A Thesis for the Degree of Master of Science

Provably-Secure Identification Schemes based on Conjugacy and DDH Problems

Zeen Kim

School of Engineering

Information and Communications University

2004

Provably-Secure Identification Schemes based on Conjugacy and **DDH** Problems

Provably-Secure Identification Schemes based on Conjugacy and DDH Problems

Advisor : Professor Kwangjo Kim

by

Zeen Kim
School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejeon, Korea
Jan. 3. 2004
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

Provably-Secure Identification Schemes based on Conjugacy and DDH Problems

Zeen Kim

We certify that this work has passed the scholastic standards required by Information and Communications University as a thesis for the degree of Master of Science

Jan. 3. 2004

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae Choon Cha, Assistant Professor
School of Engineering

Committee Member
Dae Sung Kwon, Ph.D
NSRI

M.S. Zeen Kim
2001507

Provably-Secure Identification Schemes based on Conjugacy and DDH Problems

School of Engineering, 2004, 51p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

There are many situations where it is necessary to “prove” one’s identity. Typical scenarios are to login to a computer, to get access to an account for electronic banking or to withdraw money from an automatic teller machine. Older methods use passwords or PIN’s to implement user identification. Though successfully used in certain environments, these methods also have weakness. For example, anyone to whom you must give your password to be verified has the ability to use this password and impersonate you. Zero-knowledge (and other) identification schemes provide a new type of user identification. It is possible for you to authenticate yourself without giving to the authenticator any knowledge to impersonate you.

This thesis deals with a technique, called an *identification scheme* or entity authentication scheme, which allows one party to gain assurances that the identity of another is as declared, thereby preventing impersonation. Our proposed identification scheme is based on braid groups. Most of cryptosystems are based on commutative groups, but new cryptosystems based on non-commutative groups have been proposed. (*Braid cryptosystem* is one of them.) These systems are very difficult to analyze for their non-commutative properties. In the recent years, beginning with [44], several authors proposed to build secure cryptographical schemes using noncommutative groups, in particular Artin's braid groups [1, 29, 30, 34], a natural idea as, on the one hand, braid groups are more complicated than Abelian groups, but, on the other hand, they are not too complicated to be worked with. In particular, the conjugacy problem in braid groups is algorithmically difficult, and it consequently provides one-way functions.

In this thesis we construct two new interactive identification schemes. One is based on the conjugacy problem and another is based on decision Diffie-Hellman (DDH) assumption. The first scheme is the primary one based on conjugacy problem over a braid group. We prove that the scheme based on conjugacy problem is secure against active attacks if the k-simultaneous conjugator search problem (k-SCSP) is intractable. Our proof is based on the fact that the conjugacy search problem (CSP) is hard in braid group, on the other hand, the conjugacy decision problem (CDP) is easy in braid group by Ko *et al.*'s algorithm. Second scheme has some limitation for adversary. That is an adversary can view only k-times interactions. Under the DDH assumption and simulator's limitation, we prove the scheme is secure against impersonation attack. *i.e,* the impersonator has negligible advantage.

Contents

List of Tables

List of Figures

List of Abbreviations

CP Conjugacy Problem

CSP Conjugacy Search Problem

CDP Conjugacy Decision Problem

DDH Decision Diffie-Hellman

FS Fiat-Shamir

FFS Feige-Fiat-Shamir

GQ Guillou-Quisquator

ID Identity

KK Kim-Kim

MCSP Matching Conjugacy Search Problem

MTSP Matching Triple Search Problem

PPT Probabilistic Polynomial Time

SCSP Simultaneous Conjugacy Search Problem

ZKIP Zero-Knowledge Interactive Proof

ZKP Zero-Knowledge Proof

List of Notations

B_n a braid group

$\mathcal{G}$ key generation algorithm which is modeled as PPT

k a security parameter

$\mathcal{I}$ an adversary $(\tilde{\mathcal{P}}, \tilde{\mathcal{V}})$

$\mathcal{P}$ a prover, which is a PPT algorithm

$\tilde{\mathcal{P}}$ a dishonest prover, which is a PPT algorithm

$\bar{\mathcal{P}}$ an honest prover, which is a PPT algorithm

Pub a collection of public parameters of given identification scheme

Sec a collection of secret parameters of given identification scheme

$\mathcal{V}$ a verifier, which is a PPT algorithm

$\tilde{\mathcal{V}}$ a dishonest verifier, which is a PPT algorithm

$\bar{\mathcal{V}}$ an honest verifier, which is a PPT algorithm

$\mathcal{V}^*$ the algorithm of a general (possibly dishonest) verifier, which is a PPT algorithm

$x \leftarrow_R \mathcal{S}$ an element x randomly selected according to a probability space $\mathcal{S}$

