SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2012; **5**:725–737
Published online 24 August 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.367

RESEARCH ARTICLE

A private online system for executing wills based on a secret sharing mechanism

Chin-Ling Chen[1]*, Cheng-Chi Lee[2], Yuh-Min Tseng[3] and Teng-Tai Chou[1]

[1] Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168, Jifeng E. Rd., Wufeng District, Taichung, 41349 Taiwan, China
[2] Department of Library and Information Science, Fu Jen Catholic University, No. 510 Chung Cheng Rd, Hsin Chuang District, New Taipei City, 24205 Taiwan, China
[3] Department of Mathematics, National Changhua University of Education, Chang-Hua, 500 Taiwan, China

SUMMARY

Family quarrels over inheritance, although not new, have been featured prominently in the news in recent years. Thus, the issue of executing wills for the purpose of dividing inheritance is worth investigating. Acrimony caused by family disputes or distribution of inheritance has a negative impact on society. Thus, we seek to construct a method of constructing a secure and private escrow will.

The concept of secret sharing was proposed by Shamir and Blakley in 1979. Our method seeks to alleviate problems associated with secret sharing. We divide secret S into n pieces and distribute those pieces to n specific objects. We call the n owners the shadows. We then reconstruct these shadows and retrieve the original main key. Our method functions on the basis of the secret sharing mechanism. The proposed scheme combines the convenience of the Internet with cryptology technologies to solve the security problems of the online wills. It not only reduces cost and improves performance, but also prevents family infighting. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS

certificate; digital signature; RSA; secret share; will

***Correspondence**

Chin-Ling Chen, Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168, Jifeng E. Rd, Wufeng District, Taichung, 41349 Taiwan, China.

E-mail: clc@mail.cyut.edu.tw

1. INTRODUCTION

A common feature of many adventure stories is that of a map that leads the protagonist to immense treasure. Frequently, this map has been divided into many sections so as to further protect the treasure from undesirable hands. Only he who collects the entire map is able to retrieve the treasure. Although treasure seekers live mostly in the world of fantasy, the concept of dividing maps has real-world applications. In 1979, Shamir [1] and Blakley [2] first proposed the secret sharing mechanism to solve similar problems.

A will is a legal declaration of the intention of a testator with respect to his property, policy, or position, which he desires to be carried into effect after his death. Generally speaking, it is divided into two forms. One is in writing and the other through dictation. From antiquity onward, before Chinese emperors died, they always established testaments that stated which crown prince was to become emperor, and then the testaments were given to trusted ministers or eunuchs. However, it was inevitable that the testaments were changed by these people, creating problems that resulted in severe infighting in the court. There still are many hurdles that make it impossible to ensure that a testament is private and unforgeable. News stories of relatives fighting over estates are common. Private and secure wills are thus extremely important to our societies. Therefore, the use of paper trust wills has been proposed. Naturally, a legal basis should be used to resolve the problems aforementioned, but extant paper wills retain the following problems [3]:

Without intervention of impartial institutions, it is difficult to maintain the security of a will.

(1) Because a will only becomes valid in the presence of lawyers and witnesses, there exists high cost and low efficiency.
(2) Content cannot be easily changed, and execution is inconvenient.

(3) Disclosure of a will's plain text creates privacy problems.
(4) Planning for the deceased is complex.

Using computers and Internet has been proposed to solve these issues. However, the issue of information that is only to be accessed by authorized individuals must be addressed. So the digital signatures [4–6], encryption and decryption algorithms [7–10], and such are used to solve these kinds of problems. For example, Chien and Lin [11] recently proposed a protocol using bilinear pairing to solve some problems of paper wills. However, some questions still exist at the same time; for instance, the key is deposited, the private key produces the center burden overweight, and so on. On the basis of the RSA (Rivest R. L., Shamir A., and Adleman L.) [5] technology, we use a secret sharing mechanism [11–15] to propose another private online will system, and we used the court to fulfill the functions of transmission and escrow. The proposed scheme improves upon paper wills. We use a certification mechanism [16] and allow the court to escrow living wills to design a private online system for executing wills. Even though some applications involve a trusted third party [17,18] to achieve the reliability, the advantages of internal hacking remain. In our scheme, even attacks against the information held by the judiciary are thwarted. Thus, we use the court as an impartial entity and prevent such problem as difficulty in establishing secure management, lack of privacy, and high cost. We think that a good online escrow will system should fulfill the following requirements:

- Security: guarantee reliable security of the will throughout the entire process:
- Completeness [19]: guarantees that during the process of transmission, the no part of the will can be maliciously changed.
- Verifiability [20]: accuracy of the data must be verifiable throughout the process of transmission.
- Unforgeability [21,22]: the will cannot be forged.
- Non-repudiation [23]: evidence that has been signed cannot be repudiated.
- Privacy [22]: uses cryptography; even the third party transferring unit cannot know the will's contents.
- Prevents typical classes of attacks: (i) loss of certificate of death attacks; (ii) inside attacks; (iii) impersonation attacks; (iv) man-in-the-middle attacks.
- Practicality: easy to modify and save, it not only reduces cost but also improves performance.

In the basic form, a secret sharing scheme is a protocol divided into two phases [24]: Share and Reconstruct. During Share, a dealer distributes a secret among a set of participants by sending in a secure way a piece of information to each of them, called a share. Then, during Reconstruct, some subsets of participants called qualified subsets can reconstruct the secret either by pooling together their shares or by sending their shares in a secure way to a trusted party called a combiner who collects the shares, reconstructs the secret, and sends it back to these participants. Other subsets called forbidden subsets, even by pooling together and processing their shares, do not learn any information about the secret. In such a model, the dealer and participants are assumed to be honest.

However, many applications have to deal with the case of dishonest participants and possibly a dishonest dealer. In 1998, Tompa and Woll [14] showed that Shamir's threshold scheme can be subjected to the following attack, which can be applied to all linear secret sharing schemes. A dishonest participant, during Reconstruct, can submit to the combiner an opportunely constructed fake share. Hence, the reconstructed secret is different from the original one. But, from this secret, the dishonest participant (and only he) can recover the original secret.

Other sections of this paper are organized as follows: in the second section, we introduce our online will system; in the third section, we analyze and discuss the security of the online escrow will system; finally, in the fourth section, we present our conclusion.

2. THE PROPOSED PROTOCOL FOR INTERNET WILL

2.1. Our ideas

In our scheme, the applicant can use the online escrow will protocol to inform relatives of what he wishes to be done upon his death with respect to the distribution of his estate. First, the applicant must register with the court. Then, he must transmit the pre-selected parameters to family members. Next, the applicant uses a security mechanism based on secret share to encrypt the will, which is then given to a trusted authority (such as the court). This authority would then save and transmit the contents of the will so that each family member could use the secret shadow pre-designed by the applicant to decipher the ciphered text of the will. Each phase involves non-repudiation, which enhances the practicality of this protocol.

