Asain Journal of Information Technology 4 (5): 541-549, 2005

Proxy Signature Scheme Basing on QR

Chen I-Te and Yi-Shiung Yeh
Department of Computer Science and Information Engineering,
National Chiao-Tung University, 1001, Ta-Hsueh Road, Hsinchu 30050, Taiwan

Abstract: Manbo proposed full delegation, partial delegation and delegation by warrant, three types of proxy signature scheme in 1996. The proxy signature scheme is used for an original signer delegating his capability to a designated signer, called a proxy signer. Compared to Manbo's scheme basing on discrete logarithm, Shao proposed a proxy signature scheme using the RSA cryptosystem in 2003. After Shao's scheme, M-S Hwang, S-F Tzeng and C-S Tsai proposed a proxy signature scheme basing on elliptic curves in 2004. This work proposes two novel proxy signature schemes basing on QR (Quadratic Residues). The first proxy signature scheme is full delegation method and the second one is delegation by warrant. Furthermore, the system administrator (SA) is introduced in the second scheme and the signature developed by Fan is applied. The SA designates qualified signers to sign warrants or documents. Those two novel proxy signature schemes basing on QR can take advantage of the QR property to achieve low-computing effects and to provide the same security level equal to those proxy signature basing on discrete logarithm, RSA or elliptic curves.

Key words: Proxy signature, Quadratic residues (QR), Legendre, Jacobin, Cryptography

Introduction

If original signers cannot sign a document themselves, they can delegate their signing capability to proxy signers. Since the scheme designed by Manbo does not provide strong undeniability (Lee and Kim, 1999 and Le *et al*., 2001) several scholars have tried to enhance it. For example, Kim *et al*. extended the proxy types using warrant information embedded in delegation (Hsu *et al*., 2001); Zhang and Hsu proposed a threshold proxy signature scheme (Kim *et al*., 1997 and Zhang, 1997). Moreover, Shao proposed a proxy signature scheme basing on RSA (Shao, 2003) and then, M-S Hwang, S-F Tzeng and C-S Tsai proposed a proxy signature scheme basing on elliptic curves in 2004 (Min *et al*., 2004).

Unfortunately, except Hwang's scheme basing on elliptic curves, above proxy signature schemes base on the discrete logarithm or RSA, which require extensive exponent calculations. Consequently, most schemes mentioned above might be unsuitable for use in mobile devices, which generally have only limited computing power. Our proposed proxy signature scheme basing on QR scheme is thus more efficient than other schemes basing on discrete logarithms or RSA. Moreover, the proposed scheme involves relatively few multiplications; therefore, the proposed scheme is ideal for low power and low computing device such as mobile phones, IC cards and so on.

Fundamentally, the presented proxy signature scheme bases on the QR assumptions for analyzing proxy signature security. The proposed scheme not only provides the designation property but also fulfills the following security requirements defined by B. Lee (Leeand Kim, 1999 and Lee *et al*., 1997).

Verifiability: Any verifier can verify the validity of a proxy signature either by a self-authentication form or an interactive form.

Strong Unforgeability: Nobody other than the proxy signer can create a proxy signature, neither the original signer.

Strong Identifiable: With the proxy signature, anybody can identify the corresponding proxy signer.

Strong Undeniability: The proxy signer cannot repudiate the proxy signature against anyone.

Proposed Scheme: The two proposed schemes comprise four phases: (1) initial phase, (2) proxy phase, (3) proxy-signature phase and (4) verification phase. First, some notations used in this paper are defined. The *Jacobin symbol* $[b/n]$ is defined as $\prod_{i=1}^{k}(b/p_i)^{e_i}$, where b is an integer and the *Legendre symbol* (b/p) is defined as [8]:

$(b/p)=0$	if $p \mid b$,
$(b/p)=1$	if b QR_n and
$(b/p)=-1$	if b quadratic nonresidue mod n.

