

A Proxy-Signature Scheme Based on Congruence Equation

Li Wei[†] Fan Mingyu

School of Computer Science & Engineering, University of Electronic Science and Tech of China,
 Chengdu, 610054 E-mail: 7ime@163.com

Abstract — In this paper, we proffer a proxy-blind-signature scheme based on the difficulty of big integer factorization using some characteristics of 2-order-equations, and the scheme can ensure security and efficiency, and is easily to be implemented in practice.

Index Terms — Digital Signature, Proxy-signature, Congruence Equation

multiply group based on modular n. If there exist $x, a \in \mathbb{Z}_n^*$ and $x^2 = a \pmod n$, then a is called 2-order-residual based on modular n, otherwise a is called 2-order-noresidual of n. The set of 2-order-residual is denoted with symbol Q_n and the set of 2-order-noresidual is denoted with symbol of Q_n' .

I. INTRODUCTION

A Proxy-Signature scheme means that a designated signer, which is often called proxy signer, can generate digital signature for a user. At present, most of the Proxy-Signature algorithms are based on the difficulty of discrete logarithm problems (such as Elgamal or Schnorr) and with low efficiency due to the computation of big number modular. So a means to improve the efficiency is to use some simple computation form, for example, more multiply or reverse rather than modular exponential computation.^[1]

In this paper, a new kind of Proxy-Signature scheme based on congruence equation is proposed, and its efficiency can be greatly enhanced compared with algorithms based on discrete logarithms.

II. BACKGROUND KNOWLEDGE^[2]

A. 2-order-residual

Let us use symbol n to denote a positive integer and \mathbb{Z}_n to denote a nonnegative integer less than n, $\mathbb{Z}_n^* = \{k \in \mathbb{Z}_n, \text{gcd}(k, n)=1\}$ is a

B. Legendre Symbol and Jacobi Symbol

Assume p is a odd prime number and a is an integer, then we can use symbol $\left(\frac{a}{p}\right)$ to denote

Legendre symbol, and its' value equal to 1 only if a is a 2-order-residual of p, otherwise its' value is -1. There is a formula was developed for computing Legendre symbol, which is

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \pmod p.$$

And the Jacobi symbol is a conception based on Legendre symbol, assume n is a odd positive

integer, and Jacobi symbol $\left(\frac{a}{n}\right)$ can be

$$\text{computed as } \left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) \left(\frac{a}{q}\right), \text{ here } n=pq,$$

there is a algorithm to compute $\left(\frac{a}{n}\right)$ without knowing p and q.

[†]Corresponding author, E-mail: 7ime@163.com. Supported by the National High-Tech Research and Development Plan of China (863) under Grant No 2009AA01Z403, 2009AA01Z435.

C. Some conclusions of 2-order-equations based on $GF(p)$

Lemma 1: Assume p is an odd prime integer and $a, b, c \in \mathbb{Z}_n^*$, then there are

$1 + \left(\frac{b^2 - 4ac}{p} \right)$ roots to equation $ax^2 + bx + c = 0 \pmod p$.

Lemma 2: Assume p is a prime integer and $f(x)$ is a n -orders-polynomial, if a is a root in $GF(q)$ for equation $f(x) = 0 \pmod p$, then $f(x) = (x-a)g(x)$ and $g(x)$ is a $n-1$ orders polynomial.

Lemma 3: Assume p is an odd prime integer and c is an element of $GF(q)$ which satisfy

$\left(\frac{c}{p} \right) = -1$, then we can use $\gcd(x^{p-1/2}, x^2 + mx + c) = x-a$ to solve equation $x^2 + mx + c = 0 \pmod p$,

here a is a root of the equation and $\left(\frac{a}{p} \right) = 1$.