Figure 1 is a diagram of the escrow will we have designed. The designated family members can see their designated portion of the will in plain text, but they are unable to view the unauthorized portions of the will; not even the court can view the entire will in plain text. The entire will is notarized by the court, and is thus legally binding.

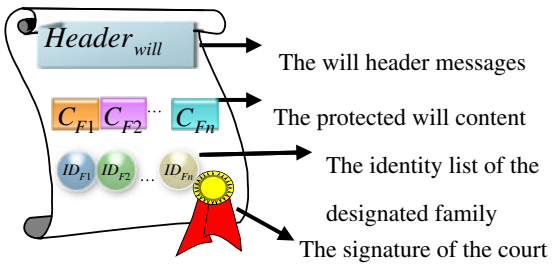

Figure 1. The conceptual diagram of the designed escrow will.

The will header $Header_{will}$ can be seen by every family member.

2.2. System framework

The structure of our scheme is illustrated in Figure 2. In our scheme, there are four parties described as follows:

- Applicant (A): A person who makes a living will.
- Court (C): The court that accepts the applicant's escrow for a living will.
- Folks (F): Those who have the right to know the contents of the will, such as family members of the applicant.
- Hospital (H): The hospital that issues the certificate of death.

(1) Applicant→Court: The applicant transmits his relevant information to the court for registration.
(2) Court→Applicant: The court checks the application and returns the certificate.
(3) Applicant→Court: Applicant determines will cipher text, decrypts the relevant portion of will and sends the will cipher to the court for escrow.
(4) Applicant→Folks: The applicant sends the relevant decryption information to each family member. Thus, each person receives relevant information that is not identical to that received by the others.
(5) Folks→Hospital: After the applicant dies, family members present the applicant's in formation (such as the identification number and birth date of the deceased) to apply for a certificate of death from the hospital.
(6) Hospital→Folks: After verification, the hospital issues the certificate of death to family members.
(7) Folks→Court: Family members submit the certificate of death along with one's own identification number to apply for an encrypted will with the court.
(8) Court→Folks: The court sends a secret shadow to the family members to decrypt the will cipher text.

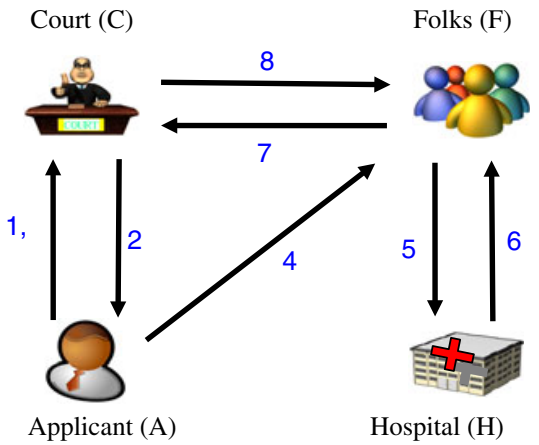

Figure 2. The structure of our scheme.

2.3. Notation

The following notations are used to explain how our scheme is constructed.

$E_x()$:	use X's public key to encrypt the message.
$D_x()$:	use X's secret key to decrypt message.
$S_x(\ )$:	use X's secret key to sign message.
$V_x(\ )$:	use X's public key to verify message.
(p_x, q_x):	a pair of prime numbers.
N_x:	a large number, where $N_x = p_x \cdot q_x$
$\phi(N_x)$:	the Euler totient function, where $\phi(N_x) = (p_x - 1) \cdot (q_x - 1)$.
PK_x:	the X's public key.
SK_x:	the X's secret key, where $PK_x \cdot SK_x = 1 \pmod{\phi(N_x)}$.
key_i:	the applicant made plain text key for the ith family member.
$Esym_{key_i}[m]$:	symmetry key key_i to encrypt the message m.
$Dsym_{key_i}[c]$:	symmetry key key_i to decrypt the message c.
$\|$:	the concatenate operation.
$h(\ )$:	one-way hash function.
$\oplus$:	exclusive-or operation, the length of all operational elements should be the same.
r:	the relevant parameters of the modified will.
m_{Fi}:	the applicant makes the will plain text for the ith family member.
C_{Fi}:	the applicant makes the will cipher text for the ith family member, where $C_{Fi} = Dsym_{key_i}[m_{Fi}]$ for $i = 1,2,\ldots, n$.
$Header_{will}$:	the header information of the will.
ID_{Fi}:	the identity code of the ith family member.
ID_A:	the identity code of the applicant.
Bir_A:	the applicant's birthday.
Sex_A:	the applicant's sex.
ID_{list}:	the family member's identity code list, where $ID_{list} = (ID_{F1} \| ID_{F2} \|, \ldots, \| ID_{Fn})$.
M_{will}:	the encrypted will.
m_{req}:	the registration request, where $m_{req} = (ID_A \| Bir_A \| Sex_A \| ID_{F1} \| ID_{F2} \|, \ldots, \| ID_{Fn})$.
$Cert_{death}$:	the death certificate that conforms to an X.509 format.
C_{req}:	the cipher text of the request application, which is generated by using the court's public key to encrypt theID_{Fi}, ID_A, and $Cert_{death}$, where $C_{req} = E_C(ID_A, ID_{Fi}, Cert_{death})$.
SG_{cert}:	the signature of the $h(C_{req})$, which is generated by using the ith family member's secret key to sign the message $h(C_{req})$, where $SG_{cert} = S_{Fi}(h(C_{req}))$.
α:	the relevant parameters generated by the applicant and sent to family member to decrypt the will cipher text.
k_i:	the factor for decrypting the secret shadow, where $k_i = (\alpha \oplus key_i)^{PK_\pi} \bmod N_\pi$.
m_{info}:	the personal information of the applicant.

2.4. The detail protocol

2.4.1. Initialization phase

First, the certificate authority (CA) chooses a pair of prime numbers (p_x,q_x) to compute the product $N_x=p_xq_x$ for the related role X. Then, the CA determines the public key and secret key pair $(PK_A,SK_A),(PK_{Fi},SK_{Fi}),(PK_C, SK_C)$ and (PK_π,SK_π) for applicant(A), folks(F_i), court (C), and Applicant and Folks, respectively [25], such that $N_A< N_{Fi}<N_C$, for $i=1$ to n. Where

$$PK_CSK_C = 1(\bmod\phi(N_C)) \tag{1}$$

$$PK_{Fi}SK_{Fi} = 1(\bmod\phi(N_{Fi})) \tag{2}$$

$$PK_\pi SK_\pi = 1(\bmod\phi(N_\pi)) \tag{3}$$

The following polynomial function is generated for embedding the common secret key SK_π of the applicant and folks:

$$f(x) = ax + SK_\pi(\bmod\phi(N_\pi)), \text{where } a \in [1,\phi(N_\pi)] \tag{4}$$

2.4.2. Registration phase

The applicant makes a request that the living will be registered by the court. The court authenticates the applicant's request and issues the certificate. The court also stores the related information in a database. The scenarios of the registration phase are shown in Figure 3.

