Group Signature Schemes Using Braid Groups

Tony Thomas, Arbind Kumar Lal
Department of Mathematics and Statistics
Indian Institute of Technology Kanpur, Kanpur
Uttar Pradesh, India-208016
{tony,arlal}@iitk.ac.in

Abstract

Artin's braid groups have been recently suggested as a new source for public-key cryptography. In this paper we propose the first group signature schemes based on the conjugacy problem, decomposition problem and root problem in the braid groups which are believed to be hard problems.

Key Words: braid group, braid cryptography, digital signature, group signature
2000 MSC: Primary: 94A60; Secondary: 20F36

1 Introduction

Braid groups have recently attracted the attention of many cryptographers as an alternative to number-theoretic public-key cryptography. The birthdate of braid group based cryptography can be traced back to the pioneering work of Anshel *et al.* in 1999 [2] and Ko *et al.* in 2000 [13]. Since then, braid groups have attracted the attention of many cryptographers due to the fact that, they provide a rich collection of hard problems like the *conjugacy problem, braid decomposition problem* and *root problem* and there are efficient algorithms for parameter generation and group operation [5].

Since the construction of a Diffie-Hellman type key agreement protocol and a public key encryption scheme by Ko *et al.* in 2000 [13], there have been many attempts to design other cryptographic protocols using braid groups. Positive results in this direction are a construction of pseudorandom number generator by Lee *et al.* in 2001 [15], key agreement protocols by Anshel *et al.* in 2001 [1], an implementation of braid computations by Cha *et al.* in 2001 [5], digital signature schemes by Ko *et* *al.* in 2002 [12], entity authentication schemes by Sibert *et al.* in 2002 [23] and a provably-secure identification scheme by Kim *et al.* in 2004 [11].

Digital signatures bind signers to the contents of the document they sign. Group signature schemes were introduced by Chaum and van Heyst [7] to allow individual members of a group to sign messages on behalf of a group. Formally a group signature scheme has the following properties [7]:

1. only members of the group can sign messages;
2. the receiver of the signature can verify that it is a valid signature of the group, but cannot identify the signer;
3. in case of a dispute at a later stage, the signature can be opened to reveal the identity of the signer.

The salient features of group signatures make them attractive for many specialized applications, such as voting and bidding. More generally, group signatures can be used to conceal organizational structures, e.g., when a company or a government agency issues a signed statement. Group signatures can also be integrated with an electronic cash system whereby several banks can securely distribute anonymous and untraceable e-cash.

Group signatures are generalization of *credential mechanisms* ([6]) and of *membership authentication schemes* ([17], [20]), in which a group member can convince a verifier that he belongs to a certain group without revealing his identity.

In this paper, we design some group signature schemes using braid groups. These are the first group signature schemes using braid groups.

In Section 2, we briefly review the basics of braid groups. We describe the initial system set up and some security assumptions needed for building up these signature schemes in Section 3. A group signature scheme whose security is based on the root problem is described in Section 4. In Section 5, we describe a group signature scheme that employ confirmation and denial protocols for identifying the actual signer. The security of this scheme is based on the root problem, conjugacy problem and its variants. A third group signature scheme whose security is based on the conjugacy problem and its variants is described in Section 6. The paper concludes with some general remarks in Section 7.

2 An Overview of Braid Groups

In this section, we briefly describe the basics of braid groups, hard problems in braid groups. A good introduction to braid groups is [3] and survey articles on braid cryptography are [14], [8].

2.1 Geometric Interpretation of Braids

A braid group B_n is an infinite non-commutative group arising from geometric braids composed of n-strands. A braid is obtained by laying down a number of parallel strands and intertwining them so that they run in the same direction. The number of strands is called the braid index. Braids have the following geometric interpretation: an n-braid (where $n \in \mathbb{N}$) is a set of disjoint n strands all of which are attached to two horizontal bars at the top and bottom such that each strand always heads downwards as one moves along the strand from top to bottom. Two braids are equivalent if one can be deformed to the other continuously in the set of braids.

Let B_n be the set of all n-braids. B_n has a natural group structure. Each B_n is an infinite torsion-free noncommutative group and its elements are called n-braids. The multiplication ab of two braids a and b is the braid obtained by positioning a on the top of b. The identity e is the braid consisting of n straight vertical strands and the inverse of a is the reflection of a with respect to a horizontal line.

Let $\mathbf{S}_n$ be the symmetric group on n symbols. Given a braid α, the strands define a map $p(\alpha)$ from the top set of endpoints to the bottom set of endpoints. In this way we get a homomorphism $p : B_n \to S_n$.

2.2 Presentations of Braid Groups

Any braid can be decomposed as a product of simple braids known as *Artin generators* σ_i, that have a single crossing between the i^{th} strand and the $(i+1)^{th}$ strand with the convention that the i^{th} strand crosses under the $(i+1)^{th}$ strand. The homomorphism, p maps the generator σ_i to the transposition τ_i $(= (i, i+1))$.

