Security Analysis and Design of Proxy Signature Schemes over Braid Groups

WEI Yun, XIONG Guo-Hua, ZHANG Xing-Kai, BAO Wan-Su

E-mail: weiyun456@sohu.com

Abstract: The braid groups have attracted much attention as a new platform of constructing cryptosystems. This paper firstly analyzes the security vulnerabilities of existing proxy signature schemes over braid groups and presents feasible attacks. Then a new proxy signature scheme is proposed based on the difficulty of the conjugacy search problem and the multiple conjugacy search problem. Security analysis shows that the proposed scheme satisfies the security requirements of proxy signature.

Key words: proxy signature; braid group; conjugacy search problem; multiple conjugacy problem

1 Introduction

The braid groups were first introduced by Artin[1]. Because of the non-commutativity property, the braid groups have become a new candidate to construct cryptosystem and attracted many cryptographers' attention. In 2000, they were first used to construct a key agreement protocol and a public key encryption scheme[2]. Since then there have been many attempts to design cryptographic primitives using braid groups. Positive proposals are key agreement protocols by Anshel et al[3], an implementation of braid computations by Cha et al[4], the first digital signature scheme by Ko et al[5], entity authentication schemes by Sibert et al and Lal et al[6, 7], public key encryption algorithm by Tang et al[8] and several digital signature schemes with additional properties[9-14].

The concept of proxy signature was introduced by Mambo, Usuda and Okamoto to allow a proxy signer to sign messages on behalf of an original signer who delegates his signing power to the proxy signer[15]. According to the delegation type, proxy signature schemes can be classified into full delegation, partial delegation and delegation by warrant schemes. Since Mambo et al.'s first scheme was published many proxy signature schemes have been proposed[16-21]. But using braid groups in the constructions of proxy signature schemes is still a new subject[12-14]. And Kumar claimed that some of them are not secure[22]. Hence, new constructions are desirable.

This paper analyzes the security weaknesses of the existing proxy signature schemes over braid groups and proposes a new scheme which satisfies the security requirements. The rest of this paper is organized as follows. The second section introduces the basics of braid groups and proxy signatures. The third section presents the security analysis of the previous proxy signature schemes over braid groups. The new scheme is proposed and analyzed in section 4 and the conclusion is drawn in section 5.

2 Preliminaries

2.1 Braid Group[2]

In this section, the basics of braid groups and hard problems in braid groups are introduced.

Definition 1 For each integer $n \geq 2$, the n-braid group B_n is an infinite non-commutative group which is defined as the group generated by $\sigma_1, \sigma_2, \cdots, \sigma_{n-1}$ with the relation:

(1) $\sigma_i \sigma_j = \sigma_j \sigma_i$ where $|i-j| \geq 2$

(2) $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$ otherwise.

The integer n is called the braid index and each element of B_n is called an n-braid. A braid is said to be positive if and only if it can be written as a product of generators $\sigma_1, \sigma_2, \cdots, \sigma_{n-1}$, i.e., no negative powers of $\sigma_1, \sigma_2, \cdots, \sigma_{n-1}$ are involved.

The identity $\varepsilon \in B_n$ is also regarded as positive. The positive braids in B_n form a semi-group B_n^+ which embeds into B_n. The fundamental braid $\Delta \in B_n$ is defined as: $\Delta = (\sigma_1\sigma_2\cdots\sigma_{n-1})(\sigma_1\sigma_2\cdots\sigma_{n-2})\cdots(\sigma_1\sigma_2)\sigma_1$.

A partial ordering $\leq$ on the elements of B_n is defined by setting $v \leq w$ if and only if there are positive braids $\alpha, \beta \in B_n^+$ satisfying $w = \alpha v \beta$. Any braid $\alpha \in B_n$ satisfying $\varepsilon \leq \alpha \leq \Delta$ is called a canonical factor. A factorization $\gamma = \alpha\beta$ of a positive braid γ into a canonical factor α and a positive braid β is said to be left-weighted if α has the maximal word length among all such factorizations. Every braid $w \in B_n$ can be written uniquely as $w = \Delta^r \alpha_1 \cdots \alpha_q$ such that $\alpha_1, \cdots, \alpha_q$ are canonical factors and $\alpha_i \alpha_{i+1} (1 \leq i < q)$ is left-weighted. And r, denoted by $\inf(w)$, is the greatest integer j satisfying $\Delta^j \leq w$. $r+q$, denoted by $\sup(w)$, is the smallest integer j satisfying $w \leq \Delta^j$. q is called the canonical length of w.