Step 1: The applicant uses an identification number ID_A, birth date Bir_A, sex Sex_A,and the family member identifications $ID_{F1},ID_{F2},\ldots,ID_{Fn}$ to make the application request m_{req}, where $m_{req}=(ID_A\|\ Bir_A\|\ Sex_A\|\ ID_{F1}\|\ ID_{F2}\|,\ldots,\|ID_{Fn})$.

Then the applicant uses his own private key to sign the request.

$$SG_{req} = S_A(m_{req}) \tag{5}$$

The applicant sends (m_{req},SG_{req}) to the court and registers with the court.

Step 2: Once the applicant's request and signature have been received, the court uses the applicant's public key to verify the accuracy of the request and signature as follows:

$$V_A(SG_{req}) \quad m_{req} \tag{6}$$

If the court agrees with the applicant's request, it uses its own private key to make a signature for the request and choose a random number r. The court also issues the agreed certificate $Cert_C$ to the applicant.

$$Cert_C = S_C(m_{req},r) \tag{7}$$

Step 3: The applicant uses the court's public key to verify the certificate as follows:

$$V_C(Cert_C) \stackrel{?}{=} (m_{req},r) \tag{8}$$

2.4.3. The encrypted will delivery and storage phase

The applicant makes the plain text of the will. He also calculates and sends the relevant secret parameters to the court and all applicable family members. The scenarios of the encrypted will delivery and storage phase are shown in Figure 4.

Step 1: The applicant computes the parameters S_{Fi} and S_{CFi} as follows:

$$S_{Fi} = f(ID_{Fi})\frac{-ID_C}{ID_{Fi}-ID_C}\bmod\phi(N_\pi) \quad \text{for } i=1 \text{ to } n \tag{9}$$

$$S_{CFi} = f(ID_C)\frac{-ID_{Fi}}{ID_C-ID_{Fi}}\bmod\phi(N_\pi) \quad \text{for } i=1 \text{ to } n \tag{10}$$

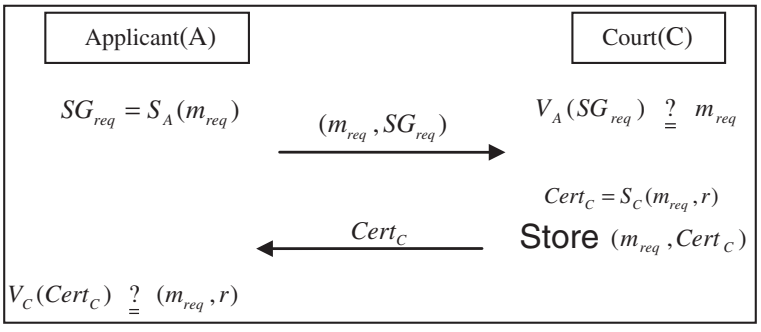

Figure 3. The scenarios of the registration phase.

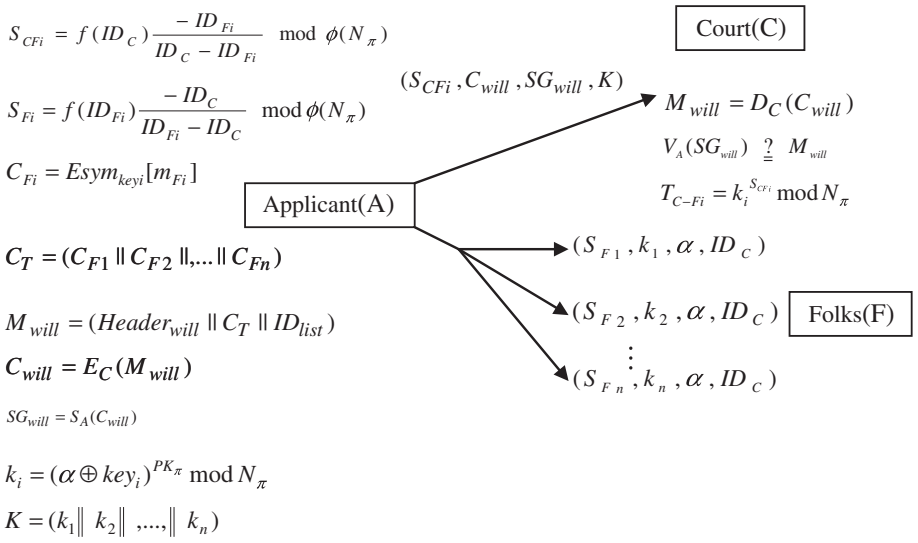

Figure 4. The scenarios of the encrypted will delivery and storage phase.

The applicant uses the symmetrical key key_i to encrypt the will in plain text m_{Fi} as follows:

$$C_{Fi} = Esym_{keyi}[m_{Fi}] \quad \text{for } i = 1 \text{ to } n \tag{11}$$

The applicant writes the full text of the will M_{will}. The full text of the will includes the header information $Header_{will}$, the cipher text C, and the identity list of the family members ID_{list} such that $M_{will} = (Header_{will}||C_T||ID_{list})$,where C_T=($C_{F1}\| \ C_{F2}\|,\ldots,\|C_{Fn}$) and ID_{list}=($ID_{F1}\| \ ID_{F2}\|,\ldots,\| \ 1\| \ ID_{F2}\|,\ldots,\|ID_{Fn}$). In order to prevent the C_{Fi} is intercepted by the attacker, the applicant uses the court's public key to encrypt the M_{will} into C_{will} as follows:

$$C_{will} = E_C(M_{will}) \tag{12}$$

Afterward, the applicant uses his/her own private key to sign the C_{will} as follows:

$$SG_{will} = S_A(C_{will}) \tag{13}$$

The applicant uses the common public key PK_π to encrypt the relevant parameters ($\alpha \oplus key_i$) and develops factor k_i to untie the secret key as follows: $k_i = (\alpha \oplus key_i)^{PK_\pi} \mod N_\pi$, for i=1 to n. The applicant sends (S_{CFi},C_{will},SG_{will},K) to the court as the application request of the escrow will, where K=($k_1\| \ k_2\|,\ldots,\| \ k_n$). Finally, the applicant sends (S_{Fi},$k_i a$,ID_C) to each relevant family member.

