-1)$.]

(4) The signer B's keys:

(a) Private Keys: d, p, and q.

(b) Public Keys: e and n.

- *Signature Generation Phase*

(1) Input m, which is the message that is to be signed.

(2) To create a signature of message m, the signer B computes the value S such that $S \equiv m^d \bmod n$.

- *Signature Verification Phase*

To verify that S is a valid signature of the message m, the verifier A can simply check the following congruence: $S^e \equiv m \bmod n$. If it holds, then S is a valid signature of the message m.

- *Cryptanalysis*

The security of RSA is based on the difficulty of the factorization.

3. **Preliminaries.** In this section, we briefly review the theory and requirements of FSS and refer the reader to [6,10,21,23,24] for a more complete account.

3.1. **Notations.** The length of a number n is a positive integer, and $|n|\,2$ denotes the bit-length of n. $p|\,q$ means p divides q. Z_n means the ring of integers modulo a number n. Z_n^* is Z_n's multiplicative group, which includes only the integers that are relatively prime to n.

3.2. **Review of FSS schemes.**

Prekey generation

A recipient, C, chooses 2 large safe prime numbers p and q. Then, C finds a large prime p_1, such that a factor of p_1-1 is the product of 2 large primes p and q, i.e., $n|p_1 - 1$ and $n = pq$. Finally, C selects an element g whose order modulo p_1 is p that satisfies:

$$g^{\frac{1}{2}p} \equiv -1(\text{mod } p_1) \tag{1}$$

The public and secret keys of the trusted center are given by (p_1, g, n) and (p, q), respectively [4,15].

Key Generation

The signer A chooses 2 integers x_1, $x_2 \in z_n$ and calculates:

$$y_i \equiv g^{x_i}(\text{mod } p_1), \quad 1 \le i \le 2 \tag{2}$$

The signer A uses $\{y_1, y_2\}$ in a trusted center. Thus, the public key is (y_i), and the private key is (x_i) from $1 \le i \le 2$.

Algorithm for signing a message $\boldsymbol{m}$

Suppose the signer A wants to sign a message m to receiver B. A computes:

$$m_1 \equiv mx_1 + x_2(\text{mod } n) \tag{3}$$

Then, the signer A produces $\{m_1\}$ as a signature of message m.

Algorithm for verifying the signature

The receiver B confirms the validity of the signature $\{m_1\}$ by testing whether the following equation holds:

$$g^{m_1} \equiv y_1^m y_2(\text{mod } p_1) \tag{4}$$

If the algorithm that generates the parameters, keys, and signing messages is successful, then the confirmation of the signature in the signature verification algorithm is the same.

Proof of Forgery

Assume that receiver B uses the signature $\{m_2\}$, which is an acceptable signature on m that signer A wants to forge. To do so, signer A calculates his own signature $m_1 \equiv mx_1 + x_2(\text{mod } n)$ and $GCD(m, -m_2, n)$, and $GCD(a_1, a_2)$. $GCD(a_1, a_2)$ means that two numbers a_1 and a_2 of the greatest common factor. Then, the composite number n could be factorized by the signer A. Therefore, the signature $\{m_2\}$ is proof of forgery.

3.3. **Schmidt-Samoa attack.** Schmidt-Samoa proposed an attack mode in 2004 [21] as follows. Assume that an attacker E, who received signer A's signature, and per the method of producing $\{m, m_1\}$, chooses an integer $x_1' \in z_n$ and calculates:

$$y_1 \equiv g^{x_1'} (\text{mod } p_1) \tag{5}$$

and E chooses another integer x_2' that satisfies:

$$m_1 \equiv m x_1' + x_2' (\text{mod } n) \tag{6}$$

Then, E selects an integer $t \in z_n^*$ and calculates:

$$s_0 \equiv (m_1 + tp) x_1' + x_2' (\text{mod } p) \tag{7}$$

$$s_0 \equiv (m_1 + tp) x_1' + x_2' (\text{mod } q) \tag{8}$$

Using the Chinese remainder theorem (CRT), m_0 can be calculated, and the attacker E can send the forged messages: $m_0 \equiv (m_1 + tp(\text{mod } n))$. In addition, the attacker E can send the same digital signature s_0 with signer A. To resolve these weaknesses of Susilo et al.'s scheme [23], Schmidt-Samoa proposed another model, in which $n = p^2 q$. If the reader is interested, specifics are provided in [21].