Full Delegation Proxy Signature Scheme Basing on QR: An original signer and a proxy signer create and publish the necessary parameters in the initialization phase. In the proxy phase, the original signer signs on the hash value $h(m_w)$ of warrant information m_w. The symbol $h(\cdot)$ denotes as hash function and m_w indicates signature restriction such as a valid period. When original signer delivers the system key to the proxy signer, the proxy signer will identifies the original signer and verifies the system key.
In proxy-signing phase, the proxy signer signs on the document m, creates proxy signature and sends the proxy signature back to the requester. Finally, the verifier checks whether the signature is valid. The details of our scheme are presented as following.

Initialization Phase

Step 1: The original signer randomly chooses $n=p_1p_2p_3p_4$ which all p_i's are distinct large primes and $p_1\equiv p_2\equiv p_3\equiv p_4\equiv 3$.

Step 2: The original signer randomly chooses four elements $B=\{b_{ij} \mid i=\pm1; j=\pm1\}$ in Z_n^*. Such that $\gcd(n,b_{ij})=1$, $[b_{ij}/p_1]=i$ and $[b_{ij}/p_2]=j$ for every i and j in {1, -1}.

Step 3: The original signer computes $A= p_1p_2$. The system secrets are p_1, p_2, p_3 and p_4,and the system publics are (n,A).

Step 4: The original signer randomly selects x and y in Z_n^* as original private keys and publishes $Y= x^2+Ay^2 \bmod n$. The proxy signer randomly selects α and β in Z_n^* as proxy private keys and publishes $D=\alpha^2+A\beta^2 \bmod n$.

Proxy Phase

Step 1: The original signer computes $[h(m_w)Y/p_i]$ and sets the result as i_j, where $j=1,2,3\ 4$.
Step 2: The original signer selects a proper integer $b_p \in B$ such that

$$\begin{cases}[b_p / p_1] = i_1 \\ [b_p / p_2] = i_2\end{cases}$$

Step 3: The original signer randomly chooses u and v in Z_n^*, then computes $K=(u^2+Av^2) \bmod n$ such that

$$\begin{cases}[b_pK / p_3] = i_3 \\ [b_pK / p_4] = i_4\end{cases}$$

and computes

$$\begin{cases}c = (xu + Ayv) \bmod n \\ d = (yu - xv) \bmod n\end{cases}$$

As a result, $b_ph(m_w)(c^2+Ad^2)$ in QR_n.
Since the original signer knows p_1, p_2, p_3 and p_4, the signer can solves a square root t of $b_ph(m_w)(c^2+Ad^2)$ in Z_n^* with $O(\log n)$ complexity.

Step 4: The original signer sends (t, b_p, K, c, d) and (p_1, p_2, p_3, p_4) to the proxy signer in a secure manner. After receiving (t, b_p, K, c, d), the proxy signer examines the validity by checking whether the following equation holds.

$t^2YK \equiv b_ph(m_w)(c^2+Ad^2)^2$

Proxy-signing Phase

Step 1: The proxy signer selects a proper integer $b_s \in B$ and randomly chooses γ and ω, computes $C= \gamma^2+A\omega^2 \bmod n$ so that $b_st^2h(m)DC$ exists in QR_n.

Step 2: With p_1, p_2, p_3 and p_4, the proxy signer can resolve the root s of $b_st^2h(m)DC$ in Z_n^* with $O(\log n)$ complexity.

Step 3: The proxy signer computes

$$\begin{cases} e = (\alpha\gamma + A\beta\varpi) \bmod n \\ f = (\beta\gamma - A\alpha\varpi) \bmod n \end{cases}$$

and sends signature (s, b_p, b_s, K, e, f, C) of message m to the requester.

Verification Proxy-signing Phase: Any verifier examines the validity of signature (s, b_p, b_s, K, e, f, C) by checking if the following equation holds.

$s^2DC \equiv b_s b_p h(m_w) h(m) YK(e^2+Af^2)^2$.