Let LB_n and RB_n be the subgroups of B_n generated by $\sigma_1, \sigma_2, \cdots, \sigma_{\lfloor n/2 \rfloor -1}$ and $\sigma_{\lfloor n/2 \rfloor +1}, \sigma_{\lfloor n/2 \rfloor +2}, \cdots, \sigma_{n-1}$, respectively. Then the commutativity property holds for any $\alpha \in LB_n$ and $\beta \in RB_n$, i.e. $\alpha\beta = \beta\alpha$.

Two braids $\alpha, \beta \in B_n$ are said to be conjugate if there exist a braid s such that $\beta = s^{-1}\alpha s$. And by $\alpha \sim \beta$ we mean α and β are conjugate. There are some mathematically hard problems related to conjugation over braid groups which can be used to design cryptographic protocols.

Definition 2 Conjugacy Decision Problem (CDP)

Instance: $(\alpha, \beta) \in B_n \times B_n$ such that $\beta = s^{-1}\alpha s$ for some $s \in B_n$.

Objective: Determine whether α and β are conjugate or not.

Definition 3 Conjugacy Search Problem (CSP)

Instance: $(\alpha, \beta) \in B_n \times B_n$ such that $\beta = s^{-1}\alpha s$ for some $s \in B_n$.

Objective: Find $t \in B_n$ such that $\beta = t^{-1}\alpha t$.

Definition 4 Multiple Conjugacy Search Problem (MCSP)

Instance: $(\alpha_1, s^{-1}\alpha_1 s), \cdots, (\alpha_N, s^{-1}\alpha_N s) \in B_n \times B_n$ for some $s \in B_n$.

Objective: Find $t \in B_n$ such that $t^{-1}\alpha_1 t = s^{-1}\alpha_1 s, \cdots, t^{-1}\alpha_N t = s^{-1}\alpha_N s$.

Definition 5 Root Problem (RP)

Instance: A positive integer $c (c > 1)$ and a braid $\beta \in B_n$ such that $\beta = \alpha^c$ for some $\alpha \in B_n$.

Objective: Find $\gamma \in B_n$ such that $\beta = \gamma^c$.

There is an efficient polynomial time algorithm for solving CDP[5]. And many algorithms have been proposed to solve CSP, its variants and RP[23-29]. But none of them was proved to be polynomial in solving CSP, MCSP or RP. So these hard problems are still used to develop cryptosystems.

2.2 Proxy Signature

A secure proxy signature scheme should satisfy the following security requirements[16].

Strong unforgeability: Only the designated proxy signature can generate valid proxy signatures. The original signer and any third party who is not designated as proxy signer can not create a valid proxy signature.

Verifiability: After the verification, the verifier can be convinced of the original signer's agreement on the signed message.

Strong identifiability: Anyone can identify both the original signer and the proxy signer from the proxy signature.

Strong undeniability: Once the proxy signer generates a valid proxy signature on behalf of the original signer, he can not deny the signature.

Prevention of misuse: The proxy signer can not misuse his delegation power.

2.3 Notations

In this paper, $a \in_R A$ denotes a random choice of an element a from the set A. For $m_1, m_2 \in \{0,1\}^*$, we mean the concatenation of m_1, m_2 by $m_1 \| m_2$. And $P_1 \Rightarrow P_2$ means P_2 holds if P_1 holds.

3 Security Analysis of Existing Proxy Signature Schemes over Braid Groups

The system parameters n and l are positive integers large enough. Let

$$B_n(l)=\{b\in B_n \mid 0\le \inf(b)\le \sup(b)\le l\}$$
$$LB_n(l)=\{b\in LB_n \mid 0\le \inf(b)\le \sup(b)\le l\}$$
$$RB_n(l)=\{b\in RB_n \mid 0\le \inf(b)\le \sup(b)\le l\}$$

Then $|B_n(l)|\le l(n!)^l$ and $B_n(l)$, $LB_n(l)$, $RB_n(l)$ are finite set[12]. $H_1: B_n\to\{0,1\}^*$, $H_2:\{0,1\}^*\to B_n$ are collision resistant one way hash functions. $m\in\{0,1\}^*$ is the message to be signed.

