A Private Will System Design on the Internet

Chin-Ling Chen
Department of Computer Science and Information Engineering, Chaoyang University of Technology
clc@mail.cyut.edu.tw

Teng-Tai Chou
Department of Computer Science and Information Engineering, Chaoyang University of Technology
s9527632@mail.cyut.edu.tw

Abstract

The design of an escrow will system means to pre-build a living will, and escrow the will to the court via Internet. Of course, a prefect escrow will system should consider security issues. The proposed Internet will system conforms to the law specification and with privacy. Besides, it achieves the security requirements of the escrow will. The proposed scheme combines the convenience of Internet and cryptology technologies to solve the security problems of the Internet will system. It not only reduces the cost and improves the performance, but also prevents the brothers fighting each other.

Keywords: Digital signature, certificate, Internet will

1. Introduction

An escrow will [18] refers to the method by which the applicant establishes a living will to list things to tell friends and relatives after one's death and to plan the assets. The details are listed in the will, which is given to an authority in trust. The purpose of the applicant in establishing the trust will is so that after death, estate can be managed by a just authority in trust, to ensure that the estate can be used according to plans by the applicant while in life, so that those authorized by the will can receive the entrusted assets.

Since ancient times, before Chinese emperors die, they always establish testaments [19] that state which crown prince is to become emperor, and then the testaments are given to trusted ministers or eunuchs who have served for long periods of time. However, it is inevitable that the testaments are changed by these people, creating problems that result in severe in-fighting in the court. There are many problems, such as the escrow problem of the testament, making it impossible to ensure that the testament is private and unforgeable. In recent years, there are often news stories about brothers fighting each other because they want the estate. All these issues show that the privacy and security of a will are extremely important. Thus, the issue of establishing paper trust wills has been proposed. Naturally, a legal basis should be used to resolve the problems mentioned, but existing paper wills still have the following problems:

- Without intervention by just institutions, it is difficult to maintain security of the will.
- Because the will becomes valid in the presence of lawyers and witnesses, there is high cost and low efficiency.
- Content cannot be readily changed, execution is inconvenient.
- Disclosure of the will plaintext creates privacy problems.
- Planning for the deceased is complex, and it is difficult to be fully executed.

Due to these disadvantages of paper wills, we use mechanisms in cryptography [2,3,10] to propose a private online will system to overcome the various problems of paper wills. In recent years, Chien et al. [6] proposed a protocol using bilinear pairing to solve some problems of paper wills. We propose yet another security mechanism based on discrete logarithm problem [4,5,7,12] ,which is a private online trust will system, and incorporates the court to fulfill the functions of transmission and escrow. Even though there are some applications such as Trusted Third Party (TTP) [1,11], it is hard for them to exist in the current society because there are still dangers of internal hacking. Even with participation of the court, it can still prevent the court's attack strategy. Thus we set the court as the department that holds the will to make it more secure and solve more of the disadvantages of paper wills, such as difficulty of secure management, lack of privacy, and cost problems. Thus we think that a good online trust will system should fulfill the following requirements:

■ Security: guarantee the reliability of security for the trust will in the entire process:

(1) Completeness [13]: guarantees that during the process of transmission, the whole will would not be maliciously changed.

(2) Verifiability [9]: accuracy of the data must be verifiable through the process of transmission.
(3) Unforgeability [14]: authenticity of the will cannot be forged.
(4) Non-repudiation [16]: evidence that has been signed cannot be repudiated.
(5) Privacy [17]: uses cryptography; even the third party transferring unit cannot know the will's contents.
(6) Prevents attack patterns: loss of certificate of death attacks, internal hacking, impersonate attacks, man-in-the-middle attacks.

- Practicality: easy to modify and save.
- Problems in costs and efficiency: fewer lawyers and witnesses, which lowers costs, complexity of time, and storage space.

Our protocol not only satisfies the above requirements, but also added some models that prevent some attacks to make our structure more comprehensive, secure, so the applicant can entrust the will without concern.

Other sections of this paper are described as follows: in the second section we will introduce the online will system we propose; in the third section we will analyze and discuss security of the online trust will system; finally, in the fourth section we make a conclusion.

2. The proposed protocol

2.1. Our ideas

In our structure, the applicant can use the online trust will protocol to inform relatives and friends of what he wishes to be done after his death, and reasonably distributes his estate. First, the applicant must register to the court, then transmit related parameter information to family members, then the applicant uses a security mechanism based on discrete logarithm problem to encrypt the will so it becomes private, then it would be given to a trusted authority (such as the court), which would then save and transmit contents of the will for the applicant, so that each family member can use the parameter set by the applicant to open the portion of the will plaintext that they have been designated to see. Each phase involves non-repudiation, which enhances the practicality of this protocol in real life.

Figure 1 is the designed escrow will conceptual diagram we have designed. The designated family members can see their correspondingly designated will plaintext m_{willi}, but they are unable to view the unauthorized portions of the will; not even the court can view will plaintext m_{willi}. The whole will has the court's signature, so that the will is legally binding, and the will header $Header_{will}$ can be seen by each family member.