Correctness Analysis: To verify the validity of the keys from signer, the proxy signer has to check whether $t^2YK \equiv b_p h(m_w)(c^2+Ad^2)^2$ is valid. If t is valid, then $t^2 \equiv b_p h(m_w)(c^2+Ad^2)$. We show as follows:
Substituting $c=xu+Ayv$ and $d=yu-xv$ into (c^2+Ad^2), we yield

$(xu+Ayv)^2+A(yu-xv)^2$
$\equiv_n (xu)^2+(Ayv)^2+2Axyuv+ A(yu)^2+ A(xv)^2-2Axyuv$
$\equiv_n (x^2+Ay^2)(u^2+Av^2)$
$\equiv_n YK$.

Assume that $[h(m_w)Y/p_j]=i_j$, where j=1, 2, 3, 4.
Choose a proper b_p and and γ,ω such that

$$\begin{cases} [b_p / p_1] = i_1 \\ [b_p / p_2] = i_2 \\ [b_p K / p_3] = i_3 \\ [b_p K / p_4] = i_4 \end{cases}$$

By the *Jacobi symbol*,
$[b_pK/p_1]$
$=[b_p (x^2+Ay^2) /p_1]$
$=[b_p/p_1] [x^2/p_1] = i_1$ and
$[b_p K /p_2]$
$=[b_p (x^2+Ay^2) /p_2]$
$=[b_p/p_2] [x^2/p_2]= i_2$.

Then,
$[b_p h(m_w) YK/p_j]$
$=[b_p h(m_w)(x^2+Ay^2)(u^2+Av^2)/p_j]$
$=[h(m_w)(x^2+Ay^2)/p_j] [b_p(u^2+Av^2)/p_j]$
$=(i_j)^2$
$=1$.
Therefore, $[h(m_w)Y /p_j][b_pK /p_j]$ is in QR_n. Form the derivation, we get

$t^2YK \equiv b_p h(m_w)(c^2+Ad^2)^2$.

Full Delegation Proxy Signature Scheme Basing on QR Security Requirements: To achieve the *verifiable* requirement, any verifier can verify the proxy signature. The following proves that our proposed scheme provides correct verification ability. Every valid proxy signature (s, b_p, b_s, K, e, f, C) satisfies

$s^2DC \equiv b_s b_p h(m_w) h(m) YK(e^2+Af^2)^2$

From the protocol, we can substitute e and f as

e^2+Af^2
$\equiv_n (\alpha\gamma + A\beta\varpi)^2 + (\beta\gamma - A\alpha\varpi)^2$

$\equiv_n(\alpha^2 + A\beta^2)(\gamma^2 - A\varpi)^2$
$\equiv_n DC$

We assume that $[b_p h(m_w)h(m)YD/p_j]= i_j$, j=1, 2, 3, 4. Choose a proper $b_s \in B$ and *u*, *v* such that

$$\begin{cases}[b_s / p_1] = i_1 \\ [b_s / p_2] = i_2 \\ [b_s C / p_3] = i_3 \\ [b_s C / p_4] = i_4 \end{cases}$$

By the procedures of Section 2.1.1, we can prove s^2 in QR_n. Therefore,
s^2DC
$\equiv_n b_s t^2 h(m)D^2C^2$
$\equiv_n b_s b_p h(m_w)h(m)(c^2+Ad^2)D^2C^2$
$\equiv_n b_s b_p h(m_w)h(m)YKD^2C^2$
$\equiv_n b_s b_p h(m_w)h(m)YK(e^2+Af^2)^2$.