3.1 Verma's Scheme and Security Analysis

In this section, the security analysis of Verma's proxy signature scheme is given.

3.1.1 The Scheme[12]

Key Generation:

The original signer chooses $a_o, x_o\in_R B_n(l)$ and computes $x_o'=a_ox_oa_o^{-1}$. Then a_o is the secret key and (x_o, x_o') is the public key. The secret and public keys of the proxy signer, a_p and (x_p, x_p'), are generated similarly.

Delegation:

The original signer chooses a braid $z_o\in_R B_n(l)$ and computes $t_o=a_oz_oa_o^{-1}$ and sends (z_o, t_o) to the proxy signer. The proxy signer checks $t_ox_o'\sim z_ox_o$.

Proxy Signature Generation:

The proxy signer chooses a braid $b\in_R B_n(l)$ and computes $h=H_2\left(H_1(t_ox_o')\,\|\,m\right)$, $\gamma=bhb^{-1}$, $\delta=bx_pb^{-1}$, $\theta=ba_p^{-1}ha_pb^{-1}$. Then $(t_o,\gamma,\delta,\theta)$ is displayed as a proxy signature on message m.

Verification:

On receiving $(t_o,\gamma,\delta,\theta)$ the verifier computes $h=H_2\left(H_1(t_ox_o')\,\|\,m\right)$ and checks whether $\gamma\sim h$, $\delta\sim x_p$, $\theta\sim h$, $\gamma\delta\sim hx_p$ and $\delta\theta\sim x_p'h$ hold. If the equations hold he accepts the signature.

3.1.2 Security Analysis

Kumar claimed that in the above scheme, there is nothing about the identity information of the original signer and the proxy signer, the limit on the delegated messages and the duration period of delegated power contained in the delegation pair. Hence, the proxy signer is able to misuse his delegated capabilities and the scheme is not secure against the original signer and proxy signer changing attacks[22].

Besides the security vulnerabilities Kumar pointed out, the scheme is vulnerable to forgery attacks mounted by any attacker. We will show that once an attacker gains a valid proxy signature on message m, he can forge another proxy signature on m passing the verification.

Assume that the signature on m he gains is $(t_o,\gamma,\delta,\theta)$. He chooses $b'\in_R B_n$ and computes $\gamma'=b'\gamma b'^{-1}$, $\delta'=b'\delta b'^{-1}$, $\theta'=b'\theta b'^{-1}$. Then the new proxy signature is $(t_o,\gamma',\delta',\theta')$, which can pass the verification because $\gamma'\sim h$, $\delta'\sim x_p$, $\theta'\sim h$, $\gamma'\delta'\sim hx_p$ and $\delta'\theta'\sim x_p'h$.

3.2 Zhang and Zeng's Scheme and Security Analysis

In this section, the security analysis of Zhang and Zeng's proxy signature scheme is given.

3.2.1 The Scheme[13]

Key Generation: same as section 3.1.1.

Delegation:

The original signer computes $t_o=a_oH_2(ID_P)a_o^{-1}$ and sends t_o to the proxy signer, where ID_P denotes the identity information of the proxy signer. The proxy signer checks $t_o\sim H_2(ID_P)$ and $t_ox_o'\sim H_2(ID_P)x_o$.

Proxy Signature Generation:

The proxy signer randomly chooses an odd prime c and computes $\gamma=t_o^c$, $\delta=a_p^{-1}H_2(m)a_p$. Then (c,γ,δ) is displayed as a proxy signature on message m.

Verification:

On receiving (c,γ,δ) the verifier checks whether $\gamma\sim H_2(ID_P)^c$, $\gamma x_o'\sim H_2(ID_P)^c x_o$, $\delta\sim H_2(m)$ and $\delta x_p'\sim H_2(m)x_p$ hold. If the equations hold he accepts the signature.