4. **The Proposed Scheme.**

In this section, we introduce a novel fail-stop scheme that is based on a discrete logarithm and factorization difficulties and also show that it is an instance of the general construction. In this scheme, the recipient C generates public and secret keys, as in the previous section. In addition, C selects an element g_1 whose order modulo p_1 is n that satisfies:

$$g_1^{\frac{n}{2}} \equiv -1 (\text{mod } p_1) \tag{9}$$

(g_1) also is a public key.

Key Generation

This step is the same as above. The signer A chooses 2 integers x_1, $x_2 \in z_n$ and calculates:

$$y_i \equiv g^{x_i} (\text{mod } p_1), \quad 1 \le i \le 2 \tag{10}$$

Signer A uses $\{y_1, y_2\}$ in a trusted center. Thus, the public key is (y_i) and the private key is (x_i) from $1 \le i \le 2$.

Algorithm for signing a message m

Suppose the signer A wants to sign a message m to receiver B. The calculations are as follows:

(1) Calculations

$$a \equiv (m) x_1 + x_2 (\text{mod } n) \tag{11}$$

$$s_1 \equiv g^a (\text{mod } p_1) \tag{12}$$

$$s_2 \equiv g_1^a (\text{mod } p_1) \tag{13}$$

(2) The signer A chooses 3 integers $k_i \in z_m^*$, $1 \le i \le 3$ and calculates:

$$r_1 \equiv g^{k_1} (\text{mod } p_1) \tag{14}$$

$$r_2 \equiv g_1^{k_2} (\text{mod } p_1) \tag{15}$$

$$s_1 \equiv a r_1 + k_1 b_1 (\text{mod } n) \tag{16}$$

$$s_2 \equiv a r_2 + k_2 b_2 (\text{mod } n) \tag{17}$$

$$r_2 s_1 \equiv a r_1 r_2 + k_1 b_1 r_2 (\text{mod } n) \tag{18}$$

$$r_1 s_2 \equiv a r_1 r_2 + k_2 b_2 r_1 (\text{mod } n) \tag{19}$$

Let

$$r_2 s_1 + r_1 s_2 \equiv s_3 (\text{mod } n) \tag{20}$$

$$s_3 \equiv a(2r_1 r_2) + (k_1 b_1 r_2 + k_2 b_2 r)(\text{mod } n) \tag{21}$$

(3) Then, signer A sends $\{r_i, b_i, s_j\}$ to receiver B ($1 \le i \le 3$, $1 \le j \le 2$).

Algorithm for verifying the signature

B receives $\{r_i, b_i, s_j\}$ and then tests the following equations to determine whether they hold:

$$s_1 \equiv y_1^{(m^2+1)} y_2 (\text{mod } p_1) \tag{22}$$

$$g^{s_1} \equiv s_1^{r_1} r_1^{b_1} (\text{mod } p_1) \tag{23}$$

$$g_1^{s_2} \equiv s_2^{r_2} r_2^{b_2} (\text{mod } p_1) \tag{24}$$

If the equations above are established, the message is accepted; otherwise, it is rejected.

Proof of Forgery

Assume that receiver B uses the message $\{r'_i, b'_i, s'_j\}$ ($1 \le i \le 3$, $1 \le j \le 2$), which is an acceptable signature on m that signer A wants to forge. Therefore, signer A calculates the steps of the signature stage, and receiver B calculates the steps of the verification stage. Between both probabilities is $(1 - q^{-1})$, and $s_1 \ne s'_2 (\text{mod } n)$; thus, the innocent signer A can be restored. The operational processes are shown in Figure 1 and Figure 2.

Figure 1 shows that receiver B uses Equation (4) to verify the message m in the traditional proof-of-forgery phase. Figure 2 shows receiver B using Equations (22)-(24) to verify the message m in the new proof-of-forgery phase.