The following discussion shows that the first proposed scheme meets the *strong unforgeability* requirement. In the proxy phase, if an attacker knows a proxy signature and forge a proxy (t, b_p, K, c', d'), the proxy signature will pass the verification. However, the attacker can hardly find $c' \equiv_n c$ and $d' \equiv_n d$ because with a given integer a in QR_n, it is infeasible to solve the root of a in Z_n^* without the parameters p_1, p_2, p_3 and p_4 (Fan and Lei, 2001). If an attacker randomly chooses c' and d' that can pass $t^2 \equiv_n b_p h(m_w)(c^2+Ad^2)$ verification, it will not pass $(c^2+Ad^2) \equiv_n YK$. Given an integer YK in QR_n, it is also infeasible to find a solution (a, b) of congruence $a^2+Ab^2 \equiv_n YK$ with gcd(n, A)≠1 (Pollard and Schnnor, 1987). A proxy signer uses the proxy signer's private keys α and β to create a proxy signature $s^2DC \equiv_n b_s b_p h(m_w)h(m)YK(e^2+Af^2)^2$ on document m. No one can forges e and f without knowing α and β (Pollard and Schnorr, 1987).
In the verification proxy-signing phase, anyone can identify the corresponding proxy signer by using the proxy signer public key to check whether $(e^2+Af^2) \equiv_n DC$. That means our proposed scheme achieve the *strong identifiability* property. Furthermore, the proxy signer cannot repudiate his signature against anyone because no one can create proxy signature without knowing the proxy private key and the system private key in polynomial time. Hence, the proposed scheme fulfills the *strong undeniability* property. Through the discussions above, we prove that our proposed scheme fulfills the security requirements that were defined by B. Lee (Lee and Kim, 1999 and Lee *et al.*, 2001).

Delegation by Warrant Proxy Signature Scheme Basing on QR: The first proposed scheme above cannot distinguish the signature of original signer from proxy signer's signature. Therefore, the SA is lead in the second one. The SA holds the secret and public system keys, which can grant the delegation capability to an original signer and the signing capability to a proxy signer, respectively. Therefore, the SA prevents misuse of unqualified proxy signers and improves the warrant mechanism used for negotiations between the original and proxy signers. Additionally, the SA takes responsibility for publishing the public keys of the original and proxy signers.
During the initial phase, the SA generates the secret and public system key pair. Meanwhile, both the original and proxy signers create the parameters required for signature authentication. Subsequently, the original signer signs a warrant information m_w in the proxy phase. The symbol $h(\bullet)$ used in the proxy phase denotes the hash function. Moreover, the symbol m_w indicates a proxy signature restriction such as the proxy valid period.
Within the proxy-signing phase, the proxy signer signs the document m and returns the proxy signature to the applicant. The verifier determines whether the proxy signature is valid during the final phase. Fig. 1 illustrates the whole phases and the following presents the details of the proposed scheme.

Initial Phase
Step 1: The SA selects the large prime numbers p_i with $p_i \equiv 3$ where $i=1, 2, 3, 4$ and lets $n= \prod_{i=1}^{4} p_i$. Thereafter, the SA sets $A=p_1p_2$ and assigns (n, A) and (p_1, p_2, p_3, p_4) as the system public and private keys respectively.

Step 2: The original signer specifies four elements as $B=\{b_{ij} | i=\pm1; j=\pm1\}$ in Z_n^* so that $[b_{i,j}/p_1]=i$ and $[b_{i,j}/p_2]=j$.

Step 3: The original signer selects x and y in Z_n^* as original private keys and sends public key $Y = (x^2+Ay^2) \bmod n$ to the SA thereafter. Relatively, the proxy signer selects £\ and £] in Z_n^* as proxy private keys and sends public key D =

$(\alpha\backslash^2+A\alpha]^2)$ mod n to the SA.

Step 4: The SA sends $[Y/p_i]=a_i$ and $[D/p_i]=r_i$, $i=1, 2, 3, 4$ to the original signer and the proxy signer respectively.

Step 5: The SA publishes the system, original and proxy's public key.

Proxy Phase

Step 1: The original signer sends warrant message m_w to the SA.

Step 2: The SA selects a proper integer $b_0 \in B$ such that

$$\begin{cases}[b_o / p_1] = a_1 \\ [b_o / p_2] = a_2\end{cases}$$

Step 3:The SA chooses $u=Z_n^*$ such that $v= h(u^*m_w)$ in Z_n^* and

$$\begin{cases}[b_o K / p_3] = a_3 \\ [b_o K / p_4] = a_4\end{cases}$$

where $K= (u^2+Av^2)$ mod n

Step 4: The SA sends (b_o, u, v) to the original signer. Thereafter the original signer lets

$$\begin{cases}c = (xu + Ayv)\text{mod}\square\square n \\ d = (yu - xv)\text{mod}\square\square n\end{cases}$$

Step 5: The original signer sends (c, d) to the SA. The SA uses system private key p_1, p_2, p_3 and p_4 to solve the square root t of $b_o(c^2+Ad^2)$ in $O(\log n)$ time complexity.