III. PROXY-BLIND-SIGNATURE SCHEME BASES ON CONGRUENCE EQUATION

In our scheme ID_A is used to represent user A , which is an original signer, and its' private key is (p_A, q_A) , here p_A and q_A are secure big prime integer with the length at least 512 bits. At first, A should generate c_A and m_A which make

$$\left(\frac{c_A}{p_A} \right) = \left(\frac{c_A}{q_A} \right) = -1 \quad \text{and}$$

$$\left(\frac{m_A^2 - 4c_A}{p_A} \right) = \left(\frac{m_A^2 - 4c_A}{q_A} \right) = 1 \text{ hold. Then the}$$

public key of user A is (m_A, c_A, R_A) , and $R_A = p_A q_A$. There is also a proxy-signer S in our scheme, which can generate legitimate signature under the commission of user A . The private key of S is (p, q) , p and q is secure big prime integer, the public key is (R, c) with c satisfy

$$\left(\frac{c}{p} \right) = \left(\frac{c}{q} \right) = -1.$$

A. The Entrust Process

In order to make the proxy signer be eligible to generate legitimate signature instead of the original signer, the user A should first construct the equation $x^2 + m_A x + c_A = 0 \pmod R_A$, according Lemma 3 a root $a_A \in [0, R_A]$ can be easily found out, then the user's entrust information (ID_A, a_A) has been generated and was send to S . Upon receiving (ID_A, a_A) , S should first verify if a_A satisfy equation $x^2 + m_A x + c_A = 0 \pmod R_A$, if a_A is not a root of the equation, then the protocol will be halted automatically.

After verification, S will first compute $b = h(ID_A, a_A)$ using a strong hash function $h(\cdot)$,

then S should verify if $\left(\frac{b^2 - 4c}{p} \right) = 1$ and

$\left(\frac{b^2 - 4c}{q} \right) = 1$, if not, S will compute $b_i = b + i$,

$i = 1, 2, \dots$, the process will iterate until an appropriate b_i is found. Then S should construct modular equation $x^2 + b_i x + c = 0 \pmod R$, and a root s_i can be easily found according to Lemma 3, then compute $c_{AS} = h(b_i, s_i)$, and two secure big prime integers p_{AS} and q_{AS} were generated as the proxy-signature private key of user A , and its' proxy-signature public key is (R_A, c_{AS}) and $R_A = p_{AS} q_{AS}$.

B. The Signature Process

The signature process is varied according to different situations in our scheme. For example, before user A sending message to someone, its' signature will be required, and in some circumstance, someone else maybe has a piece of message and asks A to sign it, so we'll discuss the problem according to different circumstance:

Situation 1: Suppose A want to send message M to someone, A should first generate equation $x^2+Mx+c_A=0 \pmod{R_A}$, here M should satisfy

$$\left(\frac{M^2 - 4c_A}{p_A} \right) = \left(\frac{M^2 - 4c_A}{q_A} \right) = 1, \text{ if } M$$

doesn't satisfy the requirement, our process mentioned in the entrust process will be used, for simplicity, we will assume M always satisfy the requirement mentioned above.

Using Lemma 3, the user A can get a root of equation $x^2+Mx+c_A=0 \pmod{R_A}$, suppose it is m and was send to S, then the following process is implemented:

i: Verify if

$$\left(\frac{h(m)^2 - 4c_{AS}}{q_{AS}} \right) = \left(\frac{h(m)^2 - 4c_{AS}}{p_{AS}} \right) = 1,$$

otherwise compute $h_j = h(m)+j$, $j=1,2,\dots$, and the process iterate until an appropriate h_j is found, here $h(\cdot)$ is the same strong hash function used above;

ii: Generate signature function $x^2+h_jx+c_{AS}=0 \pmod{R_{AS}}$, and a root S_{AS} is worked out according Lemma 3, and (i, j, s_i, S_{AS}) was send back to A.

After the process, A will get its' proxy-signature, which is $(a_A, ID_A, i, j, s_i, S_{AS}, m, M)$.

Situation 2: At this circumstance it is simpler compared with situation 1, suppose someone else, here we called B for simplicity, have some message m and ask A to sign it, then the following process is implemented:

I: B generates $h(m)$ using a strong hash function and send $h(m)$ to S;

Ii: after receiving $h(m)$, S verify if

$$\left(\frac{h(m)^2 - 4c_{AS}}{q_{AS}} \right) = \left(\frac{h(m)^2 - 4c_{AS}}{p_{AS}} \right) = 1,$$

otherwise compute $h_j = h(m)+j$, $j=1,2,\dots$, and the process iterate until an appropriate h_j is found, here $h(\cdot)$ is the same strong hash function used above;

Iii: Generate signature function $x^2+h_jx+c_{AS}=0 \pmod{R_{AS}}$, and a root S_{AS} is worked out according Lemma 3, and (i, j, s_i, S_{AS}) was send back to B.