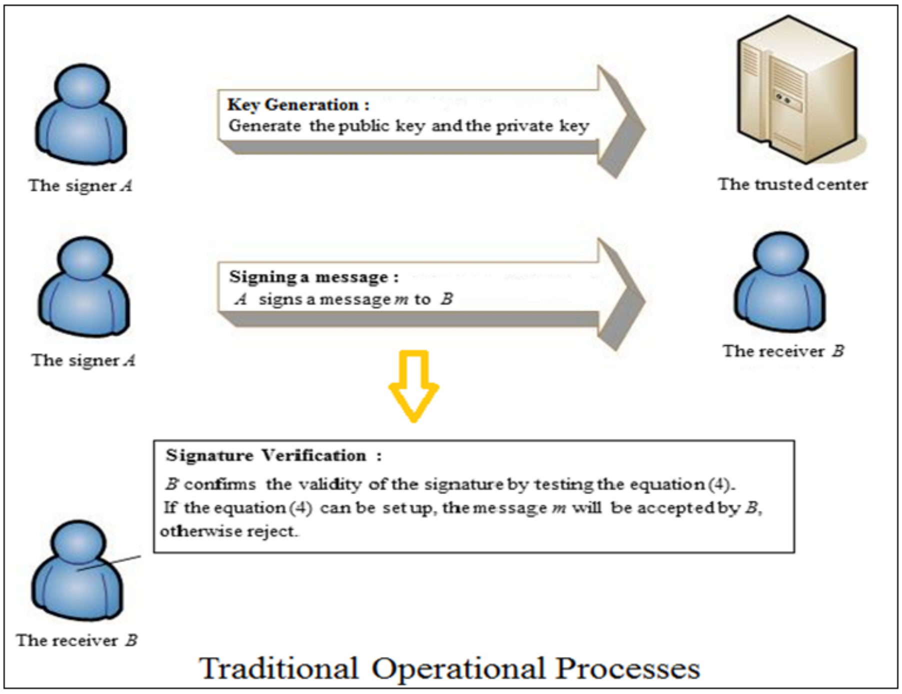

FIGURE 1. Traditional operational processes

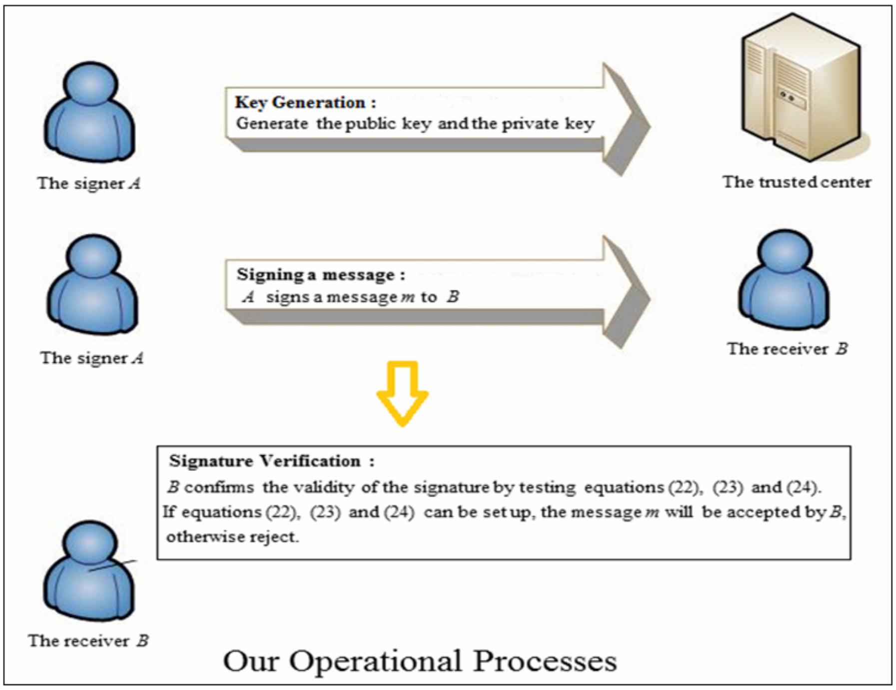

FIGURE 2. The proposed scheme's operational processes

5. Proof of Security.

Lemma 5.1. *$a \equiv b(\text{mod } m)$ and $d|m$, $d > 0$. $\Rightarrow a \equiv b(\text{mod } d)$.*

We usually apply the concept of congruencies, which is a special type of relation in cryptography, instead of equality. The definition of congruence is as follows. Please refer to the proof (5) of Definition 5.1 [15].