Step 6: The SA sends square root t to the original signer.

Step 7: The original signer sends (t, b_o, u, c, d, m_w) to the proxy signer. After receiving (t, b_o, u, c, d, m_w), the proxy signer examines whether $t^2YK\equiv_n b_o(c^2+Ad^2)^2$ or not. In addition, the proxy signer can compute $K= (u^2+Av^2)$ mod n with $v= h(u^*m_w)$ herself/ himself and retrieve the original singer's public key Y from the SA.

Proxy-signing Phase

Step 1: After receiving message m from an applicant, the proxy signer sends m to the SA.

Step 2: The SA selects a proper integer $b_p \in B$ such that

$$\begin{cases}[b_p / p_1] = r_1 \\ [b_p / p_2] = r_2\end{cases}$$

Step 3: The SA chooses £^$\in Z_n^*$ such that £s= h(£^*m) in Z_n^* and

$$\begin{cases}[b_p C / p_3] = r_3 \\ [b_p C / p_4] = r_4\end{cases}$$

where C =(£^^{2}+A£s^2) mod n

Step 4: The SA sends (b_p,£^,£s) to the proxy signer.

Step 5: The proxy signer lets

$$\begin{cases}e = (\alpha\gamma + A\beta\omega)\text{mod}\square\square n \\ f = (\beta\gamma - \alpha\omega)\text{mod}\square\square n\end{cases}$$

Step 6: The proxy signer sends (e, f) to the SA. The SA uses the system private key p_1, p_2, p_3 and p_4 to solve the square root s of $b_p t^2 DC$ with $O(\log n)$ time complexity.

Step 7: The SA sends square root s to proxy signer.

Step 8: The proxy signer sends the proxy signature $(s, b_o, b_p, u, e, f, £^, m_w)$ of message m back to the applicant.

Verification Phase
The verifier verifies whether $s^2DC \equiv_n b_o b_p YK(e^2+Af^2)^2$ to examine the validity of proxy signature $(s, b_o, b_p, u, e, f, £^, m_w)$. The verifier can retrieve $C=(£^2+A£s^2)$ and $K= (u^2+Av^2)$ automatically. Furthermore, $D = (£\backslash^2+A£]^2)$ mod n and $Y = (x^2+Ay^2)$ mod n are the public keys of the proxy and original signers, respectively.

Correctness Analysis: The proxy signer can verify the validity of delegation (t, b_o, u, c, d, m_w) by checking if $t^2YK \equiv_n b_o(c^2+Ad^2)^2$ where $K= (u^2+Av^2)$ mod n and $v= h(u^*m_w)$.
If $[b_oYK/p_i]=1$, where $i=1, 2, 3, 4$, then $t^2 \equiv_n b_o(c^2+Ad^2)$ belongs to QR_n. Therefore, the first step is to prove $[b_oYK/p_i]=1$, where $i=1, 2, 3, 4$. From the proxy phase, the SA sets $[Y/p_i] = a_i$, where $i=1, 2, 3, 4$. Also, the SA selects proper b_o and u, such that

$$\begin{cases} [b_o / p_1] = a_1 \\ [b_o / p_2] = a_2 \\ [b_o K / p_3] = a_3 \\ [b_o K / p_4] = a_4 \end{cases}$$

By the Jacobi symbol,

$[b_o K /p_1]$
$=[b_o(u^2+Av^2)/p_1]$
$=[b_o/p_1]\ [(u^2+Av^2)/p_1]$
$=[b_o/p_1]\ [u^2/p_1]$ Since $A = p_1p_2$
$=[b_o/p_1]$
$=a_1$

Similarly, $[b_o K /p_2] = a_2$.
Hence, $[b_o K /p_i]=a_i$, where $i=1, 2, 3, 4$.
$[b_oYK/p_i]$

$=[Y \qquad [b_oK$
$=(a_i)^2$
$=1$.