For each integer $n \geq\ 2$, the n-braid group B_n has the Artin presentation by generators $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ with relations

$$\begin{aligned} &\sigma_i\sigma_j = \sigma_j\sigma_i, \text{ where } \ |i-j| \geq 2, \ \text{ and} \\ &\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, \text{ for } \ 1 \leq i \leq n-2. \end{aligned} \tag{2.2.1}$$

2.3 Some Special Classes of Braids

Let B_n^+ denote the submonoid of B_n generated by $\{\sigma_1, \ldots, \sigma_{n-1}\}$. Elements of B_n^+ are called the *positive braids.* A positive braid is characterized by the fact that at each crossing the string going from left to right undercrosses the string going from right to left.

A positive braid is called *non-repeating* if any two of its strands cross at most once. We denote $D = D_n \subset B_n^+$ to be the set of all non-repeating braids. To each $\pi \in S_n$ we can associate a unique $\alpha \in D_n$ in the following way : for $i = 1, \ldots, n$ connect the upper i-th point to the lower $\pi(i)$-th point by a straight line making each *crossing positive*, *i.e.* the line between i and $\pi(i)$ is under the line between j and $\pi(j)$ if $i < j$. The following lemma says that p restricted to D_n is a bijection.

Lemma 2.1. *[9] The homomorphism $p : B_n \longrightarrow S_n$ restricted to D_n is a bijection.*

Hence non-repeating braids are also known as *permutation braids.* From this lemma it follows that $|D_n| = n!$. In this way we can identify D_n with S_n .

Let LB_n and RB_n be two subgroups of B_n consisting of braids obtained by braiding left $\lfloor \frac{n}{2} \rfloor$ strands and right $n - \lfloor \frac{n}{2} \rfloor$ strands, respectively. That is,

$$LB_n = \langle \sigma_1, \ldots, \sigma_{\lfloor \frac{n}{2} \rfloor - 1} \rangle, \text{ and } RB_n = \langle \sigma_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, \sigma_{n-1} \rangle.$$

Then we have the commutativity property that for any $\alpha \in LB_n$ and $\beta \in RB_n$, $\alpha\beta = \beta\alpha$. These subgroups of B_n are used in designing various cryptographic protocols.

2.4 Canonical Decomposition of Braids

For two words v and w in B_n, we say that $v \leq w$, if $w = avb$ for some $a, b \in B_n^+$. Then $\leq$ is a partial order in B_n [9].

The positive braid, $\Delta = (\sigma_1 \ldots \sigma_{n-1})(\sigma_1 \ldots \sigma_{n-2}) \ldots (\sigma_1\sigma_2)\sigma_1$ is called the *fundamental braid.* A braid satisfying $e \leq A \leq \Delta$ is called a *canonical factor.* There is a bijection between the set of all permutation braids and the set of all canonical factors [9]. Thus a canonical factor can be denoted by the corresponding permutation $\pi \in S_n$. By π_Δ, we mean the permutation corresponding to the fundamental braid Δ.

For a positive braid P, we say that the decomposition $P = A_0P_0$ is *left-weighted* if A_0 is a canonical factor, $P_0 \geq e$ and A_0 has the maximal word length among all such decompositions. A left-weighted decomposition $P = A_0P_0$ is unique [5]. A_0 is called the *maximal head* of P. Any braid x can be uniquely decomposed as

$$x = \Delta^u A_1 A_2 \ldots A_k, \quad \text{where} \quad u \in \mathbb{Z}, A_i \neq e, \Delta, \quad \text{is a canonical factor} \tag{2.4.1}$$

and the decomposition A_iA_{i+1} is left-weighted for each $1 \leq i \leq k-1$ [5]. This unique decomposition is called the *left canonical form* of x and so it solves the word problem. Since each canonical factor corresponds to a permutation braid, x can be denoted as

$$x = \pi_f^u \pi_1 \pi_2 \ldots \pi_k, \quad \text{where} \quad \pi_i \neq Identity, \pi_f. \tag{2.4.2}$$

Hence for implementation purposes the braid x can be represented as the tuple $(u, \pi_1, \pi_2, \ldots, \pi_k)$. The integer u, denoted by $\inf(x)$ is called the *infimum* of x and the integer $u + k$, denoted by $\sup(x)$ is called the *supremum* of x. The *canonical length* of x, denoted by len(x), is given by $k = \sup(x) - \inf(x)$.

2.5 Hard Problems in Braid Groups

We use the following hard problems in our signature schemes.

1. **Conjugacy Search Problem (CSP)**

   Let $(x, y) \in B_n \times B_n$, such that $y = a^{-1}xa$, where $a \in B_n$ or some subgroup of B_n. The *conjugacy search problem* is to find a b such that $y = b^{-1}xb$.