3.2.2 Security Analysis

In this scheme, although for messages on which the proxy signer has not signed the original signer can not generate valid proxy signatures, he can forge valid proxy signatures on the messages on which proxy signatures have been displayed. t_o is generated by the original signer. Once he gains a valid proxy signature (c, γ, δ) on m he can choose a random odd prime c' and compute $\gamma' = t_o^{c'}$. Then he displays (c', γ', δ) as another signature on m , which can pass the verification because $\gamma' \sim H_2(ID_P)^{c'}$, $\gamma' x'_o \sim H_2(ID_P)^{c'} x_o$, $\delta \sim H_2(m)$ and $\delta x'_p \sim H_2(m) x_p$.

Zhang and Zeng claimed that for attackers other than the original signer, to compute t_o they have to solve the root problem. But the fact is that any attacker with two valid proxy signatures can obtain t_o , which is explained below.

Let these two signatures be $(c_1, \gamma_1, \delta_1)$ and $(c_2, \gamma_2, \delta_2)$. Both c_1 and c_2 are primes. So there exist two integers d_1 and d_2 such that $c_1 d_1 + c_2 d_2 = 1$. After computing d_1 and d_2 , the attacker can get t_o by computing $(\gamma_1)^{d_1} (\gamma_2)^{d_2} = t_o$. Then the attacker can mount the forgery attack described above.

3.3 Lal and Verma’s Scheme and Security Analysis

In this section, the security analysis of Lal and Verma’s proxy signature scheme is given.

3.3.1 The Scheme[14]

Key Generation: same as section 3.1.1.

Delegation:

The original signer chooses a braid $z_o \in_R B_n(l)$ and computes $t_o = a_o z_o a_o^{-1}$ and sends (m_w, z_o, t_o) to the proxy signer, where m_w is the warrant on the message $m \in \{0,1\}^*$, which is to be signed, consisting of the identities of the original singer, the proxy signer and the period of delegation.

The proxy signer checks $t_o x'_o \sim z_o x_o$. If it holds he computes the proxy key $PK = a_p t_o a_p^{-1}$.

Proxy Signature Generation:

The proxy signer chooses a braid $b \in_R B_n(l)$ and computes $h = H_2\left(H_1(t_o x'_o) \oplus m_w\right)$, $\gamma = bhb^{-1}$, $\delta = b x_p b^{-1}$, $\theta = b a_p^{-1} (PK) a_p b^{-1}$. Then $(m_w, t_o, \gamma, \delta, \theta)$ is displayed as a signature on message m .

Verification:

On receiving $(m_w, t_o, \gamma, \delta, \theta)$ the verifier computes $h = H_2\left(H_1(t_o x'_o) \oplus m_w\right)$ and checks whether $\gamma\theta \sim h t_o$ and $\gamma\delta \sim h x_p$ hold. If the equations hold he accepts the signature.

3.3.2 Security Analysis

The authors claimed that the above scheme satisfies all the security requirements. Actually the scheme doses not satisfy the unforgeability, which is explained as follows.

From $\theta = b a_p^{-1} (PK) a_p b^{-1} = b a_p^{-1} a_p t_o a_p^{-1} a_p b^{-1} = b t_o b^{-1}$ we know that θ does not contain any information about the secret key of the proxy signer. h is computed from m_w, t_o and a public value x'_o , which are known after a display of a valid proxy signature. γ is computed from a random braid b and h . Similarly δ is computed from the random braid b and a public value x_p . No item of the signature contains any information about the secret key of the proxy signer. Hence, the original signer can generate a valid proxy signature on m easily. Besides the original signer, anyone with a valid proxy signature can forge a proxy signature on m passing the verification because (m_w, t_o) is revealed.

Hence, Lal and Verma’s scheme is not secure at all.

4 A New Proxy Signature Scheme

In this section, a new proxy signature scheme is proposed and the security analysis is given.

4.1 The Scheme

Key Generation:

The original signer chooses $a_o \in_R LB_n(l)$, $x_o \in_R B_n(l)$ and computes $x'_o = a_o x_o a_o^{-1}$. Then a_o is the secret key and (x_o, x'_o) is the public key. The secret and public keys of the proxy signer, a_p and (x_p, x'_p) , are generated similarly.

Delegation:

The original signer computes $w = a_o^{-1} x'_p a_o$, $z_o = H_2\left(m_w \,||\, H_1(w)\right)$, $t_o = a_o z_o a_o^{-1}$ and sends (m_w, t_o) to the proxy signer in a secure way, where m_w is the warrant which consists of the identities of the original signer and the proxy signer, the period of the delegation, the qualification of the messages on which the proxy signer can sign and w .