Therefore, $[b_oYKp_i]=1$, where $i=1, 2, 3, 4$.

Basing on proposed protocol, if t is valid, then $t^2 \equiv_n b_o(c^2+Ad^2)$.
Place $c \equiv_n xu+Ayv$ and $d \equiv_n yu-xv$ onto c^2+Ad^2, then we can compute

c^2+Ad^2
$\equiv_n (xu+Ayv)^2+A(yu-xv)^2$
$\equiv_n (xu)^2+(Ayv)^2+2Axyuv+A(yu)^2+A(xv)^2-2Axyuv$
$\equiv_n (xu)^2+(Ayv)^2+A(yu)^2+A(xv)^2$
$\equiv_n (x^2+Ay^2)(u^2+Av^2)$
$\equiv_n YK$.

Form the above derivation, the result is

$t^2YK \equiv_n b_o(c^2+Ad^2)^2$.

Delegation by Warrant Proxy Signature Scheme Basing on QR Security Requirements: Every valid proxy signature $(s, b_o, b_p, u, e, f,£^\wedge, m_w)$ satisfies $s^2DC \equiv_n b_o b_p YK(e^2+Af^2)^2$, which demonstrates the correctness of the proxy signature for achieving the ***verifiable*** requirement. First at all, use the same method mentioned above to make sure that $[b_p DC/p_i]=1$, where $i=1, 2, 3, 4$. Then, $s^2=b_p t^2 DC$belongs to QR_n. Thereafter, we substitute and $f = (\beta\gamma - \alpha\omega)$ into e^2+Af^2. It yields as following result.

e^2+Af^2
$\equiv_n (\alpha\gamma + A\varpi)^2 + A(\beta\gamma - \alpha\varpi)^2$
$\equiv_n (\alpha\gamma)^2 + (A\beta\varpi)^2 + 2A\alpha\beta\gamma\varpi + A(\beta\gamma)^2 - A(\alpha\varpi)^2 - 2A\alpha\beta\gamma\varpi$
$\equiv_n (\alpha\gamma)^2 + (A\beta\varpi)^2 + A(\beta\gamma)^2 - A(\alpha\varpi)^2$
$\equiv_n (\alpha + A\beta)^2 (\gamma^2 - A\varpi)^2$
$\equiv_n DC$

Therefore,
s^2DC
$\equiv_n b_p t^2 (DC)^2$
$\equiv_n b_p b_o (c^2+Ad^2) (DC)^2$
$\equiv_n b_o b_p YK(DC)^2$
$\equiv_n b_o b_p YK(e^2+Af^2)^2$.

The following discussion demonstrates that the second proposed scheme satisfies the *strong unforgeability* requirement. Attackers will encounter difficulty in solving the square root s and t without knowing the system private key (p_1, p_2, p_3, p_4) (Fan and Lei, 1996). Although the attackers could select a modulus pair (c', d') to pass $t^2 \equiv_n b_o(c'^2+Ad'^2)$ verification, it is still difficult for (c', d') pair to pass $(c'^2+Ad'^2) \equiv_n YK$ examination (Pollard and Schnorr) 1987). Accordingly, the attackers have difficulty in forging a proxy authentication (t, b_p, u, c, d, m_w) during the proxy phase.
The SA prevents unqualified original signers from delegating warrant; moreover, prevents unqualified proxy signers from signing a document. Consequently, the SA mechanism enhances the warrant mechanism. Additionally, the SA mechanism avoids original signers from abusing her/his delegation.
During the proxy-signing phase, proxy signers use their private keys £\ and £] to create a proxy signature $(s, b_o, b_p, u, e, f,£^\wedge, m_w)$ on document m. This security mechanism means that attackers cannot forge e and f unless they know £\ and £].
During the verification phase, any verifier can identify the corresponding proxy signer by using the public key of the proxy signer to check if $(e^2+Af^2) \equiv_n DC$. The proposed scheme thus satisfies the strong identifiability requirement.
Additionally, a proxy signer cannot repudiate that they are the issuers of their signature because no one can create a proxy signature during polynomial time without knowing the private keys of the system and proxy. Consequently, the proposed scheme fulfills the *strong undeniability* requirement. From above discussions, the proposed scheme meets the security requirements defined by B. Lee (Lee and Kim, 1999 and Lee *et al.*, 2001).