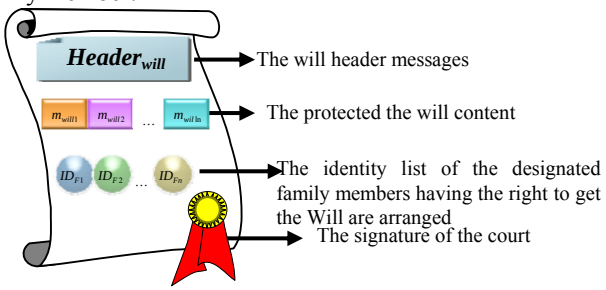

Figure 1. The conceptual diagram of the designed escrow will

2.2. System framework

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

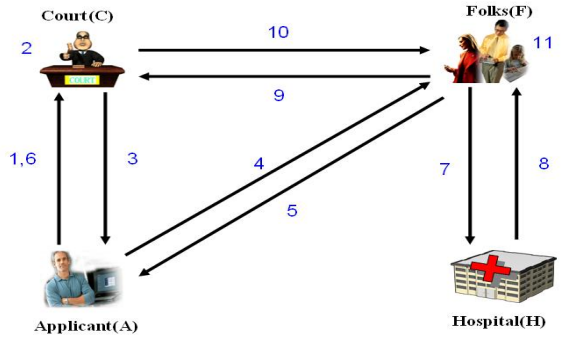

Figure 2. The structure of our scheme

- Applicant (A): A person who makes a living will.
- Court (C): The court which accepts the applicant's registrations for an escrow 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 which issues the certificate of death.

1. The applicant (A) registers with the court and submits the application.
2. The court (C) determines whether it agrees with the application.
3. The court returns the application agreement signature value.
4. The applicant separately gives related parameters to all family members (F) who have a right to get the will, providing information that decrypts the

encrypted will plaintext.

5. Family members return a self-selected parameter to the applicant.
6. The applicant uses the mechanism based on discrete logarithm problem to make a will, then encrypts the will plaintext, and gives it to the court for proof of signature, which completes the escrow will procedure.
7. After the applicant dies, family members use information relating to the applicant such as the identification number and birth date of the deceased to apply for a certificate of death with the hospital (H).
8. After verification by the hospital, the certificate of death is signed and given to family members.
9. Family members submit the certificate along with one's own identification number to apply for an encrypted will to which the court has applied its signature.
10. After the court has verified the certificate of death and family members' identification numbers, the encrypted will is returned to family members.
11. The family members can use the related parameters given when the applicant was still alive to decrypt the encrypted will.

2.3. Notations

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 number.

N_x: a large number, where $N_x = p_x \cdot q_x$.

$\phi(N_x)$: the Euler totient function, where $\phi(N_x) = (p_x - 1) \cdot (q_x - 1)$

e_x: the X's public key.

d_x: the X's secret key, where $e_x \cdot d_x = 1 \pmod{\phi(N_x)}$.

||: the concatenate operation.

m_{willi}: the applicant made the will plaintext for the i^{th} family member.

$Header_{will}$: the header information of the will.

ID_{Fi}: the identity code of the i^{th} family member.

ID_A: the identity code of the applicant.

ID_{list}: the family member code list, where $ID_{list} = (ID_{F1} \| ID_{F2}, \ldots, \| ID_{Fn})$.

Bir_A: the applicant's birthday.

m_{info}: the applicant and the i^{th} family member's personal information, where $m_{info} = (ID_A \| Bir_A \| ID_{Fi})$.

x_i, y_i: one pair parameter which is selected by the i^{th} family member and applicant respectively.

m_{Ai}, m_{Fi}: the i^{th} encrypted will plaintext which is made by the applicant.

M_{will}: the encrypted will, where $M_{will} = (Header_{will} \| (m_{A1}, m_{F1}) \| (m_{A2}, m_{F2}), \ldots, \| (m_{An}, m_{Fn}) \| ID_{list})$

$Cert_{death}$: the death certificate to conform X.509 format [20].

$h(\ )$: one way hash function.

2.4. Initialization phase

First, an applicant selects a set of random values $y_1, y_2, \ldots, y_n$ for *n* family members, and then selects $\alpha \in Z_{N_x}^*$ and $\beta \in Z_{N_x}^*$, where $N_x = p_x \cdot q_x$. Family member also chooses a secret parameter x_i such that $x_i \in Z_{N_x}^*$. Let $\gcd(\alpha, \beta) = 1$, $\alpha y_i + \beta x_i = 1$.

We involve the RSA cryptography in this protocol. Users choose two prime numbers p_x and q_x. And then compute $N_x = p_x \cdot q_x$, $\phi(N_x) = (p_x - 1) \cdot (q_x - 1)$. Find one set value (e_x, d_x) such that $e_x \cdot d_x \equiv 1 \bmod \phi(N_x)$, where e_x is public key; d_x is private key. The (e_x, N_x) is published. Afterward, the users encrypt the will plaintext *m* into *C*, $C = m^{e_x} \bmod N_x$. The receiver can use a secret key d_x to decrypt *C* and get *m*, $C^{d_x} \bmod N_x = m^{e_x d_x} \bmod N_x = m$. In our protocol, N_C is calculated by court; N_A is calculated by applicant and N_{Fi} is calculated by the i^{th} family members. Let $N_A < N_{Fi} < N_C$, for *i=1* to *n*.

2.5. Registration phase

The applicant makes a living will to the court for registration, and the court verifies the information according to needs of the applicant, and signs the signature value of approval, and then saves information relating to the applicant in the court's database. After receiving the signature value of the court's approval, the applicant verifies accuracy of the court's signature value. The primary procedures at the registration phase are shown in Figure 3 below.