2. **Multiple Simultaneous Conjugacy Search Problem (MSCSP)**

   Let $(x_1, a^{-1}x_1a), \ldots, (x_r, a^{-1}x_ra) \in B_n \times B_n$ for some $a \in B_n$ or some subgroup of B_n. The *multiple simultaneous conjugacy problem* is to find a b such that, $b^{-1}x_1b = a^{-1}x_1a, \ \ldots, \ b^{-1}x_rb = a^{-1}x_ra.$

3. **Braid Decomposition Problem (BDP)**

   Let $(x, y) \in B_n \times B_n$, where $y = a_1xa_2$ for some $(a_1, a_2) \in LB_n \times LB_n$. The *braid decomposition problem* is to find a pair $(b_1, b_2) \in LB_n \times LB_n$ such that $y = b_1xb_2$.

4. **Multiple Simultaneous Braid Decomposition Problem (MSBDP)**

   Let $(x_1, a_1x_1a_2), \ldots, (x_r, a_1x_ra_2) \in B_n \times B_n$ for some $(a_1, a_2) \in LB_n \times LB_n$. The *multiple simultaneous braid decomposition problem* is to find a pair $(b_1, b_2) \in LB_n \times LB_n$ such that, $b_1x_1b_2 = a_1x_1a_2, \ \ldots, \ b_1x_rb_2 = a_1x_ra_2$.

5. **Root Extraction Problem (RP)**

   Let $x = a^p$, where $a, x \in B_n$ and $p \in \mathbb{N}$. Then the *root problem* (for the exponent p) is to find a braid $b \in B_n$ such that $b^p = x$.

3 Preliminaries

In this section, the initial system set up, intractability assumptions, some other assumptions and some notation used in this paper are given.

3.1 Initial Setup

The system parameters n and l are chosen to be sufficiently large positive integers and are made public. Since the braid group B_n is discrete and infinite, we cannot have a uniform probability distribution on B_n. But there are finitely many positive n-braids with l canonical factors, we may consider randomness for these braids. Such a braid can be generated by concatenating l random canonical factors. Let,

$$\begin{aligned} B_n(l) &= \{b \in B_n \mid 0 \leq inf(b) \leq sup(b) \leq l\}, \\ LB_n(l) &= \{b \in LB_n \mid 0 \leq \inf(b) \leq \sup(b) \leq l\} \text{ and} \end{aligned}$$

$$RB_n(l) \quad = \quad \{b \in RB_n \mid 0 \leq \inf(b) \leq \sup(b) \leq l\}.$$

Then $|B_n(l)| \leq l(n!)^l$ and so $LB_n(l), RB_n(l)$ and $B_n(l)$ are finite sets. We use the random braid generator given in [5] (which produces random braids in $O(ln)$ time) for generating random braids. Also, we consider uniform probability distribution on these sets.

Let $H : \{0,1\}^* \rightarrow B_n(l)$ be a collision free hash function. H can be constructed by composing a usual hash function of bit strings with a conversion from bit strings of fixed length to elements of $B_n(l)$. A way to construct this conversion function, $c : \{0,1\}^k \rightarrow B_n(l)$ is given in [12].

3.2 Notations

We use the following notations through out this paper.

- By $a \in_r A$, we mean a random choice of an element a from the set A.
- By $P \xrightarrow{Q} V$, we mean P sends Q to V.

3.3 Group Manager

Let T be a group manager, who chooses the private key of the group and creates the public key of the group. T also manages the members of the group. T is needed in identifying the actual signer in our first and third signature schemes. T is not needed in our second signature scheme.

3.4 Intractability Assumptions

We assume that the hard problems **CSP, MSCSP, BDP, MSBDP, RP**, stated in Section 2.5 are intractable in braid groups. However, we assume that the *conjugacy decision problem* given below is easy in braid groups.

Let $(x, y) \in B_n \times B_n$. The *conjugacy decision problem* is to decide whether x and y are conjugates or not, that is to decide whether there exists an $a \in B_n$ such that $y = a^{-1}xa$ or not. The conjugacy decision problem may be solved using the algorithm given in [12].

3.5 Some New Assumptions

In this paper, we make two assumptions. The first assumption is similar to the **EDL** intractability assumption used in [16]. The **EDL** (Equality of Discrete Logarithms) intractability assumption can be stated as follows : given $x, y \in_r G = \langle f \rangle = \langle g \rangle$, it is computationally infeasible to determine the equality of $\log_f x$ and $\log_g y$ over $\mathbf{Z}_n$, where $ord(g) = n$. So we have our first assumption as

Assumption 3.1. *For $(\alpha, \beta) \in B_n \times B_n$, let*

$$F_\beta(\alpha) = \{(a, b) \in B_n \times B_n : \alpha = a\beta b\}.$$

Then, given two pairs of braids (α, β) and (γ, δ) in $B_n \times B_n$, it is computationally infeasible to check whether $F_\beta(\alpha) \cap F_\delta(\gamma) \neq \emptyset$ or not.