Step 2: The court uses its own private key to decrypt C_{will}

$$M_{will} = D_C(C_{will}) \tag{14}$$

And then the court also use the applicant's public key to verify accuracy of the will.

$$V_A(SG_{will}) \quad M_{will} \tag{15}$$

and then the court computes the secret shadow T_{C-Fi} as follows:

$$\begin{aligned} T_{C-Fi} &= k_i^{S_{CFi}} \mod N_\pi \\ &= k_i^{f(ID_C)[-ID_{Fi}/ID_C - ID_{Fi}]} \mod N_\pi \end{aligned} \tag{16}$$

Afterward, the court stores

$$(M_{will}, SG_{will}, ID_A, ID_{Fi}, T_{C-Fi}), \text{ for } i = 1 \text{ to } n$$

2.4.4. Death certificate application phase

After the applicant has died, family members submit the applicant's relevant information to the hospital to apply for the certificate of death. After the hospital has verified this information, the death certificate is given to the family members. The procedures of applying for a death certificate are shown in Figure 5.

Step 1: Upon the applicant's death, the family members submit relevant information regarding the applicant (for example identification numberID_A, birth dateBir_A) as well as their own identification code ID_{Fi}, resulting in m_{info}, as follows:

$$m_{\text{info}}(ID_A \| \ Bir_A \| \ ID_{Fi}) \tag{17}$$

The relative uses his/her private key to sign m_{info}

$$SG_{\text{info}} = S_{Fi}(m_{\text{info}}) \tag{18}$$

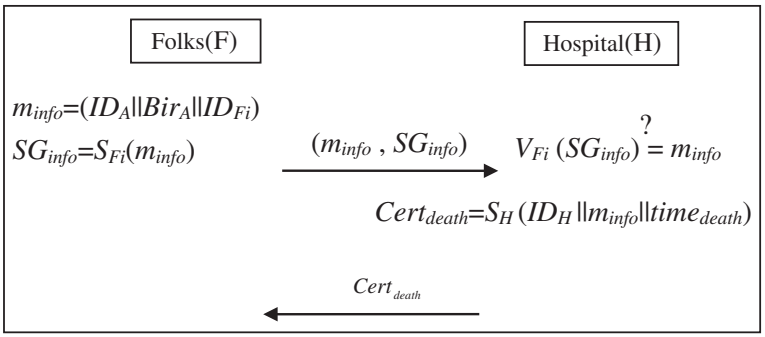

Figure 5. The scenarios of applying for death certificate phase.

and then sends $(m_{\inf o}, SG_{\inf o})$ to the hospital to ask for a death certificate.

Step 2: After receiving signatures from family members, the hospital uses the public key of the relevant family member to verify the correctness of the signature as follows:

$$V_{Fi}(SG_{\inf o}) \quad m_{\inf o} \tag{19}$$

Step 3: Once the given verification is completed, the hospital uses its private key to issue the death certificate, which includes the hospital code ID_H, m_{info}, and death time $time_{death}$:

$$Cert_{death} = S_H(ID_H \| m_{\inf o} \| time_{death}) \tag{20}$$

and then sends $Cert_{death}$ to the ith family members.

2.4.5. Decrypt the will cipher text phase

Family members submit the death certificate to the court to request the will. The court verifies the identities of family members then returns the contents of the will with the approval signature to the ith family members; family members use the parameter previously received from the applicant to decrypt the will into plain text. The procedures for decrypting the will cipher text are shown in Figure 6.

Step 1: The relative folks use the court's public key to encrypt the death certification $Cert_{death}$, the applicant's identification code ID_A, and his/her own identification code ID_{Fi}.

$$C_{req} = E_C(ID_A, ID_{Fi}, Cert_{death}) \tag{21}$$

Afterward, each relative uses his/her private key to make a signature:

$$SG_{cert} = S_{Fi}\big(h\big(C_{req}\big)\big) \tag{22}$$

Moreover, they send $(ID_{Fi}, C_{req}, SG_{cert})$ to the court to request the execution of the stored escrow will.

Step 2: Having received the folk's application, the court uses its private key to decrypt the relevant messages:

$$(ID_A, ID_{Fi}, Cert_{death}) = D_C\big(C_{req}\big) \tag{23}$$

The court also uses the public keys of family members to verify the correctness of the signature.

$$V_{Fi}(SG_{cert}) \stackrel{?}{=} h\big(C_{req}\big) \tag{24}$$

Moreover, the court checks the family member's identityID_{Fi}, verifies whetherID_A is in the court's database, and then uses the hospital's public key to verify the death certificate. The court then computes the secret shadow T_{C-Fi} and uses its private key to make a signature as follows:

$$SG_{shadow} = S_C(T_{C-Fi}) \tag{25}$$

The court uses the public key of the ith family member to encrypt the corresponding C_{Fi} and sign it as follows:

$$C_i = E_{Fi}(C_{Fi}) \tag{26}$$

$$SG_{Fi} = S_C(C_i) \tag{27}$$

Finally, the court sends $(SG_{shadow}, SG_{Fi}, C_i, T_{C-Fi})$ to the ith family member.

Step 3: The relatives use the court's public key to verify the received signature.

$$V_C(SG_{Fi}) \stackrel{?}{=} C_i \tag{28}$$

and uses his/her own private key to decrypt the C_i

$$C_{Fi} = D_{Fi}(C_i) \tag{29}$$

Next, the court's public key is used to verify the correctness of the secret shadow T_{C-Fi} as follows:

$$V_C(SG_{shadow}) \quad T_{C-Fi} \tag{30}$$

If the mentioned equality holds, it means that the escro will have not been altered. Moreover, the relatives calculate yet another secret shadow T_{Fi} as follows:

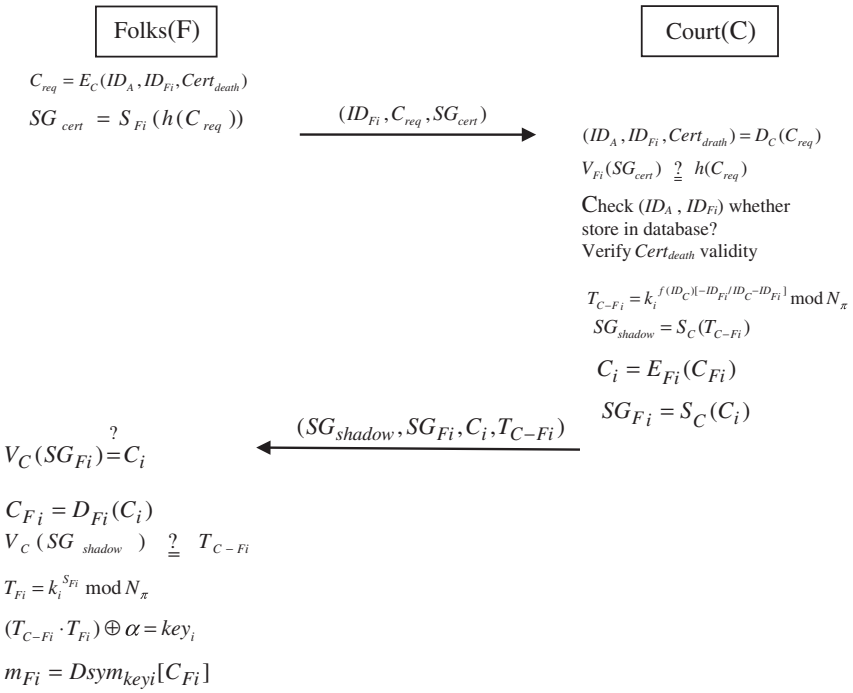

Figure 6. The scenarios of the will decrypting phase.