Time Complexity and Security Analysis: The complexity of the hash function can be negligible compared to that of the multiplication operation. The proposed proxy signature scheme does not use exponential and divisional operations throughout the proposed four phases. Consequently, an original signer and a proxy signer complete the proxy phrase in just 16 multiplications. During the proxy-signing phase, the proxy signer also uses just 5 multiplications to create a proxy signature. Only 17 multiplications are required to verify the validity of the proxy signature. The above computations are performed under Z_n^*. A modular exponent requires about 1024+512=1536 1024-bit modular multiplications; a 2048-bit modular multiplication is 8 times complexity than 1024-bit modular multiplication; and we ignore the negligible time complexity of addition operation in Table 1. Therefore, the proxy signature scheme basing on QR is more efficient than any other scheme basing on discrete logarithm.
The security of proposed schemes bases on QR assumption. Since $n=p_1 p_2 p_3 p_4$ and $A=p_1p_2$, how to choose (p_1, p_2, p_3, p_4) is very important. Comparing to the security level of 1024 bit RSA or discrete logarithms; the proposed schemes have to choose (p_1, p_2, p_3, p_4) such that n is around 2048 bit. Because the A is published, n is easy to be divided into A and $p_3 p_4$ ($n= A^* p_3 p_4$), where $A=p_1p_2$. To break the proposed scheme, RSA in 1024 bits for example, attackers have to reduce. As a result, the 2048 bit n is necessary. Furthermore, the multiplication in 2048 bit is still remarkable faster than exponential in 1024 bit shown in Table 1.

Table 1: Time complexity of Manbo's and proposed scheme

	Proxy phrase	Proxy-signing phase	Verification
Manbo's Scheme	1536+2=1538	1536×2+2=3074	1536+2=1538
Proposed Scheme (1024 bits)	16	5	17
Proposed Scheme (2048 bits)	†16×8=128	†5×8=40	†17×8=136