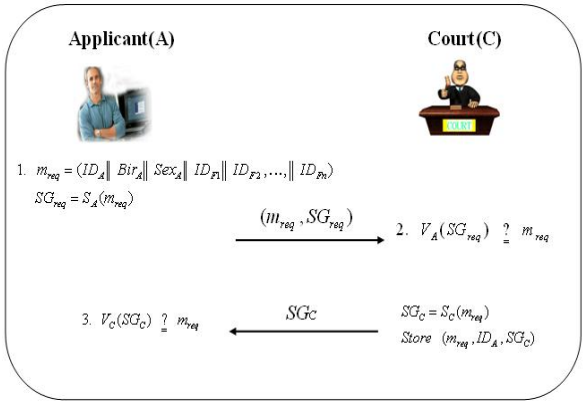

Figure 3. The scenarios of registration phase

Step1: The applicant creates the application request information m_{req} using his own identification number ID_A, birth date Bir_A, sex Sex_A, and the family members $ID_{F1}, ID_{F2}, ..., ID_{Fn}$ to whom the applicant desires to give the will.

$$m_{req} = (ID_A \| Bir_A \| Sex_A \| ID_{F1} \| ID_{F2}, \ldots, \| ID_{Fn})$$

Then he uses his own private key to sign the application materials, which is sent along with the application request m_{req} to the court, and registers with the court.

$$SG_{req} = S_A(m_{req})$$

Step2: Once the applicant's application request and signature value have been received, the court uses the applicant's public key to verify accuracy of the application and signature value.

$$V_A(SG_{req}) \stackrel{?}{=} m_{req}$$

If the court agrees with the applicant's application, the court uses its own private key to make a signature to the application.

$$SG_C = S_C(m_{req})$$

And saves the applicant's information and the approval signature value (m_{req}, ID_A, SG_C) in the court's database and transmits SG_C to the applicant.

Step3: After the applicant receives the court's approval signature value, he uses the court's public key to verify the accuracy.

$$V_C(SG_C) \stackrel{?}{=} m_{req}$$

2.6. The will to deliver and store phase

The applicant uses the self-selected parameters and transmit them to all related family members who have a right to the will, and the family members return a self-selected parameter, then the applicant uses the self-selected parameter and parameters returned by family members to calculate the ciphertext of the will that can be received by each family members, which is then sent to the court for proof of signature. The procedures for the transmission and signature stage of the will are shown in Figure 4.

Step 1: The applicant uses the self-selected parameters α and β, and selected $y_1, y_2, ..., y_n$ for n family members, the designated code for the court ID_C and will header $Header_{will}$ are encrypted with the public key of family members, getting the following $C_{\sec i}$.

$$C_{\sec 1} = E_{F1}(Header_{will}, ID_C, ID_A, y_1, \alpha, \beta)$$
$$C_{\sec 2} = E_{F2}(Header_{will}, ID_C, ID_A, y_2, \alpha, \beta)$$
$$\vdots$$
$$C_{\sec n} = E_{Fn}(Header_{will}, ID_C, ID_A, y_n, \alpha, \beta)$$

And then sends $C_{\sec 1}, C_{\sec 2}, ..., C_{\sec n}$ to every related family member.

Step2: When each family member receives his own document, he can use his own private key for decryption, which would use related parameters y_i, α, β to decrypt the will ciphertext.

$$(Header_{will}, ID_C, ID_A, y_1, \alpha, \beta) = D_{F1}(C_{\sec 1})$$
$$(Header_{will}, ID_C, ID_A, y_2, \alpha, \beta) = D_{F2}(C_{\sec 2})$$
$$\vdots$$
$$(Header_{will}, ID_C, ID_A, y_n, \alpha, \beta) = D_{Fn}(C_{\sec n})$$

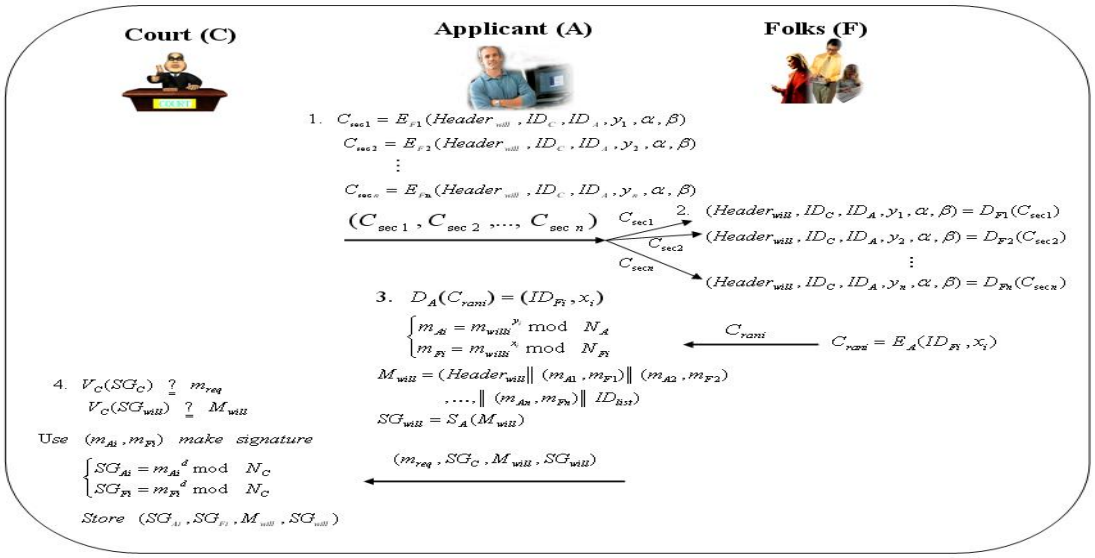

Figure 4. The scenarios of the will to deliver and store phase

Secondly, each family member would select a random number x_i , encrypted with the applicant's public key along with one's own identification code ID_{Fi}

$$C_{rani} = E_A(ID_{Fi}, x_i)$$

then C_{rani} is returned to the applicant.