The proxy signer computes $z_o = H_2\left(m_w \,\|\, H_1(w)\right)$ and checks $t_o x'_o \sim z_o x_o$.

Proxy Signature Generation:

The proxy signer chooses $b \in_R RB_n(l)$ and computes $\alpha = b x_p b^{-1}$, $h = H_2\left(m \,\|\, H_1(\alpha)\right)$, $\beta = t_o a_p b t_o^{-1} h t_o b^{-1} a_p^{-1} t_o^{-1}$, $\gamma = t_o b x'_p b^{-1} t_o^{-1}$, $\theta_1 = b t_o^{-1} h t_o b^{-1}$, $\theta_2 = b w b^{-1}$ and displays $(m_w, \alpha, \beta, \gamma, \theta_1, \theta_2)$ as a proxy signature on message m .

Verification:

On receiving $(m_w, \alpha, \beta, \gamma, \theta_1, \theta_2)$ the verifier computes $z_o = H_2\left(m_w \,\|\, H_1(w)\right)$, $h = H_2\left(m \,\|\, H_1(\alpha)\right)$ and accepts the signature if and only if $\alpha \sim x_p$, $\beta \sim h$, $\beta\gamma \sim \theta_1 \alpha$, $\gamma \sim x'_p$, $\gamma x'_o \sim z_o \theta_2 z_o^{-1} x_o$.

4.2 Security Analysis

Correctness:

The following equations prove the correctness of the signature scheme:

$$t_o x'_o = a_o z_o a_o^{-1} a_o x_o a_o^{-1} = a_o z_o x_o a_o^{-1} \Rightarrow t_o x'_o \sim z_o x_o$$

$$a_o, a_p \in LB_n, b \in RB_n \Rightarrow a_p b = b a_p, a_p^{-1} b^{-1} = b^{-1} a_p^{-1}, a_o^{-1} b = b a_o^{-1}, b^{-1} a_0 = b^{-1} a_0$$

$$\alpha = b x_p b^{-1} \Rightarrow \alpha \sim x_p$$

$$\beta = t_o a_p b t_o^{-1} h t_o b^{-1} a_p^{-1} t_o^{-1} \Rightarrow \beta \sim h$$

$$\gamma = t_o b x'_p b^{-1} t_o^{-1} \Rightarrow \gamma \sim x'_p$$

$$\begin{aligned} & a_p b = b a_p, a_p^{-1} b^{-1} = b^{-1} a_p^{-1} \\ \Rightarrow & \beta\gamma = t_o a_p b t_o^{-1} h t_o b^{-1} a_p^{-1} t_o^{-1} t_o b x'_p b^{-1} t_o^{-1} = t_o a_p b t_o^{-1} h t_o b^{-1} a_p^{-1} b a_p x_p a_p^{-1} b^{-1} t_o^{-1} \\ & \quad = t_o a_p b t_o^{-1} h t_o b^{-1} a_p^{-1} a_p b x_p b^{-1} a_p^{-1} t_o^{-1} = t_o a_p (b t_o^{-1} h t_o b^{-1})(b x_p b^{-1}) a_p^{-1} t_o^{-1} = t_o a_p \theta_1 \alpha a_p^{-1} t_o^{-1} \\ \Rightarrow & \beta\gamma \sim \theta_1 \alpha \end{aligned}$$

$$\begin{aligned} & a_o^{-1} b = b a_o^{-1}, b^{-1} a_0 = b^{-1} a_0 \\ \Rightarrow & \gamma x'_o = (a_o z_o a_o^{-1}) b x'_p b^{-1} (a_o z_o^{-1} a_o^{-1})(a_o x_o a_o^{-1}) \\ & \quad = a_o z_o a_o^{-1} b x'_p b^{-1} a_o z_o^{-1} x_o a_o^{-1} = a_o z_o b a_o^{-1} x'_p a_o b^{-1} z_o^{-1} x_o a_o^{-1} \\ & \quad = a_o z_o b w b^{-1} z_o^{-1} x_o a_o^{-1} = a_o (z_o \theta_2 z_o^{-1} x_o) a_o^{-1} \\ \Rightarrow & \gamma x'_o \sim z_o \theta_2 z_o^{-1} x_o \end{aligned}$$