$$\begin{aligned} T_{Fi} &= k_i^{S_{Fi}} \bmod N_\pi \\ &= k_i^{f(ID_{Fi})[-ID_C/ID_{Fi}-ID_C]} \bmod N_\pi \end{aligned} \tag{31}$$

The relatives use the received secret shadow T_{C-Fi}, the calculated secret shadow T_{Fi}, and parameter a to make an exclusive-or operation to untie the symmetrical key key_i of the will cipher text. The derivations are described as follows:

$$\begin{aligned} &(T_{C-Fi} \cdot T_{Fi}) \oplus \alpha \\ &= (k_i^{S_{CFi}} \cdot k_i^{S_{Fi}}) \oplus \alpha \\ &= \left((k_i)^{f(ID_C)[-ID_{Fi}/ID_C-ID_{Fi}]}\right)\left((k_i)^{f(ID_{Fi})[-ID_C/ID_{Fi}-ID_C]}\right) \oplus \alpha \\ &= (k_i)^{(aID_{CFi}+SK_\pi)[-ID_{Fi}/(ID_C-ID_{Fi})]+(aID_{Fi}+SK_\pi)[-ID_C/(ID_{Fi}-ID_C)]} \oplus \alpha \\ &= (k_i)^{SK_\pi} \oplus \alpha \\ &= \left((\alpha \oplus key_i)^{PK_\pi}\right)^{SK_\pi} \oplus \alpha \\ &= key_i \end{aligned} \tag{32}$$

The family uses the symmetrical key key_i to decrypt the will plain text m_{Fi}.

$$m_{Fi} = Dsym_{keyi}[C_{Fi}] \tag{33}$$

3. ANALYSIS

In this section, we discuss the system requirements proposed in section one. We discuss the requisites to complete the following five security requirements: (i) completeness; (ii) verifiability; (iii) unforgeability; (iv) non-repudiation; and (v) privacy.

Thereafter, we analyze the following attack models:

(1) Attack by loss of death certificate.
(2) Internal attack.
(3) Impersonation attack.
(4) Man-in-the-middle attack.

Finally, we analyze of our system's practicability.

3.1. Security analyses

In this section, we demonstrate that our Internet will system satisfies the system security requirements presented in Section 1. We first show the security preliminaries of the adopted cryptographic schemes in our Internet will system. Then we show that our Internet will system achieves each of the five security requirements listed as follows: (i) completeness; (ii) verifiability; (iii) unforgeability; (iv) non-repudiation; (v) privacy.

3.1.1. Preliminaries of security

In the proposed Internet will system, the security is based on the RSA assumption. As shown in Section 2.3, we use two functionalities that include the public key encryption $E_X(\ )$/decryption $D_X(\ )$ [26,27] and the public key signing $S_X(\ )$ /verification $V_X(\ )$ [28–30], respectively.

In 1994, Bellare and Rogaway [26] introduced an optimal asymmetric encryption (OAEP) scheme using RSA assumption [5]. The OAEP scheme was believed to achieve semantic security against adaptive chosen cipher attacks.

In 2001, Fujisaki *et al*. [27] proved that the OAEP scheme based on the RSA assumption is semantically secure against adaptive chosen cipher attacks in the random oracle model [31]. They mainly showed that the security of the OAEP scheme can actually be proven under the one-wayness of the RSA function. In our Internet will system, we use Fujisaki *et al*.'s asymmetric encryption scheme to provide the public key encryption/decryption function. The security theorem of Fujisaki *et al*.'s asymmetric encryption scheme is presented hereafter.

Theorem 1. In the random oracle model, if an adversary with adaptive chosen cipher capability and a non-negligible advantage can violate the semantic security of OAEP scheme, then there exists a challenger to solve the one-wayness of the RSA function.

Proof. For the details of the proof for this theorem, refer to Ref. [27].

For the adopted signature scheme in our Internet will system, we may use the well-known RSA probabilistic signature scheme (RSA-PSS scheme) [28] under the RSA assumption in the random oracle model or the Cramer–Shoup signature scheme [29,30] without relying on random oracles under the strong RSA assumption. In 2000, Cramer and Shoup [29] proposed a signature scheme and proved that the proposed scheme is existential unforgeability under adaptive chosen-message attacks. In 2003, Fischlin [30] improved the Cramer–Shoup signature scheme to allow faster signing and verification. Here, we use the well-known RSA-PSS scheme [28] to our Internet will system. The security theorem of the RSA-PSS signature scheme is presented here. The following theorem shows the security of the RSA-PSS based on the security of the RSA assumption [5].

Theorem 2. In the random oracle model and under the security of the RSA function, the RSA probabilistic signature scheme (RSA-PSS scheme) is existential unforgeability under adaptive chosen-message attacks.

Proof. For the details of the proof for this theorem, refer to Ref. [28].

3.1.2. Analysis of completeness

In the discussion of completeness, we can use the following verification formula to verify whether the whole will has been forged in the process of transmission:

$$V_A(SG_{will}) \quad M_{will} \tag{15}$$

Any modification of M_{will} can be detected, and signal transmissions between different roles in each phase relies on signature mechanisms to meet the security requirement of completeness.

3.1.3. Analysis of verifiability

We explore verifiability from the perspective of signatures. In every phase, when the applicant signs his own application and sends the request to someone else, the other side must be able to first verify the accuracy of the signature; this portion is divided into five portions for discussion:

- During the registration phase, the applicant creates an application request using his own information $SG_{req}=S_A(m_{req})$, the accuracy of which can be verified by the court as follows:

$$V_A\left(SG_{req}\right) \quad m_{req} \tag{6}$$

- During the encrypted will delivery and storage phase, the applicant calculates the secret text of the will and signs it $SG_{will}=S_A(M_{will})$, the accuracy of which can be verified by the court as follows:

$$V_A(SG_{will})^A \quad M_{will} \tag{15}$$

- During the application for death certificate phase, family members submit partial information concerning the applicant and combine it with their own ID_{Fi}, creating one piece of information $m_{info}=(ID_A \| Bir_A \| ID_{Fi})$. They then use their private key to make a signature of family members $SG_{\mathrm{inf}o}=S_{Fi}(m_{\mathrm{inf}o})$. Then the hospital can verify the accuracy of the signature as follows:

$$V_{Fi}(SG_{\mathrm{inf}o}) \quad m_{\mathrm{inf}o} \tag{19}$$

- During the decryption of the will's cipher text phase, the court uses its private key to make a signature $SG_{shadow}=S_C(T_{C-Fi})$. After obtaining the signature, a family member can verify the signature as follows:

$$V_C(SG_{shadow}) \quad T_{C-Fi} \tag{30}$$

3.1.4. Analysis of unforgeability

We discuss the unforgeability from both the court's perspective and that of the family members.