The second assumption is about cardinalities of certain sets, which may be stated as follows.

Assumption 3.2. *Let n, l be sufficiently large positive integers, $\alpha, \beta, \gamma \in_r B_n(l)$, $a_1, a_2 \in_r LB_n(l)$ and $a \in_r RB_n(l)$. Then the cardinality of the set*

$$E_a(\beta, \gamma) = \{b \in RB_n(l) : \ b^{-1}\alpha b = a^{-1}\alpha a, \ b^{-1}\beta b \neq a^{-1}\beta a\}$$

is bounded below by a non decreasing function $p(n, l)$ of n and l.

In this paper, we do not undertake any theoretical or numerical study to check the validity of the above assumptions.

4 Group Signature Scheme 1

In [7] Chaum *et al.* describe a group signature scheme using public-key systems. In this case the group manager T chooses a public key system, gives each person a list of secret keys (these lists are all disjunct) and publishes the complete list of corresponding public keys (in random order) in a Trusted Public Directory. Each person can sign a message with a secret key from his list, and the recipient can verify this signature with the corresponding public key from the public list. Each key will be used only once, otherwise the signature created with that key gets linked. T

knows all the list of secret keys, so that in case of a dispute, he can identify the signer. Hence T is needed for the setup and for opening of the signature.

We can adopt this group signature scheme directly to the braid group frame work as follows : T chooses a set E of braids and raises them to the p^{th} power, where p is an integer greater than 1. Each person is given a list of braids from E (these lists are all disjunct) and the complete list of p^{th} powers of elements of E (in random order) is published in a Trusted Public Directory. To sign a message m, a group member chooses a braid α from his list and forms the signature $S_m = \alpha H(m)$. The recipient can verify this signature by computing $(S_m H(m)^{-1})^p$ and checking it with the corresponding public key in the Trusted Public Directory. Each key will be used only once. T knows all the list of secret keys, so that in case of a dispute, he can identify the signer.

A problem with this scheme is that the group manager knows all the secret keys of the group members and can therefore also create signatures. This problem can be overcome by making each user to untraceably send one (or more) public keys to a public list, which will be the public key of the group. But it has to be ensured that only the group members will be able to send public keys to that list.

Although, the scheme is very elegant it has the obvious disadvantage that a key can be used only once. However, we can trivially see that the security of this scheme is equivalent to solving the root problem. Hence this group signature scheme is highly secure. This is the only cryptographic scheme on braid groups whose security depends solely on the root problem (RP).

5 Group Signature Scheme 2

In this section, we describe a group signature scheme which does not involve a group manager. The security of the scheme is based on the hardness of **BDP, MSCSP** and **RP**. Here the recipient of the signature can easily check whether the signature has come from a particular group or not. But the identity of the signer can not be verified unless the verifier engages in an interactive protocol with the signer as in the case of undeniable signatures.

5.1 Key Generation

Let G be a group with k members $P_1, P_2, \ldots, P_k$. The members of the group agree on a secret braid $\alpha \in B_n(l)$. $\beta = \alpha^4$ is published as the public key of the group. Also, each member P_i of the group chooses $(u_i, v_i) \in LB_n(l) \times LB_n(l)$ as his secret key. In this case, the public key of P_i is $x_i = u_i^{-1}\beta v_i$.

We shall denote by PK the tuples $(\beta, \{x_i\}_1^k)$ generated as above.

5.2 Signature Generation

Let m be the message to be signed. Suppose P_i wants to sign m. He computes the signature $S_m = u_i^{-1}y^{-1}\alpha^2 y u_i$, where $y = H(m)$.

We shall denote by $SIG(m)$, the set of valid signatures on m.

5.3 Confirming the Group Identity of the Signature

Given an alleged signature $\hat{S}_m$, suppose that a verifier V wants to check whether it is a valid signature from the group G. V computes $\hat{S}_m^2$ and checks whether it is conjugate to β using the algorithm described in [12].

Note that $\hat{S}_m^2 = u_i^{-1}y^{-1}\beta y u_i$. Hence if $\hat{S}_m$ is a valid signature of a member of G, then S_m^2 is conjugate to β.

5.4 Confirmation Protocol

Suppose that a signer P_i claims that a signature $\hat{S}_m$ was made by him. Then a verifier V first checks the group identity of the signature using the above protocol and then verifies the claim of P_i by engaging in an interactive confirmation protocol with him. Let us denote the prover P_i by P. When $\hat{S}_m$ is a valid signature of m by P, he will be able to convince V of this fact, while if the signature is invalid then no prover even if he is computationally unbounded will be able to convince V to the contrary except with a negligible probability.

Signature Confirmation Protocol

Input : Prover: Secret keys (α, u_i, v_i).