Step3: Once the applicant has received the encrypted data, decryption with his/her own private key to get the results ID_{Fi} and x_i .

$$D_A(C_{rani}) = (ID_{Fi}, x_i)$$

Then the applicant uses the self-selected parameter y_i and family member parameter x_i to calculate:

$$\left.\begin{aligned} m_{Ai} &= m_{willi}{}^{y_i} \bmod N_A \\ m_{Fi} &= m_{willi}{}^{x_i} \bmod N_{Fi} \end{aligned}\right\} \text{ for } i=1 \text{ to } n$$

The applicant makes full text of will.

$M_{will} = (Header_{will} \| (m_{A1}, m_{F1}) \| (m_{A2}, m_{F2})$
$,\ldots, \| (m_{An}, m_{Fn}) \| ID_{list})$

where $ID_{list} = (ID_{F1} \| ID_{F2} \| \ldots \| ID_{Fn})$,

for *i=1* to *n*.

Then the applicant uses his own private key to sign the full will text M_{will}

$$SG_{will} = S_A(M_{will})$$

Then sends $(m_{req}, SG_C, M_{will}, SG_{will})$ to the court.

Step4: The court uses its own public key to verify whether the applicant has registered.

$$V_C(SG_C) \stackrel{?}{=} m_{req}$$

And uses the applicant public key to verify accuracy of the will.

$$V_A(SG_{will}) \stackrel{?}{=} M_{will}$$

Step5: Then the court uses its own private key *d* to sign m_{Ai} and m_{Fi} .

$$\left.\begin{aligned} SG_{Ai} &= m_{Ai}{}^d \bmod N_C \\ SG_{Fi} &= m_{Fi}{}^d \bmod N_C \end{aligned}\right\} \text{ for } i=1 \text{ to } n$$

According to the family member codes ID_{Fi} which gets from the ID_{list} , the court store the corresponding signatures and information $(m_{req}, SG_C, M_{will}, SG_{will})$

2.7. Apply for death certificate phase

After the applicant has died, family members submit the applicant's relevant information to the hospital in application for the certificate of death. After

the hospital has verified the information, the certificate of death would be given to the family members. The procedures of applying for a certificate of death are shown in Figure 5.

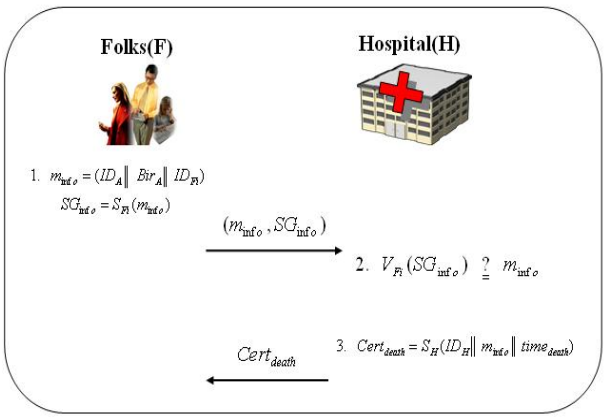

Figure 5. The scenarios of applying for death certificate phase.

Step1: When the applicant died, the family members propose related information of the applicant (for example identification number ID_A, birth date Bir_A) and family member's identification code ID_{Fi} to make a $m_{\inf o}$.

$$m_{\inf o} = (ID_A \| Bir_A \| ID_{Fi})$$

The folks use his/her private key to sign $m_{\inf o}$

$$SG_{\inf o} = S_{Fi}(m_{\inf o})$$

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

Step2: After receiving the signature from family members, the hospital uses the public key of family member to verify the correct of signature

$$V_{Fi}(SG_{\inf o}) \stackrel{?}{=} m_{\inf o}$$

Step3: Once the above verification is correct, the hospital use its private key to issue the death certificate which includes hospital code ID_H, applicant's personal information m_{info} and death time $time_{death}$

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

And then sends $Cert_{death}$ to the i^{th} family members.

2.8. Decrypt the will ciphertext phase

Family members submit the certificate of death to the court to request for the will of the applicant. The court verifies identities of family members then returns the contents of the will with the approval signature to the i^{th} family members, family members use the parameter previously received from the applicant to decrypt the will plaintext, receiving their own will plaintexts. The procedures for decrypting the will plaintext are shown in Figure 6.

Step1: The folks use the court's public key to encrypt the death certification $Cert_{death}$, the applicant's identification number ID_A and his/her own identification code ID_{Fi}.

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

Afterward, use the hash function to compute $h(C_{req}, ID_{Fi})$, and uses folk's private key to make signature.

$$SG_{cert} = S_{Fi}(h(C_{req}, ID_{Fi}))$$

Moreover, sends $(ID_{Fi}, C_{req}, SG_{cert})$ to the court to request for applying the stored escrow will.