After the process, B can get S's proxy-signature for A, which is $(a_A, ID_A, i, j, s_i, S_{AS}, m)$.

C. The Verification Process

Verification process for situation 1:

- a: compute $b_i = h(ID_A, a_A) + i$, $h_j = h(m) + j$;
- b: verify if $s_i^2 + b_i s_i + c = 0 \pmod{R}$;
- c: verify if $S_{AS}^2 + h_j S_{AS} + c_{AS} = 0 \pmod{R_{AS}}$;
- d: verify if $m^2 + Mm + c_A = 0 \pmod{R_A}$;

Only if b、c and d all satisfied will the signature be accepted.

Verification process for situation 2:

- a: compute $b_i = h(ID_A, a_A) + i$, $h_j = h(m) + j$;
- b: verify if $s_i^2 + b_i s_i + c = 0 \pmod{R}$;
- c: verify if $S_{AS}^2 + h_j S_{AS} + c_{AS} = 0 \pmod{R_{AS}}$;

Only if b and c all satisfied will the signature is accepted.

IV. THE PROOF OF CORRECTNESS SECURITY AND EFFICIENCY

It is easy for us to know that our scheme is right. In the entrust process, A has constructed equation $x^2+m_Ax+c_A=0 \pmod{R_A}$, this is equal to the following modular equation group:

$$\begin{cases} x^2 + m_A x + c_A = 0 \pmod{p_A} \\ x^2 + m_A x + c_A = 0 \pmod{q_A} \end{cases}, \text{ and to legal}$$

user A, it is easily for him to solve each equation to get two roots x_1 and x_2 using Lemma 3 because he know p_A and q_A , then the following

$$\text{equation group was get: } \begin{cases} x = x_1 \pmod{p_A} \\ x = x_2 \pmod{q_A} \end{cases},$$

using the Chinese Remainder Theorem it is easily to get the appropriate root, and it is unique in $[0, R_A]$.

In the entrust process, (ID_A, a_A) is transmitted

in plaintext, for the difficulty of big integer factorization problem, it is difficult for anyone else to falsify a_A without being noticed, and any illegal participants can't reuse a_A since ID_A is unique.

The scheme can be implemented efficiently, for the main computation burden lies on the construction of congruence equation in the entrust process and signature process, for it needs to compute Legendre symbol, its' complexity is $O((\lg p)^2 + (\lg q)^2)$, and it is not very difficult for us to compute it.

In the entrust process and signature process, it is often required to rectify m_i according to

whether $\left(\frac{m_i^2 - 4c_i}{p_i}\right)$ and $\left(\frac{m_i^2 - 4c_i}{q_i}\right)$

equal 1, according to the character of 2-order-residual theorem, the probability is 0.25, so the process won't iterate too many times.

V. CONCLUSION

In this paper, we proffer a proxy-blind-signature scheme based on the difficulty of big integer factorization using some characteristics of 2-order-equations, and the scheme can ensure security and efficiency, and is easily to be implemented in practice.

REFERENCES

- [1] Cui Guoha, Ge Ping, "Signature Scheme Based on Big Integer Factorization", Computer Applications, pp 842-843, April, 2005.
- [2] IRELAND, ROSEN M, A Classical Introduction to Modern Number Theory, 2nd ed. Berlin: Springer-Verlag, 2004, 40-43.
- [3] ABE M, OKAMOTO T. Provably secure partially blind signature. [C]. Proceeding of Crypt'2000, Springer-Verlag, 2000:271-286.
- [4] GUO Tao, LI Zhitang, PENG Jianfen, et al. Blind signature and off-line e-cash system based on elliptic curve[J]. Journal of China Institute of

Communications, 2003,24(9): 142-146.

- [5] CAELLI W J, DAWSON E P, et, al. Elliptic curve cryptography and digital signature[J]. Computer and Security, 1999,18: 7-16.