† For comparing in 1024-bit, we time 8 to keep time complexity consistency

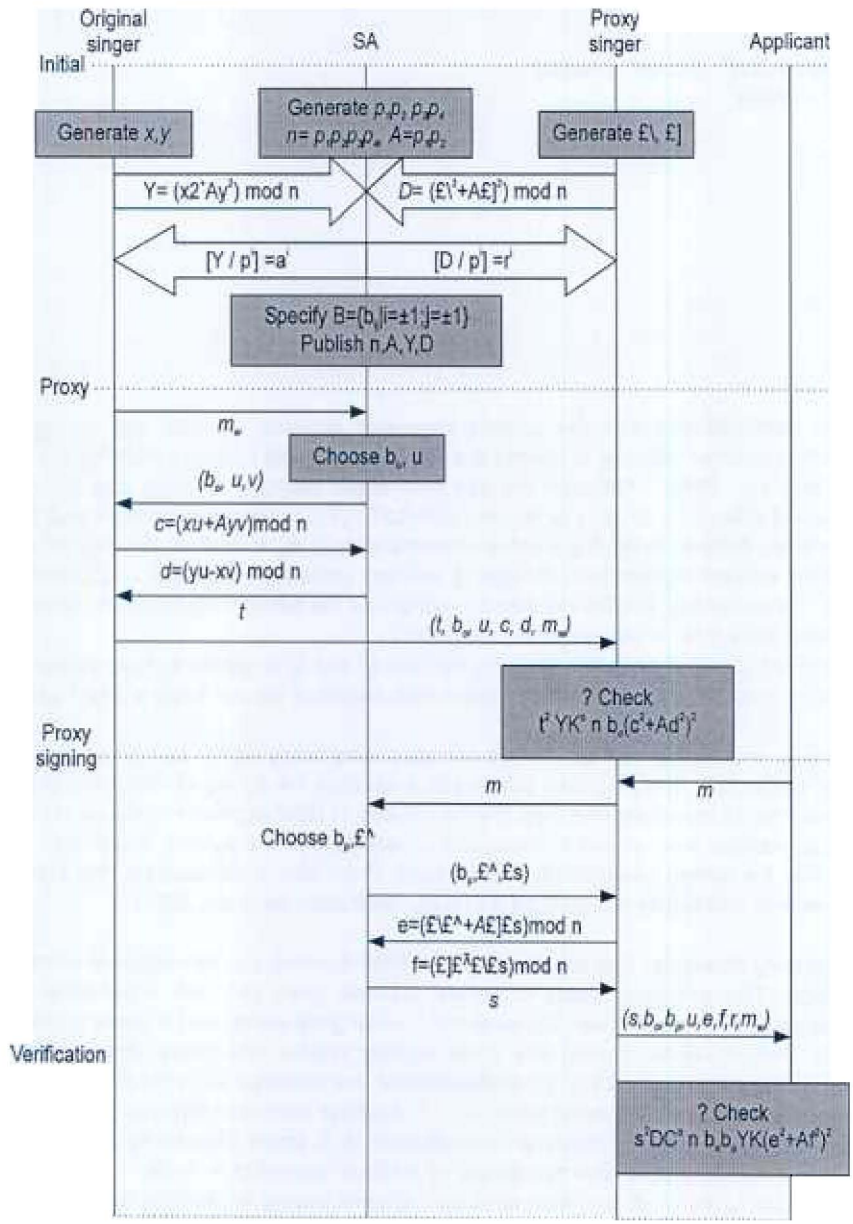

Fig. 1: Delegation by warrant proxy signature scheme basing on QR

Conclusion

Adopting Fan's signature, this work proposes two novel proxy signature schemes basing on the QR. Moreover, this work thoroughly discusses the properties of the first proposed scheme for fulfilling the requirements defined by B.

Lee. The SA holding the private system key can enhance the security of the second proposed scheme. The proposed schemes are computationally simpler than discrete logarithms or RSA. Accordingly, the proposed schemes are suitable for low computing capacity devices. However, Fan's signature uses too many parameters, so do our proposed schemes. In the future, we hope we can simplify the parameters of proposed schemes to save storage space.

References

Fan, C. I. and C. L. Lei, 1996. "Efficient blind signature scheme based on quadratic residues," Electronic Letters, 32: 811-813

Hsu, C. L., T. S. Wu and T. C. Wu, 2001. "New nonrepudiable threshold proxy signature scheme with known signers," The Journal of Systems and Software, 58:119-124

Min-Shiang Hwang, Shiang-Feng Tzeng and Chwei-Shyong Tsai, 2004. "Generalization of proxy signature based on elliptic curves," Computer Standards and Interfaces, 26: 73–84

Kim, S., S. Park and D. Won, 1997. "Proxy signatures, Revisited," Proc. of ICICS'97, Springer-Verlag, LNCS 1334, Pp: 223-232

Lee, B. and K. Kim, 1999. "Strong proxy signatures," IEICE Trans. Fundamentals, E82-A: 1-11

Lee, B., H. Kim and K. Kim, 2001. "Strong proxy signature and its applications," Proc. of SCIS 2001, 11B-1: 603-608

Mambo, M., K. Usuda and E. Okamoto, 1996. "Proxy signatures: Delegation of the power to sign messages, " IEICE Trans. Fundamentals, E79-A: 1338-1354

Alfred J. Menezes, Paul C. Van, Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996

Pollard, J. M., C. P. Schnorr, 1987. "An efficient solution of the congruence $x^2 + ky^2 = m(\text{mod } n)$," IEEE Trans. Information Theory, 33: 702-709, 1987

Shao, Z. H., 2003. "Proxy Signature Schemes Based on Factoring," Information Processing Letters 85, pp: :137-143

Zhang, K., 1997. "Threshold Proxy Signature Schemes," 1997 Information Security workshop, Japan, pp: 191-199