Verifier: Public key $(\beta, \{x_j\}_{j=1}^{k})$ and alleged $\hat{S}_m$.

1. V chooses $a \in_r RB_n(l)$, computes $Q = a^{-1}(\hat{S}_m)^2 x_i a$ and $V \xrightarrow{Q} P$.
2. P chooses $b, c \in_r B_n(l)$, computes $R = bu_iQv_i^{-1}c$ and $P \xrightarrow{R} V$.
3. $V \xrightarrow{a} P$.
4. P Checks the value of Q and then $P \xrightarrow{(b,c)} V$.
5. V verifies that $R = ba^{-1}y^{-1}\beta y\beta ac$.

If equality holds then V accepts $\hat{S}_m$ as the signature on m, otherwise "undetermined".

Theorem 5.1. Confirmation Theorem. *Let $(\beta, \{x_i\}_1^k) \in PK$.*

Completeness*: Given $S_m \in SIG(m)$, if P follows the signature confirmation protocol then V always accepts S_m as a valid signature.*

Soundness*: A Cheating prover P^* even computationally unbounded cannot convince V to accept $\hat{S}_m \notin SIG(m)$ with probability greater than $\frac{1}{p(n,l)}$.*

Proof. **Completeness**: Let S_m be a valid signature. P computes

$$R = b(u_iQv_i^{-1})c = b(u_ia^{-1}(\hat{S}_m)^2x_iav_i^{-1})c = ba^{-1}(y^{-1}\beta y\beta)ac.$$

which V verifies after getting (b, c) from P and accepts the signature as valid. Hence the protocol is complete.

Soundness: The idea is that there are many values of a which give the same value for the challenge Q and different values for the response R and a cheating prover P^* has no way to distinguish between these different values of a, even if he has infinite computational power. That is, from Assumption 3.2, there are at least $p(n, l)$ choices, for $a \in RB_n(l)$ which give the same value of Q but giving different values of R. Hence it is infeasible for a cheating prover P^* to distinguish between these different values of a, even if he has infinite computational power. Therefore a cheating prover P^*, even computationally unbounded, cannot convince V to accept $\hat{S}_m \notin SIG(m)$ with probability greater than $\frac{1}{p(n,l)}$. Thus the protocol is sound. $\square$

Remark 5.1. *A closer examination of the protocol reveals that it has the zero-knowledgeness property also (see[26]).*

5.5 Disavowal Protocol

If P_i wants to prove to V that $\hat{S}_m$ is not his signature on m, he engages in a disavowal protocol with V. As in the case of confirmation protocol, we denote P_i by P. In the case that $\hat{S}_m$ is not a valid signature, P will be able to convince V of this fact, while if $\hat{S}_m$ is a valid signature of P on m, even if he is computationally unbounded he will not be able to convince V that the signature is invalid except with negligible probability.

Disavowal Protocol

Input : Prover : Secret keys (α, u_i, v_i).
Verifier : Public key $(\beta, \{x_j\}_1^k) \in PK, y$ and alleged $\hat{S}_m$.

1. V chooses $a, b \in_r RB_n(l)$ such that a and b commute and computes $Q_1 = a^{-1}(\hat{S}_m)^2 x_i a$, $Q_2 = b^{-1}(\hat{S}_m)^2 x_i b$ and $V \overset{(Q_1,Q_2)}{\longrightarrow} P$.

2. P computes the response $R_1 = u_i Q_1 v_i^{-1}$, $R_2 = u_i Q_2 v_i^{-1}$ and $P \overset{(R_1,R_2)}{\longrightarrow} V$.

3. V verifies that $b^{-1}(R_1\beta^{-1})b = a^{-1}(R_2\beta^{-1})a$.

If equality holds V accepts $\hat{S}_m$ as an invalid signature. Otherwise P is answering improperly.

Theorem 5.2. Denial Theorem *Let* $(\beta, \{x_i\}_1^k) \in PK$.
Completeness*: Suppose that $\hat{S}_m \notin SIG(m)$. If P and V follow the protocol, then V always accepts that $\hat{S}_m$ is not a valid signature of m.*
Soundness*: Suppose that $\hat{S}_m \in SIG(m)$. Then a cheating prover, even computationally unbounded, cannot convince V to reject the signature with probability greater than $\frac{1}{p(n,l)}$.*

Proof. **Completeness**: Assume that $\hat{S}_m \notin SIG(m)$. We have,

$$R_1 = u_i a^{-1}(\hat{S}_m)^2 x_i a v_i^{-1} = a^{-1}u_i(\hat{S}_m)^2 u_i^{-1}\beta a \quad \text{and}$$
$$R_2 = u_i b^{-1}(\hat{S}_m)^2 x_i b v_i^{-1} = b^{-1}u_i(\hat{S}_m)^2 u_i^{-1}\beta b.$$

Therefore,

$$b^{-1}R_1 b = a^{-1}R_2 a = a^{-1}b^{-1}(u_i(\hat{S}_m)^2 u_i^{-1}\beta)ba.$$

Hence the protocol is complete.