Definition 5.1. [15] *Let a, b, c, d denote integers. Then:*
(1) $a \equiv b(\text{mod } m)$, $b \equiv a(\text{mod } m)$ and $a - b \equiv 0(\text{mod } m)$ are equivalent statements.
(2) If $a \equiv b(\text{mod } m)$ and $b \equiv c(\text{mod } m)$, then $a \equiv c(\text{mod } m)$.
(3) If $a \equiv b(\text{mod } m)$ and $c \equiv d(\text{mod } m)$, then $a + c \equiv b + d(\text{mod } m)$.
(4) If $a \equiv b(\text{mod } m)$ and $c \equiv d(\text{mod } m)$, then $ac \equiv bd(\text{mod } m)$.
(5) If $a \equiv b(\text{mod } m)$ and $d\,|m$, $d > 0$, then $a \equiv b(\text{mod } d)$.
(6) If $a \equiv b(\text{mod } m)$ then $ac \equiv bc(\text{mod } mc)$ for $c > 0$.

Lemma 5.2. *Assume that there are 3 integers $w_l \in z_m^*$ $(1 \leq l \leq 3)$, where the numbers of w_i and w_j are known and the number of w_k is unknown and satisfies the following equations:*

$$w_i \neq w_j(\text{mod } n) \tag{25}$$

$$t_1 \equiv g^{w_1}(\text{mod } p_1) \tag{26}$$

$$t_2 \equiv g_1^{w_2}(\text{mod } p_1) \tag{27}$$

$$t_1t_2 \equiv (gg_1)^{w_3}(\text{mod } p_1) \tag{28}$$

$$\Rightarrow (1)\ w_k \neq w_i(\text{mod } n) \tag{29}$$

$$(2)\ w_k \neq w_j(\text{mod } n) \tag{30}$$

(3) To solve the complexity that w_k is equal to is at least solving the discrete logarithm problem with the same complexity.

Proof:
Case (1)
For $k = 3$, w_3 is unknown and w_1 and w_2 are known numbers. Thus, we have the following relationships:

$$t_1 \equiv g^{w_1} (\text{mod } p)$$
$$t_2 \equiv g_1^{w_2} (\text{mod } p)$$
$$t_1 t_2 \equiv (g g_1)^{w_3} (\text{mod } p)$$

The 3 equations above are based on Equations (26)-(28), and Lemma 5.1. If Proof (1) is invalid, i.e., for $i = 1$, $w_3 = w_1 (\text{mod } n)$. For $i = 2$, the discussion is similar and is omitted.

Then,

$$\begin{aligned} t_1 t_2 &\equiv (g g_1)^{w_1} (\text{mod } p) \\ &\equiv g_1^{w_1} g_2^{w_2} (\text{mod } p) \\ &\neq t_1 g_2^{w_2} (\text{mod } p) \end{aligned}$$

on the basis of Equations (25) and (26); thus, $t_1 t_2 \neq t_1 t_2 (\text{mod } p)$, based on Equation (27).

By the law of contradiction [15], Proof (1) is valid. For the same reason, Proof (2) can be obtained. Proof (3) is a discrete logarithm problem. The attacker has to solve the factorization problem of the composite number [14,19]. Thus, the proposed scheme's conclusion is valid. The proofs of Cases (2) and (3) for $k = 2$ and $k = 1$, respectively, are similar and omitted.

Lemma 5.3. *Equation (23) is true.*

Proof: $g^{s_1} \equiv g^{a r_1} g^{k_1 b_1} (\text{mod } p_1)$, based on Equation (16), and Lemma 5.1, $g^{s_1} \equiv s_1^{r_1} r_1^{b_1} (\text{mod } p_1)$, based on Equations (12) and (14). Thus, this lemma has been proved.

The proof of Equations (22) and (24) is similar to that of Equation (23) and is omitted.