Step2: Once receiving the folk's application, the court uses its private key to decrypt the related messages.

$$(ID_A, ID_{Fi}, Cert_{death}) = D_C(C_{req})$$

The court also uses the public key of family member to verify the correct of signature.

$$V_{Fi}(SG_{cert}) \stackrel{?}{=} h(C_{req}, ID_{Fi})$$

Moreover, the court checks family member's code ID_{Fi}, verifies whether ID_A is in court's database or not. If it is true, then uses the public key of the hospital to verify the death certificate. If the above verifications are all correct, then return the court's signature $(SG_{Ai}, SG_{Fi}, M_{will}, SG_{will})$ to the i^{th} family member.

Step3: The folks use the applicant's public key to verify the received signature.

$$V_A(SG_{will}) \stackrel{?}{=} M_{will}$$

Then use court's public key to verify sign's correct.

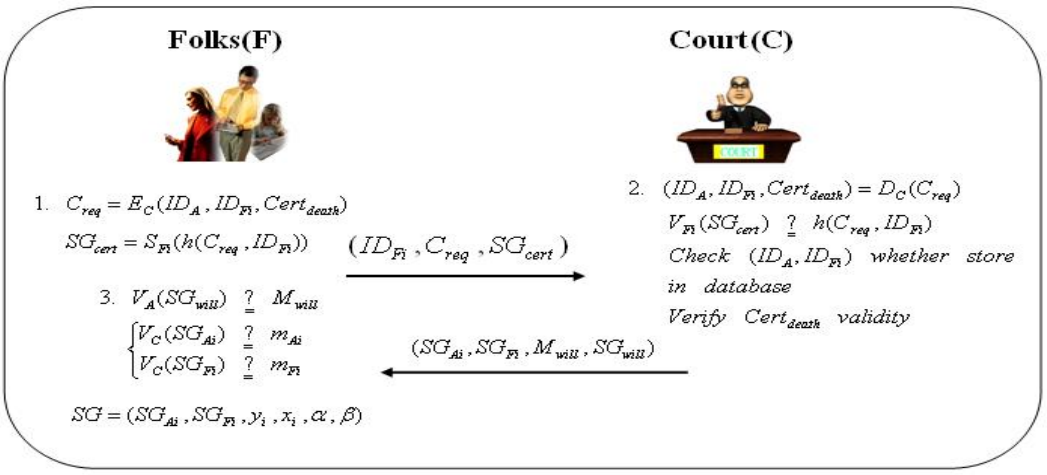

Figure 6. The scenarios of decrypting will text phase

$$\left.\begin{array}{l} V_C(SG_{Ai}) \stackrel{?}{=} m_{Ai} \\ V_C(SG_{Fi}) \stackrel{?}{=} m_{Fi} \end{array}\right\} \text{ for } i=1 \text{ to } n$$

Then uses (y_i, α, β) which is sent by applicant and self-selected x_i by initial phase to decrypt the designated will $(m_{willi})^d \bmod N_C$ .The derivations are described as follows:

$$\begin{aligned} SG &= (SG_{Ai}, SG_{Fi}, y_i, x_i, \alpha, \beta) \\ SG &\equiv SG_{Ai}^{\alpha} \; SG_{Fi}^{\beta} \bmod N_C \\ &\equiv (m_{Ai}^{d})^{\alpha} \; (m_{Fi}^{d})^{\beta} \bmod N_C \\ &\equiv (m_{willi}^{y_i d})^{\alpha} \; (m_{willi}^{x_i d})^{\beta} \bmod N_C \\ &\equiv m_{willi}^{(\alpha y_i + \beta x_i) d} \bmod N_C \\ &\equiv (m_{willi})^{d} \bmod N_C \end{aligned}$$

So, the i^{th} family member can use the court's public key *e* to get the escrow will which is stored in court database $(SG)^e \bmod N_C = m_{willi}$. Due to the m_{willi} with the court's signature $SG = (m_{will})^d \bmod N_C$, Thus the designated will is legal.

3. Analyses and discussions

In this section, we will discuss the system requirements proposed in section one separately. First of all, we discuss that needs to reach these five security requirements below with the cryptography view:

(1) Completeness.
(2) Verifiability.
(3) Unforgeability.
(4) Non-repudiation.
(5) Privacy.

Thereafter we analysis common attack models, the practicability of the analysis system and terms of the cost question again of ours finally.

3.1. Security analyses

3.1.1. Analysis of completeness

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

$$V_A(SG_{will}) \stackrel{?}{=} M_{will}$$

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

3.1.2. Analysis of verifiability

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

1. In the registration phase: the applicant creates an application request using one's own information $SG_{req} = S_A(m_{req})$, the accuracy of which can be verified by the court

$$V_A(SG_{req}) \stackrel{?}{=} m_{req}.$$

2. In the will to deliver and store 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 $V_A(SG_{will}) \stackrel{?}{=} M_{will}$.
3. Apply for death certificate phase: family members proposes partial information of the applicant and combines with one's own *ID*, creating one piece of information, and uses the private key signature of family members $SG_{\inf o} = S_{Fi}(m_{\inf o})$, then the hospital can verify accuracy of the signature value $V_{Fi}(SG_{\inf o}) \stackrel{?}{=} m_{\inf o}$.
4. Decrypt the will text phase: the applicant sends the encrypted will to the court, then the court uses its private key to sign.

$$\left.\begin{aligned} SG_{Ai} &= m_{Ai}{}^{d} \bmod N_C \\ SG_{Fi} &= m_{Fi}{}^{d} \bmod N_C \end{aligned}\right\} \text{ for i=1 to n}$$

After family members receive secret text of the will, they can verify accuracy of the court's signature.

$$\left.\begin{aligned} V_C(SG_{Ai}) &\stackrel{?}{=} m_{Ai} \\ V_C(SG_{Fi}) &\stackrel{?}{=} m_{Fi} \end{aligned}\right\} \text{ for i=1 to n}$$

3.1.3. Analysis of unforgeability

In the discussion on unforgeability, we divide the explanation into the court's perspective and the family members' perspective.