Soundness: Assume that $\hat{S}_m \in SIG(m)$. Let R_1 and R_2 be the responses given by P^* in the protocol. Let if possible, $b^{-1}R_1 b = a^{-1}R_2 a$. Then

$$R_2 = a(b^{-1}R_1 b)a^{-1} = a\beta a^{-1}, \text{ where } \gamma = b^{-1}R_1 b.$$

In the worst case, we may regard γ as a known constant for P when he tries to determine R_2. But then the ability to determine R_2 amounts to the establishment of an invalid signature, which contradicts Theorem 5.1 (soundness of the confirmation protocol). Hence the protocol is sound. □

Remark 5.2. *For the ease of analysis, the disavowal protocol was given in a non zero-knowledge fashion. However, zero-knowledge versions of the disavowal protocol can also be constructed in a similar manner (see [26]).*

6 Group Signature Scheme 3

In this section, we describe another group signature scheme. This scheme is given in the usual frame work of group signature schemes as described in [19]. The security of the scheme is based on the hardnes of **CSP, MSCSP** and **MSBDP**. Here the recipient of the signature can easily verify the group identity of the signature. However, if a dispute occurs the group manager can open the signature and identify the signer.

6.1 Setup

The group manager T chooses a secret braid $s \in_r LB_n(l)$, $k_1, k_2 \in_r RB_n(l)$, and $\alpha \in_r B_n(l)$ and publishes $x = s^{-1}\alpha s$ as the public key of the group.

6.2 Join

Suppose now that a user P wants to join the group. We assume that the communication between a group member and T is secure, that is private and authentic.

The following protocol is performed between the user P and the Trusted Authority T.

1. $T \xrightarrow{(s,\alpha)} P$.

2. P chooses $u \in_r B_n(l)$, $a \in_r LB_n(l)$ computes $v = u^{-1}\alpha u$, $w = a^{-1}ua$ and $P \xrightarrow{(v,w)} T$.

3. T computes $z_1 = k_1^{-1}wk_1$, $z_2 = k_2^{-1}wk_2$ and $T \xrightarrow{(z_1,z_2)} P$.

4. P computes $\beta_1 = az_1a^{-1}$ and $\beta_2 = az_2a^{-1}$.

Consequently, at the end of the protocol, T creates a new entry in the group database with v as the public key of the member P.

6.3 Sign

Let m be the message which has to be signed. Suppose that the group member P wants to sign m. He computes $S_1 = s^{-1}ys$ and $S_2 = s^{-1}\beta_1^{-1}y\beta_2 s$, where $y = H(m)$. Signature is the pair $S_m = (S_1, S_2)$.

6.4 Verify

A recipient of the signature after getting S_m, checks whether S_1 is conjugate to y to check whether S_m is a valid signature of y or not.

To check the group identity of the signature, V checks whether S_1x is conjugate to $y\alpha$. If it holds, V accepts S_m as a signature from the group G.

6.5 Open

In case of a dispute, the group manager can identify the signer of the signature $\hat{S}_m = (\hat{S}_1, \hat{S}_2)$ in the following way. He first computes $\hat{S}_3 = k_1s\hat{S}_2s^{-1}k_2^{-1}$. Now he can find out whether P is the signer by checking whether $\hat{S}_3v$ is conjugate to $k_1yk_2^{-1}\alpha$ or not. If it holds, the signature was made by P.

6.6 Security Analysis

In this section we will show that this group signature scheme satisfies some of the properties of for an ideal group signature.

1. **Unforgeability**: Since to sign on behalf of the group, one should know the secret key s, only group members can sign on behalf of the group. However, an attacker gets several pairs of braids and its conjugates by s. Hence under the assumption that multiple simultaneous conjugacy decomposition problem (MSCDP) is hard in braid groups an attacker cannot get s and the signature scheme stands unforgeable.

   Remark 6.1. *We may make our scheme more secure by avoiding an attack on MSCDP in the following way : the group manager chooses $s_1, s_2 \in_r LB_n(l)$ instead of $s \in_r LB_n(l)$. He makes the group public key as $s_2^{-1}\alpha s_1$. Now, given a message m the signer computes the signature as $S_m = (S_1 = s_1^{-1}ys_2$, $S_2 = s_1\beta_1^{-1}y\beta_2 s_2)$. The protocols for verification and opening the signature can be rewritten in a similar way.*

2. **Unlinkability**: Let m_1 and m_2 be two messages signed by the group members. Let $y_1 = H(m_1)$ and $y_2 = H(m_2)$. Let $S_{m_1} = (S_1^1, S_2^1)$ and $S_{m_2} = (S_1^2, S_2^2)$. Now, the problem of linking S_{m_1} and S_{m_2} reduces to deciding whether S_2^1 and S_2^2 are linked or not. Now, $S_2^1 = s^{-1}\beta_1^{-1}y_1\beta_2 s$ and $S_2^2 = s^{-1}\beta_1^{-1}y_2\beta_2 s$. Hence deciding whether S_2^1 and S_2^2 are linked or not reduces to checking whether the pairs (S_2^1, y_1) and (S_2^2, y_2) have the same factors or not. Now, this is infeasible by Assumption 3.1. Hence the signature scheme is unlinkable.