The court cannot forge the content of the escrow will because the will's contents have been secured by the applicant using the symmetrical encryption algorithm. Thus, the court cannot know the content of the will. The court only calculates the secret shadow T_{C-Fi}; it cannot obtain the decryption key key_i to alter the will. The applicant's signature prevents the court from adding an identity code ID to the identity list ID_{list}. If the identity list ID_{list} is included in M_{will} and the M_{will} is altered, the signature SG_{will} will also be altered.

$$SG_{will} = S_A(M_{will}) \tag{12}$$

Family members cannot change the content of the will after receiving it. This is because the family members must use both his/her secret shadow T_{Fi} and the court's secret shadow T_{C-Fi} to get the key key_i to obtain the will.

$$m_{Fi} = Dsym_{keyi}[C_{Fi}] \tag{28}$$

Because each family members' key key_i is encrypted $\left(k_i = (\alpha \oplus key_i)^{PK_\pi} \bmod N_\pi\right)$, other family members are unable to forge another will. We thus achieve the unforgeability.

3.1.5. Analysis of non-repudiation

Because there is no face-to-face contact in Internet transactions, senders and receivers request digital signatures as evidence of the transaction. Such a design cannot be maliciously thwarted by the other party. In current online service applications, digital signatures are used to resolve the non-repudiation issue. In this section, we will explain how our structure fulfills non-repudiation demands during every phase. The following will explain how the proposed scheme meets the non-repudiation requirement.

Non-repudiation during the registration phase:

In Table I, we illustrate the proofs of non-repudiation during registration phase.

(1) Because the court received and verified the applicant's signature SG_{req}, the applicant cannot deny the registration.
(2) Once the court sends back the application certificate $Cert_C$ to the applicant, the court cannot deny this transaction.

Non-repudiation during the encrypted will delivery and storage phase:

During this phase, non-repudiation is achieved. In Table II, we illustrate the proofs of non-repudiation during this phase.

(1) Once court receives the applicant's signature SG_{will}, the applicant cannot deny this transaction.

Non-repudiation in the death certificate application phase:

In this phase, non-repudiation is achieved. In Table III, we illustrate the non-repudiation proofs for this phase.

(1) Once a hospital receives the relatives' request signature $SG_{\inf o}$, the relatives cannot deny this transaction during the death certificate application phase.
(2) The hospital also cannot deny the issued death certificate $Cert_{death}$.

Non-repudiation in the decryption of will text phase:

In this phase, non-repudiation is achieved. In Table IV, we illustrate the non-repudiation proofs for this phase.

(1) Once the court has received and verified the relatives' will cipher text signature SG_{cret}, the relatives cannot deny this request during the decryption will text phase.
(2) Once every relative has received the court's signature SG_{shadow} on the secret shadow, the court cannot deny this transaction during the decryption will text phase.

3.1.6. Analysis of privacy

In this protocol, we have added a security mechanism based on secret sharing to encrypt the will plain text before transmitting it to the court. The court and the relatives each compute secret shadows T_{C-Fi} and T_{Fi}, respectively. In this paper, a third party (the court) is involved to store and forward the will to each relative; we need not worry that the contents of the will be disclosed. The will's plain text is secure and protected. In our protocol, only those who know α, secret shadows T_{C-Fi}, and T_{Fi} can obtain the key key_i to untie an escrow will, where $key_i=(T_{C-Fi} \cdot T_{Fi}) \oplus \alpha$ and α is the related parameter that the applicant gave to relatives to decrypt the will cipher text. The decryption key key_i is used to obtain the will plain text m_{Fi} as follows: $m_{Fi}=Dsym_{keyi}[C_{Fi}]$. The proposed scheme holds with respect to privacy issues.

Table I. Non-repudiation during the registration phase.

Non-repudiation evidence	Evidence issuer	Evidence holder	Verification equation
(m_{req}, SG_{req})	Applicant (A)	Court (C)	$V_A(SG_{req})\ \ m_{req}$
$((m_{req}, r), Cert_C)$	Court (C)	Applicant (A)	$V_C(Cert_C)(m_{req}, r)$

Table II. Non-repudiation during the encrypted will delivery and storage phase.

Non-repudiation evidence	Evidence issuer	Evidence holder	Verification equation
(M_{will}, SG_{will})	Applicant (A)	Court (C)	$V_A(SG_{will})^A \quad M_{will}$

Table III. Non-repudiation during the death certificate application phase.

Non-repudiation evidence	Evidence issuer	Evidence holder	Verification equation
(m_{info}, SG_{info})	Folks (F)	Hospital (H)	$V_{Fi}(SG_{info})\ \ m_{info}$
$Cert_{death}$	Hospital (H)	Court (C)/Folks (F)	Use court's public key to verify $Cert_{death}$

Table IV. Non-repudiation during the decryption of will cipher text phase.

Non-repudiation evidence	Evidence issuer	Evidence holder	Verification equation
$(h(C_{req}), SG_{cert})$	Folks (F)	Court (C)	$V_{Fi}(SG_{cert})\ \ h(C_{req})$
(T_{C-Fi}, SG_{shadow})	Court (C)	Folks (F)	$V_C(SG_{shadow})\ \ T_{C-Fi}$

3.2. Analysis of attack patterns

3.2.1. Attack by loss of death certificate

If the death certificate sought by the family has been lost or stolen, a thief could obtain the applicant's identity code ID_A and family member's identity code ID_{Fi} and send them to the court along with the certificate of death so as to impersonate a family member and apply for a cipher text of the will. This cannot happen in our protocol, because family members must use their own secret keys to sign the application request $C_{req}$and signature value SG_{cert}.

$$C_{req} = E_C(ID_A, ID_{Fi}, Cert_{death}) \tag{21}$$

$$SG_{cert} = S_{Fi}\big(h\big(C_{req}\big)\big) \tag{22}$$

After the court has received (C_{req},SG_{cert}), the family member's public key can be used to verify the accuracy of the signature.

$$V_{Fi}(SG_{cert})\ \ h\big(C_{req}\big) \tag{24}$$

The main purpose is to prevent lost or stolen certificates of death. In this way, even though one has all the relevant information and the certificate of death, without the private key of family members for signature purposes, the court can detect illegality. Thus, one cannot receive the corresponding cipher text of the will and signature.

3.2.2. Internal attack

We achieve will completeness and privacy as described previously, even though we save and transmit the contents of the will through a third authority for the applicant. However, concerns remain that there may be internal attacks from within the authority institution. Our design uses a security mechanism based on secret sharing. The applicant first uses a symmetric encryption algorithm to encrypt the will plain text.

$$C_{Fi} = Esym_{keyi}(m_{Fi}) \text{for } i = 1 \text{ to } n \tag{11}$$

To obtain the will plain text, one must obtain the secret shadow of a family member; no one can know the contents of the will (of course including the members of the court). This ensures security against internal attack.