6. Discussion.

Theorem 6.1. *The probability of $s_2 \neq s_2' (\text{mod } n)$ is $(1 - \frac{1}{q})$.*

Proof: From Lemmas 5.2 and 5.3, we know that signer A has the same value a in signing a message m and that attacker E has the same value a' in signing a message m. From [23], we know that $a' \equiv a + lp (\text{mod } n)$, $(0 \leq l < q - 1)$ and that the probability of $a' \equiv a (\text{mod } n)$ is $\frac{1}{q}$. According to Equation (13), the probability of s_2 is equal to s_2' and is $\frac{1}{q}$. Thus, Theorem 6.1 is proven.

From the discussion above, in [6,10,21,23,24], there are proofs that the algorithms are secure for the signer. However, assuming that attacker E intercepts s_2 from signer A, based on Equation (13) in the proof-of-forgery stage, the attacker E can calculate the value of a. If the signer A does not change the private key $\{(x_1) \text{ or } (x_2)\}$ or public key $\{(y_1) \text{ or } (y_2)\}$ after the proof-of-forgery stage, then A is going to send message m to the receiver B and calculates:

$$a_1 \equiv (m) x_1 + x_2 (\text{mod } n) \tag{31}$$

$$s_2''' \equiv g_1^{a_1} (\text{mod } p_1) \tag{32}$$

From Equations (31), (32), and (11), the attacker E can calculate the values of $\{x_1, x_2\}$. Then, attacker E can intercept the correlation data from Equation (3) in [21,23] and send any forged message m_2 to any receiver B′. In the future, signer A will be unable to establish his innocence. Establishing a situation in which signer A does not need to replace his private and public keys in the proof-of-forgery stage after restoring his innocence is the chief proposal of this paper.

A comparison

Table 1 compares the 3 FSS schemes. Due to the interactions between parameters, a general evaluation was difficult to perform. To explain the computational complexity, we define certain operation symbols as follows:

σ: related to the signer's security
k: related to the recipient's security
K: $\max(k, \sigma)$
Ḱ: Ḱ $\approx$ 2K

TABLE 1. Comparison of computational and efficiency parameters

	Susilo et al.'s scheme [23]	Schmidt-Samoa's scheme [21]	Proposed Scheme
PK (mult)	4K	k	4K
Sign (mult)	1	$\rho = \max(\sigma, k/3)$	2
Test (mult)	4K	$4\rho = \max(4\sigma, 4k/3)$	3K
Length of PK	2	$6\rho = \max(6\sigma, 2k)$	2
Length of SK	4K	$6\rho = \max(6\sigma, 2k)$	4K
Length of a signature	2K	$3\rho = \max(3\sigma, k)$	2K
Underlying hard problem	DL & Factoring	Factoring	DL & Factoring

Although, the proposed scheme performs as well as the FSS scheme of [23], the security of our scheme is higher.

7. **Conclusion.** We have proposed a novel plan, based on the complexity of the fail-stop scheme, which is built on a solution to the discrete logarithm and factorization problems in digital signature algorithms. This fail-stop scheme will not expose the $n = p \times q$ secret and proves the victim's innocence, guarding against malicious behavior and denial-of-service attacks. In the networked space, electronic commerce activities are frequent, for which existing protection mechanisms must be improved and secure environments established. The proposed scheme provides a degree of support in maintaining signatures for e-commerce transactions.

Acknowledgements. The authors would like to thank the National Science Council of Taiwan, for financially supporting this research under Contract No. NSC 100-2410-H-145-001, NSC 101-2410-H-145-001, and NSC 102-2623-E-145-001-D.

REFERENCES

[1] M. Abdalla, X. Boyen, C. Chevalier and D. Pointcheval, Distributed public-key cryptography from weak secrets, *Proc. of the 12th International Conference on Theory and Practice in Public Key Cryptography*, Irvine, pp.139-159, 2009.

[2] N. Baric and B. Pfitzmann, Collision-free accumulators and fail-stop signature schemes without trees, *Proc. of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques*, Konstanz, Germany, pp.480-494, 1997.

[3] G. Bleumer, B. Pfitzmann and M. Waidner, A remark on signature scheme where forgery can be proved, *Lecture Notes in Computer Science*, vol.473, pp.441-445, 1991.

[4] J. J. R. Chen and Y. C. Liu, A traceable group signature scheme, *Mathematical and Computer Modelling*, vol.31, nos.2-3, pp.147-160, 2000.