Strong unforgeability:

In $(m_w, \alpha, \beta, \gamma, \theta_1, \theta_2)$, β is dependent on t_o and the secret key of the proxy signer a_p . t_o is only known to the original signer and the proxy signer. The original signer can compute $t_o^{-1} \beta t_o = a_p b t_o^{-1} h t_o b^{-1} a_p^{-1}$ and $t_o^{-1} \gamma t_o = b x'_p b^{-1} = a_p \alpha a_p^{-1}$. It is multiple conjugacy search problem to extract a_p from $t_o^{-1} \beta t_o$, $t_o^{-1} \gamma t_o$ and x'_p . So the original signer can not get the secret key of the proxy signer. Then he can not forge a signature.

For an attacker other than the original signer, it is more difficult to make a forgery because he has to get both the secret keys of the original signer and the proxy signer. It is multiple conjugacy search problem to extract a_o from w and a'_o . And extracting a_p from a'_p is the conjugacy search problem.

Hence, the proposed proxy scheme satisfies the strong unforgeability under the assumption that the conjugacy search problem and the multiple conjugacy search problem are hard to solve.

Verifiability: m_w and the public key of the original signer appear in the verification. Hence, the verifier can be convinced that the original signer agrees on the signed message.

Strong identifiability: The identities of the original signer and the proxy signer are contained in m_w and the public keys of both signers appear in the verification equations $\alpha \sim x_p$, $\gamma \sim x'_p$, $\gamma x'_o \sim z_o \theta_2 z_o^{-1} x_o$. Anyone can easily identify the original signer and the proxy signer.

Strong undeniability: Since the proxy signature depends on the delegate key t_o and the secret key of the proxy signer a_p , no one else can forge a valid proxy signature. The proxy signer can not deny his behavior if a signature passes the verification.

Prevention of misuse: Since the delegation warrant consists the period of the delegation and the qualification of the messages on which the proxy signer can sign, the proxy signer can not misuse his delegated signing capabilities.

Some remarks on $w = a_o x'_p a_o^{-1}$ **:** In the scheme, the proxy signer can not verify whether w contains the information of the original signer's secret key. But it does not bring any security weakness to the scheme. The only motivation for the original signer to deceive the proxy signer with a wrong w is that he can forge a valid proxy signature which is infeasible without the secret key of the proxy signer. So we can deduce that it is not a profitable decision for the original signer to send a wrong w .

5 Conclusions

As a new platform for constructing cryptosystem, the braid groups have become a hot research topic. Several proxy signature schemes were constructed using braid groups. But none of them is secure. This paper analyzes the security vulnerabilities of the existing proxy signatures schemes over braid groups and proposes a new proxy signature scheme based on the difficulty of the conjugacy search problem and the multiple conjugacy search problem. Security analysis shows that the proposed scheme satisfies the security requirements of proxy signature.

Acknowledgement We thank all the people providing valuable suggestions.

References:

[1] Artin E. Theory of braids, Annals of Math , 1947, 101-126.

[2] Ko K H, Lee S J, Cheon J H, et al. New public key cryptosystem using Braid groups. In: Proceedings of Crypto-2000, Lecture Notes in Computer Science, Springer-Verlag, 2000, 1880: 166-183..

[3] Anshel I, Anshel M, Fisher B, et al.. New key agreement protocol in braid group cryptography. Topics in Cryptology- CT- RSA 2001, Lectures in Computer Science, Benlin: Springer Verlag, 2001, 2020: 1-15.

[4] Cha J C, Ko K H, Lee S J, et al. An efficient implementation of braid groups. In: Advances in Cryptology: Proceedings of ASIACRYPT 2001, Lecture Notes in Computer Science, Springer-Verlag, 2001, 2248: 144-156.

[5] Ko K H, Choi D H, Cho M S, et al. New signature scheme using conjugacy problem. http://eprint.iacr.org/2002/168.

[6] Sibert H, Dehornoy P, Girault M, Entity authentication schemes using braid word reduction. http://eprint.iacr.org/2002/187.

[7] Lal S and Chaturvedi A. Authentication schemes using braid groups. http://arXiv.org/cs.CR/0507066.