3.2.3. Impersonation attack

In this protocol, a family member who knows that other family members may obtain more assets and wants to impersonate other family members would be prevented from doing so. Not only would the related parameter (secret shadow) S_{Fi} given to each family member by the applicant not be the same, but also the factor used to untie secret shadow $k_i$would not be the same. Each family member must receive the related S_{Fi} and k_i. First, the applicant uses the court's secret shadow T_{C-Fi} (where $T_{C-Fi} = k_i^{S_{CFi}} \bmod N_\pi$) and combines it with another secret shadow T_{Fi} ($T_{Fi} = k_i^{S_{Fi}} \bmod N_\pi$), and using other related parameters α can only get the key key_i.

$$(T_{C-Fi} \cdot T_{Fi}) \oplus \alpha = key_i \tag{32}$$

After obtaining the key key_i, the will cipher text C_{Fi} can be decrypted into the will plain text m_{Fi}.

$$m_{Fi} = Dsym_{keyi}[C_{Fi}] \tag{33}$$

Thus, attacks that involve pretending to be other family members would not work.

3.2.4. Man-in-the-middle attack

Our protocol prevents information theft or false transmission information and false signatures during the transmission processes. Because of the entire transmission processes maintaining verifiability and non-repudiation, man-in-the-middle attacks cannot succeed against our protocol.

3.3. Discussion and performance analysis

The proposed scheme can easily be applied to the Internet. Our method allows living wills to be applied to existing online service protocols with enhanced security, practicality, and efficiency. The testator can also go online at any time in any location to entrust the contents of his or her

will. Digital wills lower costs by eliminating lawyers and witnesses. Based on the premise of legal efficiency, the online will system we propose is fast, convenient, simple, and practical.

Functionality comparisons between our proposed system and the Chien–Lin system [9] are summarized in Table V. The difference is that in our system, only the designated beneficiary can open the designated will. Note that the Chien–Lin system used paring-based encryption and signature schemes into their system. On the contrary, our system adopts the RSA-based encryption and signature schemes. Recently, some implementations [32,33] of pairing-based operations have been proposed. Especially, these implementations focus on the related pairing-based operations for low-power computing devices (i.e., smartcards). According to the presented experimental data, pairing-based operations are time consuming than the RSA-based operations.

In the following, we analyze our proposed Internet will system in terms of communication round, communication message size, and computational cost. For convenience, the following notations are used to concern with them. T_{EXP}/T_M denote the RSA computational cost of exponential/multiplication operation. T_S/T_V denote the RSA computational cost of the adopted signature signing/verifying functions $S_X(\ )/V_X(\ )$. T_E/T_D denote the RSA computational cost of the adopted encryption/decryption functions $E_X(\ )/D_X(\ )$. $|M|$ denotes the bit length of communication message. Thus, we have that $|C|$ is the bit length of block cipher message and $|S|$ is the length of signature. Without loss of generality, we may suppose the length of $|M|$, $|C|$, or $|S|$ is 1024 bits. Here, we assume that the applicant sends the escrow will to n folks. For example, if n=6, the total communication cost of the will delivery and storage phase is (6*3+1)*1024+1024+(6+2)*1024=28672 bits. If the transmission rate is 1 Mbps, the longest possible communication time is 28·672 ms. Table VI summarizes the communication round, the communication message size, and the computational cost of our proposed Internet will system.

We also make a performance comparison between our system and Chien and Lin's system in Table VII. The

Table V. Functionality comparisons between our proposed system and the Chien–Lin system.

	Our proposed escrow Internet will system	The Chien–Lin holographic will system [10]	The Chien–Lin sealed will system [10]
Security properties	Authenticity, integrity, non-repudiation, punctual decryption	Authenticity, integrity, non-repudiation	Authenticity, integrity, non-repudiation, punctual decryption
Adopted cryptographic system	RSA	Bilinear pairing	Bilinear pairing
Only the designated beneficiaries can open the dedicated part of the will, respectively	Yes	No	No

Table VI. Performance of the proposed Internet will system.

Phase	Communication round	Communication message size	Computational cost
Registration phase	2	$2\,S + M$	$2T_S+2T_V$
The will delivery and storage phase	$n+1$	$(3n)\,M + S +(n+3)\,C$	$(n+1)T_E+1T_D+(n+1)T_{EXP}+2T_S+2nT_M$
Apply for death certificate phase	2	$M +2\,S$	$2T_S+T_V$
Decrypt the will ciphertext phase	2	$M +3\,S +3\,C$	$(n+1)T_E+(n+2)T_D+(n+1)T_{EXP}+(n+3)T_S+(2n+2)T_V$

Table VII. Performance comparisons between our system and Chien and Lin's system.

	Chien and Lin's holographic e-will system [10]	Chien and Lin's sealed e-will system [10]	Our system
Computational cost of testator in the e-will creation phase	$4T_S+T_V$	$3T_S+T_V+(L_{will}/L_{key})\ T_{ENC}+T_{IBE}$	$2nT_M+nT_{ENC}+nT_{EXP}+T_S$
Computational cost of trusted authority in the e-will creation phase	$3T_S+2T_V$	$4T_S+3T_V$	T_V+T_{EXP}
Computational cost of hospital in the e-will open phase	$5T_S$	$5T_S$	T_S+T_V
Computational cost of trusted authority in the e-will open phase	$2T_V$	$2T_V+T_{DEC}+T_{IBD}$	$T_D+2T_V+T_{EXP}+T_S$

following notations are adopted directly from Chien and Lin's scheme. T_S denotes the computational cost for one signature operation, T_V denotes that for one signature verification, T_{ENC}/T_{DEC} denotes that for one symmetric encryption/decryption, and T_{IBE}/T_{IBD} denotes that for one identity-based encryption/decryption, respectively. Likewise, L_{will} denotes that of one e-will and L_{sig} denotes that for one signature. If Chien and Lin's scheme adopts Hess's signature scheme [34], then T_S require one modular exponentiation and two scalar multiplications in the G_1 field, and T_V requires one modular exponentiation and one pairing operation. If Chien and Lin's scheme adopts Boneh–Franklin's [35] identity-based encryption scheme, then T_{IBE} requires one scalar multiplication in G_1, one symmetric encryption, and one pairing operation, and T_{IBD} requires one pairing and one hash operations.

In our system, only the designated beneficiaries can open the dedicated part of the will. In this case, assume that there are n beneficiaries that are dedicated to open their own part of the will. In the e-will creation phase, the testator in the e-will creation phase and the court (the trusted authority) of our e-will open phase system require more computational costs than Chien and Lin's system. The point is that our system provides characteristics as shown in Table V. On the other hand, in the e-will creation phase and the hospital in the e-will open phase of our system require less computational costs than Chien and Lin's system according to Refs [32,33,36].