3. **Anonymity**: Given a group signature, to identify the actual signer is computationally hard to do for everyone but the group manager. Consider a signature on m by P. Let $S_m = (S_1, S_2)$. Now $S_2 = s^{-1}(k_1^{-1}u^{-1}k_1)y(k_2^{-1}uk_2)s$ and in order to show that the signature belongs to P, a group member has to prove that $(k_1 s\hat{S}_2 s^{-1}k_2^{-1})v$ is conjugate to $k_1 y k_2^{-1}\alpha$. There is no apparent way of proving the identity of the signer other than by getting the private keys of the signer or that of the Trusted Authority. But any group member can compute

$sS_2s^{-1} = (k_1^{-1}u^{-1}k_1)y(k_2^{-1}uk_2)$. Now, the only way for a group member $\hat{P}$ with secret key $\hat{v}$ to find out the identity of the signer is to get the value of k_1 and k_2 from $k_1^{-1}\hat{v}k_1$ and $k_2^{-1}\hat{v}k_2$ which he obtained from the group manger. But this amounts to solving a conjugacy search problem and by assumption the conjugacy search problem is hard. Hence the signature scheme is anonymous.

4. **Exculpability** : The group manager does not get any information about a group member's secret key u as well as signing keys $k_1^{-1}uk_1$ and $k_2^{-1}uk_2$. The values of u as well $k_1^{-1}uk_1$ and $k_2^{-1}uk_2$ are computationally hidden from the group manager because of the protocols involved in the Join session of the member P. Hence the group manager cannot sign on behalf of a group member. Similarly, any group member cannot sign on behalf of any other member. Hence exculpability holds.

5. **Traceability**: Assume that the signature $S_m = (S_1, S_2)$ on the message m was made by P. Now the group manager can compute

$$\begin{aligned} S_3v &= (k_1sS_2s^{-1}k_2^{-1})v = (k_1s(s^{-1}\beta_1^{-1}y\beta_2s)s^{-1}k_2^{-1})(u^{-1}\alpha u) \\ &= (k_1\beta_1^{-1}y\beta_2k_2^{-1})(u^{-1}\alpha u) = (k_1(k_1^{-1}u^{-1}k_1)y(k_2^{-1}uk_2)k_2^{-1})(u^{-1}\alpha u) \\ &= u^{-1}k_1yk_2^{-1}\alpha u. \end{aligned}$$

Hence S_3v is conjugate to $k_1yk_2^{-1}\alpha$. Thus, the group manager can open any valid group signature and identify the actual signer. Hence the signature is traceable.

7 Concluding Remarks

In this paper, we constructed three group signature schemes based on some hard problems in braid groups. Our schemes are the first in this direction using braid groups. It is open to use other hard problems in braid groups for designing more group signature schemes and other cryptographic protocols.

The first signature scheme has the property that its security is entirely depending on the root problem. This is the only cryptographic scheme on braid groups whose security is solely depending on the root problem. Root problem is believed to be harder than the conjugacy and decomposition problems. Hence we may believe that this scheme is the most secure one. The second scheme combines the notion of undeniable signatures with group signatures. Our third scheme is set in the usual frame work of group signatures.

The problem of checking the equality of factors in a 3-factor decomposition of two given braids with the middle factors known is employed in Assumption 3.1. We leave this assumption as well as Assumption 3.2 for further investigation. The first step in the investigation of the second assumption may be to the estimate of number of conjugates of a random element which are equal. Numerical experiments might throw some light on these assumptions.

The birth of braid cryptography has simulated the search for other exotic mathematical structures for doing public-key cryptography. People have started looking at other nonabelian groups [25], [24], [18], [10] and combinatorial groups [22], [21] for building public-key cryptosystems. Although, we have described our schemes in the frame work of braid groups, these protocols can be carried over to many other nonabelian groups with slight modifications. Further, one can modify these protocols to other variations of group signatures like, the ring signatures and undeniable group signatures discussed in Section 1. Hence, we hope that this study will motivate further research on digital signatures based on nonabelian groups and combinatorial groups.

References

[1] I. Anshel, M. Anshel, B. Fisher, D. Goldfeld, *New key agreement protocols in braid group cryptography*, Progress in Cryptology- CT-RSA 2001, Lecture Notes in Computer Science, Springer-Verlag, 2020 (2001), pp. 13-27.

[2] I. Anshel, M. Anshel, D. Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett., Vol.6, (1999), pp. 287-291.