[8] Tang X M, Hong F and Cui G H. A public key encryption algorithm on braid groups. Journal of Software, 2007, 18(3): 722-729.

[9] Thomas T, Lal A K. Group Signature Scheme Using Braid Groups. http://arXiv.org/cs.CR/0602063.

[10] Zou S H, Zeng J W and Quan J J. Designated verifier signature scheme based on braid groups. http://eprint.iacr.org/2006/329.

[11] Verma G K. Blind signature schemes over Braid groups. http://eprint.iacr.org/2008/027.

[12] Verma G K. A proxy signature scheme over braid groups. http://eprint.iacr.org/2008/160.

[13] Zhang L L, Zeng J W. Proxy signature based on braid group. Journal of Mathematical Study, 2008, 41(1): 56-64.

[14] Lal S and Verma V. Some Proxy Signature and Designated Verifier Signature Schemes over Braid Groups. http://arXiv.org/cs.CR/09043422.

[15] Mambo M, Usuda K and Okamoto E. Proxy signature: delegation of the power to sign messages. IEICE Trans. Fundamentals, 1996, E79-A(9): 1338-1353.

[16] Zhang F and Kim K. Efficient ID-based blind signature and proxy signature from bilinear pairings. In: Proceedings of ACISP 03, Lectures in Computer Science, Benlin: Springer Verlag, 2003, 2727: 312-323.

[17] C. Gu, Y. Zhu. Provable Security of ID-based Proxy Signature Schemes. In: Proceedings of ICCNMC'05, Lectures in Computer Science, Benlin: Springer Verlag, 2005, 3619: 1277-1286.

[18] Gu C X and Zhu Y F. An efficient ID-based proxy signature scheme from pairing. http://eprint.iacr.org/2006/158.

[19] Wu W, Mu Y, W Susilo, et al. Identity-based proxy signature from pairing. In: Proceedings of ATC 2007, Lectures in Computer Science, Benlin: Springer Verlag, 2007, 4610: 22-31.

[20] Jacob C N, K. Matsuura and Paterson K G. Proxy Signatures secure against proxy key exposure. In: Proceedings of PKC 2008, Lectures in Computer Science, Benlin: Springer Verlag, 2008, 4939: 141-161.

[21] Wang B. A new identity based proxy signature scheme. http://eprint.iacr.org/2008/323.

[22] Kumar J. Security analysis of a proxy signature scheme over braid groups. http://eprint.iacr.org/2009/158.

[23] Garber D, Kaplan S, et al. Length-based conjugacy search in the braid group. http://arXiv.org/math. GR/0209267.

[24] Hofheinz D, Steinwandt R. A Practical Attack on Some Braid Group Based Cryptographic primitives. In: Proceedings of PKC2003, Lectures in Computer Science, Benlin: Springer-Verlag, 2003, 2567: 187-198.

[25] Garber D, Kaplan S. Probabilistic solutions of equations in the braid group. http://arXiv.org/math. GR/0404076.

[26] Myasnikov A, Shpilrain V and Ushakov A. A practical attack on a braid group based cryptographic protocol. In: Proceedings of Crypto 2005, Lectures in Computer Science, Benlin: Springer-Verlag, 2005, 3621: 86-96.

[27] Tsaban B. On an authentication scheme based on the root problem in the braid group. http://eprint.iacr.org/2005/264.

[28] Myasnikov A, Shpilrain V and Ushakov A. Random subgroups of braid groups: an approach to cryptanalysis of a braid group based cryptographic protocol. In: Proceedings of PKC 2006, Lectures in Computer Science, Benlin: Springer-Verlag, 2006, 3958: 302-314.

[29] Myasnikov A and Ushakov A. Length based attack and braid groups: cryptanalysis of Anshel-Anshel-Goldfeld Key （AAGK）exchange protocol. In: Proceedings of PKC 2007, Lectures in Computer Science, Benlin: Springer-Verlag, 2007, 4450: 76-88.

WEI Yun is a Ph.D. candidate in the Institute of Electronic Technology, Information Engineering University. Her current research interest is analysis and design of cryptographic protocols. **XIONG Guo-Hua** is a doctoral supervisor in the Institute of Electronic Technology, Information Engineering University and senior engineer in the institute of Electronic Technology. His researches areas are cryptography and coding theory.