4. CONCLUSIONS

In contemporary society, the Internet is widely used; many online service systems have been proposed. Furthermore, the disadvantages of paper services are apparent. This paper uses mechanisms such as secret sharing, digital signatures, the encryption and decryption of public keys, symmetric encryption/decryption, and certificates to achieve the security needs of completeness, verifiability, unforgeability, non-repudiation, and privacy, and these can prevent various types of attack patterns. The private online system for executing wills we propose is enhanced by its practicality and can lower costs and increase efficiency. In summary, our protocol makes the following contributions:

(1) It can prevent family disputes resulting from fighting over estates.
(2) As an online escrow will, it decreases costs and increases efficiency.
(3) Privacy considerations are protected by powerful security mechanisms, which can withstand various types of attack.
(4) It provides convenient preservation because an outside institution is responsible for saving and transmitting the will.
(5) It improves on the disadvantages of paper and meets various security requirements.

REFERENCES

1. Shamir A. How to share a secret. *Communications of the ACM* 1979; **22**(11): 612–613.
2. Blakley G. Safeguarding cryptographic keys. Proceeding of AFIPS 1979 National Computer Conference. AFIPS Press: New York, 1979; 313–317.
3. Undertake and the trust law office in plan of the property legacy. Hong Kong, 2001. http://www.asset.hk/
4. Asokan N, Shoup V, Waidner M. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications* 2000; **18**: 593–610.
5. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM* 1978; **21**(2): 120–126.
6. Tseng YM. Digital signature of type and application. *Communications of the CCISA* 2001; **7**(3): 59–67.
7. Tseng YM, Jan JK, Chien HY. On the security of generalization of threshold signature and authenticated encryption. *IEICE Transactions on Fundamentals* 2001; **84**(10): 2606–2609.
8. Wang CT, Chang CC, Lin CH. Generalization of threshold signature and authenticated encryption for group communications. *IEICE Transactions on Fundamentals* 2000; **83**(6): 1228–1237.
9. Toorani M, Falahati A. A secure cryptosystem based on affine transformation. *Security and Communication Networks* 2011; **4**(2): 207–215.
10. Itani W, Kayssi A, Chehab A. Smart encryption channels for securing virtual machine-based networked applications. *Security and Communication Networks* 2009; **2**(6): 507–518.
11. Chien HY, Lin RY. The study of secure E-will system on the Internet. *Journal of Information Science and Engineering* 2009; **25**(3): 877–893.
12. Chien HY, Jan JK, Tseng YM. A practical (t, n) multisecret sharing scheme. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer* 2000; **83**(12): 2762–2765.
13. Harn L. Efficient sharing (broadcasting) of multiple secret. *Computers and Digital Techniques* 1995; **142**(3): 237–240.
14. Tompa M, Woll H. How to share a secret with cheaters. *Journal of Cryptology* 1988; **1**(3): 133–138.
15. Yang CC, Chang TY, Hwang MS. A (t, n) multi-secret sharing scheme. *Applied Mathematics and Computation* 2004; **151**(2): 483–490.
16. Housley R, Polk W, Solo D. Internet X.509 public key infrastructure certificate and CRL profile. *PKIX Working Group Internet Draft* 2001, http://www.ietf.org/rfc/rfc2459.txt
17. Ajman S, Morris R, Liskov B. A trusted third-party computation service. In *MIT Laboratory for Computer Science 200 Technology Square*, Cambridge, 2001;

512–521, Technical Report MIT-LCS-TR-847, MIT, May 2001.
18. Franklin MK, Reiter MK. Fair-exchange with a semi-trusted third party. *Proceedings of the Fourth ACM Conference on Computer and Communications Security*, pp. 1–5, April 1–4, 1997, Zurich, Switzerland.
19. Chen YY, Jan JK, Chen CL. The design of a secure anonymous Internet voting system. *Computers and Security* 2004; **23**: 330–337.
20. Susilo W, Zhang F, Mu Y. Identity-based strong designated verifier signature schemes. Springer-Verlag, 2004; LNCS 3108: 313–324.
21. Chen YY, Chen CL, Jan JK. Design of a fair proxy raffle protocol on the Internet. *Computer Standards & Interfaces* 2005; **27**(4): 417–424.
22. Chen YY, Chen CL, Jan JK. A mobile ticket system based on personal trusted device. *Wireless Personal Communications* 2007; **40**(4): 569–578.
23. Zhou J, Gollmann D. A fair non-repudiation protocol. *Proceedings of the IEEE Symposium Research in Security and Privacy* 1996; 55–61.
24. Stinson DR. An explication of secret sharing schemes. *Designs, Codes Cryptography* 1992; **2**(4): 357–390.
25. Chen YY, Jan JK, Chen CL. The design of a secure anonymous Internet voting system. *Computers & Security* 2004; **23**(4): 330–337.
26. Bellare M, Rogaway P. Optimal asymmetric encryption—how to encrypt with RSA. *Eurocrypt '94*, 1995; LNCS 950: 92–111.
27. Fujisaki E, Okamoto T, Pointcheval D. Stern J. RSA-OAEP is secure under the RSA assumption. *Proceedings of 21st Annual International Cryptology Conference (Crypto'01)*, Santa Barbara, CA, USA, Aug 19–23, 2001; 260–274.
28. Bellare M, Rogaway P. The exact security of digital signatures—how to sign with RSA and Rabin. In *Eurocrypt'96* 1996; LNCS 1070: 399–416.
29. Cramer R, Shoup V. Signature schemes based on the strong RSA assumption, ACM Trans. *On Information and System Security* 2000; **3**(3): 161–185.
30. Fischlin M. The Cramer–Shoup strong-RSA signature scheme revisited. In *PKC 2003, LNCS* **1567**. 2003; 116–129.
31. Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security (ACM CCS '93)*. Fairfax, Virginia, USA, 1993; 62–73.
32. Scott, M, Costigan, N, Abdulwahab, W. Implementing cryptographic pairings on smartcards, Proceedings of CHES 06, Yokohama, Japan, 10–13 October, 2006; 134–147.
33. Bertoni, G, Breveglieri, L, Chen, L, Fragneto, P, Harrison, K, Pelosi, G. A pairing SW implementation for smart-cards. *Journal of Systems and Software* 2008; **81**(7): 1240–1247.
34. Hess F. Efficient identity based signature schemes based on pairings. In *Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography*, LNCS 2595, 2002; 310–324.
35. Boneh D, Franklin M. Identity based encryption from the Weil pairing. In *Proceedings of Advances in Cryptology CRYPTO*, LNCS 2139, 2001; 213–229.
36. Ghoreishi, SS, Pourmina, MA. High speed RSA implementation based on modified Booth's technique and Montgomery's multiplication for FPGA platform. 2009 Second International Conference on Advances in Circuits, Electronics and Micro-Electronics, 2009; 86–93.