[3] J. S. Birman, *Braids, links and mapping class groups*, Annals of Math. Study 82, Princeton University Press, 1974.

[4] J. Camenisch, *Efficient and generalized group signatures*, Advances in Cryptology: Proceedings of EUROCRYPT 1997, Lecture Notes in Computer Science, Springer-Verlag, 1233 (1997), pp. 465-479.

[5] J. C. Cha, K. H. Ko, S.J. Lee, J. W. Han, J. H. Cheon, *An efficient implementation of braid groups*, Advances in Cryptology: Proceedings of ASIACRYPT 2001, Lecture Notes in Computer Science, Springer-Verlag, 2248 (2001), pp. 144-156.

[6] D. Chaum, *Showing credentials without identification*, Advances in Cryptology: Proceedings of EUROCRYPT 85, Lecture Notes in Computer Science, Springer-Verlag, 219 (1986), pp. 241-244.

[7] D. Chaum, E. van Heijst, *Group signatures*, Advances in Cryptology: Proceedings of EUROCRYPT 91, Lecture Notes in Computer Science, Springer-Verlag, 547 (1991), pp. 241-246.

[8] P. Dehornoy, *Braid-based cryptography*, Contemp. Math., 360 (2004), pp. 5-33.

[9] D. Epstein, J. Cannon, D. Holt, S. Levy, M. Paterson, W. Thurston, *Word Processing in Groups*, Jones Bartlett, 1992.

[10] D. Grigoriev, I. Ponomarenko, *Constructions in public-key cryptography over matrix groups*, Available at: http://arxiv.org/abs/math/0506180.

[11] Z. Kim, K. Kim, *Provably-secure identification scheme based on braid groups*, SCIS 2004, The 2004 Symposium on Cryptography and Information Security, Sendai, Japan, Jan. 27-30, 2004.

[12] K. H. Ko, D. H. Choi, M. S. Cho, J. W. Lee, *New signature scheme using conjugacy problem*, Available at: http://eprint.iacr.org/2002/168.pdf.

[13] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. S. Kang, C. S. Park, *New public-key cryptosystem using braid groups*, Advances in Cryptology: Proceedings of CRYPTO 2000, Lecture Notes in Computer Science, Springer-Verlag, 1880 (2000), pp. 166-183.

[14] E. Lee, *Braid groups in cryptology*, IEICE Tarns. Fundamentals, Vol.E87-A, No.5 (2004), pp. 986-992.

[15] E. K. Lee, S. J. Lee, S. G. Hahn, *Pseudorandomness from braid groups*, Advances in Cryptology: Proceedings of CRYPTO 2001, Lecture Notes in Computer Science, Springer-Verlag, 2139 (2001), pp. 486-502.

[16] Yuh-Dauh Lyuu, Ming-Luen Wu, *Group undeniable signatures*, International Journal of Computer Research, 12, No.2 (2003), pp. 301-309.

[17] K. Ohta, T. Okamoto, K. Koyama, *Membership authentication for hierarchical multigroup using the extended Fiat-Shamir scheme*, Advances in Cryptology: Proceedings of EUROCRYPT 90, Lecture Notes in Computer Science, Springer-Verlag, 473 (1991), pp. 446-457.

[18] Seong-Hun Paeng, Kil-Chan Ha, J. H. Kim, S. Chee, C. Park, *New public key cryptosystem using finite non abelian groups*, Advances in Cryptology: Proceedings of CRYPTO 2001, Lecture Notes in Computer Science, Springer-Verlag, 2139 (2001), pp. 470-485.

[19] C. Popescu, *An efficient group signature scheme for large groups*, Studies in Informatics and Control Journal, Vol. 10, No.1 (2001), pp. 7-14.

[20] H. Shizuya, K. Koyama, T. Itoh, *Demonstrating possession without revealing factors and its applications*, Advances in Cryptology: Proceedings of AUSCRYPT 90, Lecture Notes in Computer Science, Springer-Verlag, 453 (1990), pp. 273-293.

[21] V. Shpilrain, A. Ushakov, *Thompson's group and public key cryptography*, Available at : http://arxiv.org/abs/math.GR/0505487.

[22] V. Shpilrain, G. Zapata, *Combinatorial group theory and public key cryptography*, Available at : http://eprint.iacr.org/2004/242.

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

[24] R. Steinwandt, *Non-abelian groups in public key cryptography*, Abstract available at : http://www.cms.math.ca/Events/winter04/abs/Plen.html.

[25] E. Stickel, *A new public-key cryptosystem in non abelian groups*, Proceedings of the Thirteenth International Conference on Information Systems Development. Vilnius Technika, Vilnius 2004, S. 70-80.

[26] T. Thomas, *On public-key cryptography using hard problems in braid groups*, Ph.D Thesis, I.I.T Kanpur, Kanpur, India, September 2005.