1. For the court, it cannot forge content of the trust will, because contents of the will have been encrypted by the applicant after calculation, so the court cannot see the content of the will text.

$$\left.\begin{aligned} m_{Ai} &= m_{willi}{}^{y_i} \bmod N_A \\ m_{Fi} &= m_{willi}{}^{x_i} \bmod N_{Fi} \end{aligned}\right\} \text{ for i=1 to n}$$

Also enclosing applicant's stamped signature of this escrow testament on the testament, the court can't join anyone's identity code *ID* to in ID_{list} at will either, because M_{will} includes ID_{list}, if whole testament M_{will} is altered, stamped signature value SG_{will} will be followed to change.

$$SG_{will} = S_A(M_{will})$$

2. For family members, they cannot change the content of the will text after receiving it, because the applicant has made calculations using related parameters to designate which family members would receive the will; the designated contents of the will cannot be forged.

$$\left.\begin{aligned} m_{Ai} &= m_{willi}{}^{y_i} \bmod N_A \\ m_{Fi} &= m_{willi}{}^{x_i} \bmod N_{Fi} \end{aligned}\right\} \text{ for i=1 to n}$$

Because the will has the court's signature value, family members cannot forge a counterfeit will text.

$$\left.\begin{aligned} SG_{Ai} &= m_{Ai}{}^{d} \bmod N_C \\ SG_{Fi} &= m_{Fi}{}^{d} \bmod N_C \end{aligned}\right\} \text{ for i=1 to n}$$

3.1.4. Analysis of non-repudiation

The non-repudiation means one party is requested to provide related proof (such as digital signature) to the other party, which cannot be maliciously denied by the other party. In current applications, digital signatures are used to resolve the non-repudiation issue. In this section, we will explain how our structure fulfills demands of non-repudiation at every phase, the explanation is as follows:

(1) Non-repudiation in the registration phase:

In this phase, non-repudiation is achieved. In Table 1, we illustrate the non-repudiation proofs during this phase.

Table 1. Non-repudiation in the registration phase

Non-repudiation Evidence	Evidence Issuer	Evidence Holder	Verification Equation
(m_{req}, SG_{req})	Applicant (A)	Court (C)	$V_A(SG_{req}) \stackrel{?}{=} m_{req}$
(m_{req}, SG_C)	Court (C)	Applicant (A)	$V_C(SG_C) \stackrel{?}{=} m_{req}$

1. Since the Count received and verified the Applicant's signature SG_{req}, the Applicant cannot deny this registration.
2. Once the Court sends the application signature SG_C to applicant, the court cannot deny this transaction.

(2) Non-repudiation in the will to deliver and store phase:

In this phase, non-repudiation is achieved. In Table 2, we illustrate the non-repudiation proofs during this phase.

Table 2. Non-repudiation in the will to deliver and store phase

Non-repudiation Evidence	Evidence Issuer	Evidence Holder	Verification Equation
(m_{req}, SG_C)	Applicant (A)	Court (C)	$V_C(SG_C) \stackrel{?}{=} m_{req}$
(M_{will}, SG_{will})	Applicant (A)	Court (C)	$V_A(SG_{will}) \stackrel{?}{=} M_{will}$

1. Once the receiving the application for will deliver, the court can verify whether the applicant already registration or not.
2. When court received the applicant signature, the applicant cannot deny this transaction in the will to deliver and store phase.

(3) Non-repudiation in the apply for death certificate phase:

In this phase, non-repudiation is achieved. In Table 3, we illustrate the non-repudiation proofs during this phase. When hospital receive the folk's request signature value SG_{info}, the folk cannot deny this transaction in the apply for death certificate phase.

Table 3. Non-repudiation in the apply for death certificate phase

Non-repudiation Evidence	Evidence Issuer	Evidence Holder	Verification Equation
(m_{info}, SG_{info})	Folks(F)	Hospital (H)	$V_{Fi}(SG_{info}) \stackrel{?}{=} m_{info}$

(4) Non-repudiation in the decryption will text phase:

In this phase, non-repudiation is achieved. In Table 4, we illustrate the non-repudiation proofs during this phase.

Table 4. Non-repudiation in the decryption will text phase

Non-repudiation Evidence	Evidence Issuer	Evidence Holder	Verification Equation
$(h(C_{req}, ID_{Fi}), SG_{cert})$	Folks(F)	Court(C)	$V_{Fi}(SG_{cert}) \stackrel{?}{=} h(C_{req}, ID_{Fi})$
$(m_{Ai}, m_{Fi}, SG_{Ai}, SG_{Fi})$	Court(C)	Applicant (A)	$\begin{cases} V_C(S_{Ai}) \stackrel{?}{=} m_{Ai} \\ V_C(S_{Fi}) \stackrel{?}{=} m_{Fi} \end{cases}$

1. Since the court received and verified the folks' will ciphertext signature SG_{cret}, the folks cannot deny this request in the decryption will text phase.
2. When every folk receive the court signature value (SG_{Ai}, SG_{Fi}) on the will, the court cannot deny this transaction in the decryption will text phase.