[5] J. J. R. Chen, A. P. Chen and R. W. M. Lin, A novel blind signature scheme possessed with dual protections, *Proc. of the 37th Annual International Carnahan Conference on Security Technology*, Taiwan, pp.123-127, 2003.
[6] X. F. Chen, F. G. Zhang, W. Susilo and Y. Mu, Efficient generic on-line/off-line signatures without key exposure, *Lecture Notes in Computer Science*, vol.4521, pp.18-30, 2007.
[7] S. Y. Chiou, *The Design and Analysis of Digital Signatures Based on Factoring and Discrete Logarithm Problems*, National Cheng Kung University Department of Electrical Engineering, Tainan, 2004.
[8] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol.22, no.6, pp.644-654, 1976.
[9] T. El Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, vol.31, no.4, pp.469-472, 1985.
[10] X. Hu and S. Huang, Comment fail-stop blind signature scheme design based on pairings, *Wuhan University Journal of Natural Sciences*, vol.6, pp.1545-1548, 2006.
[11] Y. Ishai, J. Katz, E. Kushilevitz, Y. Lindell and E. Petrank, On achieving the "best of both worlds" in secure multiparty computation, *SIAM Journal on Computing*, vol.40, no.1, pp.122-141, 2011.
[12] J. Katz, R. Ostrovsky and M. Yung, Efficient and secure authenticated key exchange using weak passwords, *Journal of the ACM*, vol.57, no.1, pp.1-39, 2009.
[13] C. S. Laih and W. C. Kuo, New signature schemes based on factoring and discrete logarithms, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol.E80-A, no.1, pp.46-53, 1997.
[14] K. S. McCurley, A key distribution system equivalent to factoring, *Journal of Cryptology*, vol.1, no.2, pp.95-105, 1988.
[15] I. Niven, H. S. Zuckerman and H. L. Montgomery, *An Introduction to the Theory of Numbers*, John Wiley and Sons, 1991.
[16] T. P. Pedersen and B. Pfitzmann, Fail-stop signatures, *SIAM Journal on Computing*, vol.26, no.2, pp.291-330, 1997.
[17] B. Pfitzmann, Fail-stop signatures: Principles and applications, *Proc. of the 8th World Conference on Computer Security, Audit and Control*, Oxford, pp.125-134, 1991.
[18] R. L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, vol.21, no.2, pp.120-126, 1978.
[19] R. Safavi-Naini and W. Susilo, A general construction for fail-stop signature using authentication codes, *Cryptography and Computational Number Theory*, vol.20, pp.343-356, 2001.
[20] R. Safavi-Naini, W. Susilo and H. X. Wang, An efficient construction for fail-stop signature for long messages, *Journal of Information Science and Engineering*, vol.17, no.6, pp.879-897, 2001.
[21] K. Schmidt-Samoa, Factorization-based fail-stop signatures revisited, *Lecture Notes in Computer Science*, vol.3269, pp.118-131, 2004.
[22] V. Susilo, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, 2005.
[23] W. Susilo, R. Safavi-Naini, M. Gysin and J. Seberry, A new and efficient fail-stop signature scheme, *Computer Journal*, vol.43, no.5, pp.430-437, 2000.
[24] W. Susilo and Y. Mu, Provably secure fail-stop signature schemes based on RSA, *International Journal of Wireless and Mobile Computing*, vol.1, no.1, pp.53-60, 2005.
[25] W. Susilo, R. Safavi-Naini and J. Pieprzyk, RSA-based fail-stop signature schemes, *Proc. of International Conference on Parallel Processing Workshops*, Japan, pp.161-166, 1999.
[26] W. Susilo, Short fail-stop signature scheme based on factorization and discrete logarithm assumptions, *Theoretical Computer Science*, vol.410, no.8-10, pp.736-744, 2009.
[27] E. van Heyst and T. P. Pedersen, How to make efficient fail-stop signatures, *Lecture Notes in Computer Science*, vol.658, pp.366-377, 1993.
[28] E. van Heijst, T. P. Pedersen and B. Pfitzmann, New constructions of fail-stop signatures and lower bounds, *Lecture Notes in Computer Science*, vol.740, pp.15-30, 1993.