3.1.5. Analysis of privacy

In this protocol, we added a security mechanism which based on discrete logarithm problem to encrypt the will text before transmitting to the court.

$$\left.\begin{aligned} m_{Ai} &= m_{willi}^{\ y_i} \bmod N_A \\ m_{Fi} &= m_{willi}^{\ x_i} \bmod N_{Fi} \end{aligned}\right\} \text{ for i=1 to n}$$

In this paper, even a third party (court) is involved to store and forward the will to each folk, it needs not to worry that the contents will be disclosed. The will text is secure and protected one's privacy. Furthermore, only if folk has the secret parameter which is given by the applicant before dying and along with the folk's self-selected parameter x_i to be able to access the will text. Thus, the privacy is protected throughout the entire process.

3.1.6. Analysis of attack patterns

(1) Attack by loss of death certificate:

If the death certificate applied for by family members using their identity code ID_{Fi} to the hospital has been lost or stolen, even though the thief could obtain the applicant's identity code ID_A and family member's identity code ID_{Fi}, and send it to the court along with the certificate of death, pretending to be a family member to apply for ciphertext of the will, this would not happen in our protocol, because family members need to use their own secret keys to sign the application request C_{req} (include the $ID_A, ID_{Fi}, Cert_{death}$)

$$C_{req} = E_C(ID_A, ID_{Fi}, Cert_{death})$$
$$SG_{cert} = S_{Fi}(h(C_{req}, ID_{Fi}))$$

After the court has received it, they can use the family member's public key to verify accuracy of the signature.

$$V_{Fi}(SG_{cert}) \stackrel{?}{=} h(C_{req}, ID_{Fi})$$

Here, our protocol adds the signature step, the main purpose of which is to prevent the problem of lost or stolen certificates of death. In this way, even though one has all related information and the certificate of death, without the secret key of family members for signature purposes, the court can detect illegality, thus one cannot receive the corresponding ciphertext of the will and signature.

(2) Internal attack:

In order to achieve the goals of will completeness and privacy described above, even though we save and transmit contents of the will through a third authority for the applicant, we are still concerned that there may be some internal attack within the authority institution. In our design, we added a security mechanism based on discrete logarithm problem. In it, the applicant first uses related parameters to calculate will text so that it becomes encrypted.

$$\left.\begin{aligned} m_{Ai} &= m_{willi}^{\ y_i} \bmod N_A \\ m_{Fi} &= m_{willi}^{\ x_i} \bmod N_{Fi} \end{aligned}\right\} \text{ for i=1 to n}$$

Without obtaining the secret parameter $(x_i, y_i, \alpha, \beta)$, no one can know contents of the will (of course this includes the members within the authority institution). This ensures security against internal attack.

(3) Impersonate attack:

In this protocol, if one family member knows that other family members may receive more assets and want to assume the identity of other family members, it would be impossible. In the related parameters given to each family member by the applicant, except for the most important y_i, which is different, the other related parameters (α, β) to decrypt the will text are the same.

$$\begin{aligned} C_{\text{sec1}} &= E_{F1}(Header_{will}, ID_C, ID_A, y_1, \alpha, \beta) \\ C_{\text{sec2}} &= E_{F2}(Header_{will}, ID_C, ID_A, y_2, \alpha, \beta) \\ &\vdots \\ C_{\text{secn}} &= E_{Fn}(Header_{will}, ID_C, ID_A, y_n, \alpha, \beta) \end{aligned}$$

After ensuring that each family member has his own y_i, it is necessary to return each family member's self-selected x_i to the applicant, and after the applicant is certain that related parameters have been received, x_i and y_i are used to calculate ciphertext of the will.

$$\left.\begin{aligned} m_{Ai} &= m_{willi}^{\ y_i} \bmod N_A \\ m_{Fi} &= m_{willi}^{\ x_i} \bmod N_{Fi} \end{aligned}\right\} \text{ for i=1 to n}$$

Since every family member has different secret parameter x_i, the escrow will signature can only be obtained $SG \equiv (m_{willi})^d \bmod N_C$, if one also has the parameters from the applicant y_i, α, β.

$$\begin{aligned} SG &\equiv SG_{Ai}^{\ \alpha} \ SG_{Fi}^{\ \beta} && \bmod N_C \\ &\equiv (m_{Ai}^{\ d})^{\alpha} \ (m_{Fi}^{\ d})^{\beta} && \bmod N_C \\ &\equiv (m_{willi}^{\ y_i d})^{\alpha} \ (m_{willi}^{\ x_i d})^{\beta} && \bmod N_C \\ &\equiv m_{willi}^{\ (\alpha y_i + \beta x_i) d} && \bmod N_C \\ &\equiv (m_{willi})^d && \bmod N_C \end{aligned}$$

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

(4) Man-in-the-middle attack:

In our protocol, if someone wants to steal information or transmit false information, signatures to others during the transmission processes, due to the entire transmission processes with verifiability and non-repudiation, thus the man-in-the-middle attacks cannot occur in our protocol. It enhances the security of online escrow will.

3.2. Discussion on practicality

The escrow will design in this paper can easily be applied to service on the World Wide Web. Our method allows living wills to be applied to existing online service protocols with security, practicality, and efficiency. The applicant can also go online at any time and place to entrust contents of the will. Unlike the paper wills we mentioned before, online wills can help save costs for lawyers and witnesses. Based on the basic premise of legal efficiency, the online escrow will system we propose is fast, convenient, and simple, and fulfills the needs and goal of practicality.

4. Conclusion

In our times, the internet is widely used; many online service systems have been proposed. In turn, the disadvantages of paper services have been found. This paper uses mechanisms such as digital signature, the encryption and decryption of public key, and certificates to achieve the security needs of completeness, verifiability, unforgeability, non-repudiation, and privacy, and can prevent various types

of attack patterns. The private online escrows will system in our proposal is enhanced in that our protocol is practical, and can lower costs and increase efficiency. In summary of the above, our protocol makes the following contributions:

- Can prevent family disputes resulting from fighting over the estate.
- As an online escrow will, it decreases costs and increases usage efficiency.
- Privacy considerations are protected by powerful security mechanisms, which can withstand various types of attack.
- Convenient preservation, the authority institution would save and transmit the will, meeting the function of the escrow will.
- Improves on the disadvantages of paper and meets various security requirements.

References

[1] Ajmani, S., R. Morris, and B. Liskov, "A Trusted Third-Party Computation Service." ***MIT Laboratory for Computer Science 200 Technology Square,*** **Cambridge**, pp. 512-521, Dec. 2001.

[2] Asokan, N., V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures, " ***IEEE Journal on Selected Areas in Communications***, Vol. 18, pp.593-610, 2000.

[3] Boneh, D., "Simplified OAEP for the RSA and Rabin functions." *In J. Kilian, editor*, CRYPTO'01. Vol. 12, pp.122-130, Springer-Verlag, 2001.

[4] Blake, I. F, and T. Garefalakis, "On the complexity of the discrete logarithm and Diffie–Hellman problems", ***Journal of Complexity***, Vol.20, pp.148-170, 2004.

[5] Cherepnev, M. A., "On the connection between the discrete logarithms and the Diffie–Hellman problem," *Discrete Math Application*., Vol. 6, pp.341-349, 1996.

[6] Chien, H. Y, and R. Y. Lin, "ID-Based E-Will System using Bilinear Pairings", ***IWCS2006***, Hong Kong, Jun. 20-22, 2006.

[7] Chen, T. H., W. B. Lee, and G. Horng, "Remarks on some signature schemes based on factoring and discrete logarithms," ***Applied Mathematics and Computation***, Vol. 169, No. 2, pp.1070-1075, Oct. 15, 2005.

[8] Camenisch, J. L., J.M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem." ***Advances in Cryptology-EUROCRYPT'94***, Perugia, Italy, pp.428-432, 1994.

[9] Damgard, I. B., "Practical and provably secure release of a secret and exchange of signatures." ***In Advance Cryptology-EUROCRYPT '93***, Vol. 765 of Lecture Notes in Computer Science, pp.200-217. Springer-Verlag, 1994.

[10] ElGamal, T., "A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms," ***IEEE Transaction on Information Theory***, Vol. 31, pp.469-472, 1985.

[11] Franklin, M. K, and M.K. Reiter, "Fair-Exchange with a Semi-Trusted Third Party," ***Proc. Fourth ACM Conf. Computer and Comm. Security***, Japan, pp. 1-7, Apr. 1997.

[12] Lee, W. B, and K. C. Liao, "Constructing Identity-Based Cryptosystems for Discrete Logarithm Based Cryptosystems," ***Journal of Network and Computer Applications***, Vol. 27, No. 4, pp. 191-199, Nov. 2004.

[13] Rivest, R. L., A. Shamir, and L. Adleman, "Method for Obtaining Digital Signatures and Public Key Cryptosystems," ***Communications of the ACM***, Vol. 21, No. 2, pp.120-126, 1978.

[14] Sigrid, G. R., C. Rudolph, and H. Vogt, "On the security of fair non-repudiation protocols" in Colin Boyd, Wenbo Mao (Eds.), ***ISC, Lecture Notes in Computer Science***, Vol. 2851, pp.193-207, Springer, 2003.

[15] Tseng, Y. M., "Digital signature of type and application", ***Communications of the CCISA***, Vol. 7, No.3, pp. 59-67, 2001.

[16] Zhou, J, and D. Gollmann, "A Fair Non-repudiation Protocol," ***Proc. IEEE Symp. Research in Security and Privacy***, pp. 55-61, 1996.

[17] Zhang, N., Q. Shi, and M. Merabti, "Revocation of privacy-enhanced public-key certificates," ***The Journal of Systems and Software,*** Vol.75, pp.205-214, 2005.

[18] Several Issues relating to the "Detailed Regulation on the Notarization of a Will", ***Justice of China***, No.6, 2005. http://scholar.ilib.cn/abstract.aspx?A=Zgsf200606019

[19] ***News morning paper*** "battle of a century of mystery succeeding to the throne" China Cantonese 2005. http://big5.xinhuanet.com/gate/big5/news.xinhuanet.com/st/2005-05/25/content_2998558.htm

[20] Housley, R., W. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile." ***PKIX Working Group Internet Draft***, April 2001. http://www.ietf.org/internet-drafts/draft-ietf-pkix-new-part1-06.txt