$\Pr_{\mathcal{S}}[x]$ probability that $\mathcal{S}$ associates with the element x

$\in_{\mathcal{R}}$ chosen at random

Chapter 1
Introduction

Modern cryptography is concerned with algorithms and schemes which ensure confidentiality, integrity and proof of origin for digital communications. In conventional cryptosystems, these various functionalities are provided in a setting where the transmitter and the receiver share a common key, whose secrecy is requested for proper operation. A major breakthrough took place in 1976 with the appearance of public-key cryptography. In this seminar [16], Diffie and Hellman proposed a new concept, allowing the use of two matching keys, one for encryption and a different one for decryption. The main novel character of the concept is that the encryption key need not be kept secret. Shortly afterwards, Rivest, Shamir and Adleman [41] invented the celebrated RSA algorithm. This algorithm is a public key system making heavy use of operations modulo a large integer n obtained by multiplying together two prime numbers and whose security is related to difficulty of factoring n. Since then, nearly all new cryptographic schemes have been based on the mathematical hard problems, despite the fact that this produces a significant computing load. Even if the question of finding appropriate alternative techniques is considered a major open problem in the area of public key cryptography, little progress has been made.

Subsequent research in the area has been aimed at achieving simpler functionalities at a lower cost in terms of computing load. This research has been quite successful in the setting of identification, where a user attempts to convince another entity of his identity by means of an on-line communication.

Of course, the transaction should not give enough information to allow anyone else to misrepresent himself as the legitimate user, including the entity carrying the identification process. A major step forward in this area was made with zero-knowledge proofs, introduced in 1985, in a paper [24] by Goldwasser, Milcali and Rackoff and whose practical significance for public key identification was soon demonstrated in the work of Fiat and Shamir [19]. Still, zero-knowledge based techniques have continued to rely on number theory, even though the new protocols do not exactly follow the basic public key paradigm invented by Diffie and Hellman and requiring trap-door functions. Rather, they are based on one-way functions, which is a less stringent requirement and which opens the way to use simpler techniques, more combinatorial in spirit. After then, there were many attempts to build identification schemes.

1.1 Identification and Its Objectives

There are many situations where it is necessary to "prove" one's identity. Typical scenarios are to login to a computer, to get access to an account for electronic banking or to withdraw money from an automatic teller machine. Older methods use passwords or PIN's to implement user identification. Though successfully used in certain environments, these methods also have weakness. For example, anyone to whom you must give your password to be verified has the ability to use this password and impersonate you. Zero-knowledge (and other) identification schemes provide a new type of user identification. It is possible for you to authenticate yourself without giving to the authenticator the any useful knowledge to impersonate you.

The identification scheme is an interactive protocol where a prover, $\mathcal{P}$, tries to convince a verifier, $\mathcal{V}$, of his identity. Only $\mathcal{P}$ knows the secret value corresponding to his public one, and the secret value allows to convince $\mathcal{V}$ of his identity. If we replace "identity" by "authenticity" of messages, identification schemes are nearly equivalent to *signature schemes*.

From the point of view of the verifier, the outcome of an identification protocol is either *acceptance* of the prover's identity as authentic, or *rejection*. More specifically, the objectives of an identification protocol include the following [35]:

1. In the case of honest parties $\mathcal{P}$ and $\mathcal{V}$, $\mathcal{P}$ is able to successfully authenticate himself to $\mathcal{V}$, *i.e.*, $\mathcal{V}$ will complete the protocol having accepted $\mathcal{P}$'s identity.

2. (*Transferability*) $\mathcal{V}$ cannot reuse an identification exchange with $\mathcal{P}$ so as to successfully impersonate $\mathcal{P}$ to a third party $\mathcal{A}$.

3. (*Impersonation*) The probability is *negligible* that any party $\mathcal{A}$ distinct from $\mathcal{P}$, carrying out the protocol and playing the role of $\mathcal{P}$, can cause $\mathcal{V}$ to complete and accept $\mathcal{P}$'s identity.

4. All the previous objectives hold even if: a polynomially large number of previous authentication between $\mathcal{P}$ and $\mathcal{V}$ have been observed; the adversary $\mathcal{A}$ has participated in previous protocol executions with either or both $\mathcal{P}$ and $\mathcal{V}$; and multiple instances of the protocol, possibly initiated by $\mathcal{A}$, may be run simultaneously.

The precise definition of goals for an identification protocol is given with respect to provable security against the attacks in later chapter. Informally speaking, the objectives derive the idea of zero-knowledge-based protocols whose executions do not reveal any partial information which makes $\mathcal{A}$'s task any easier whatsoever.

1.2 Our Contributions

This thesis deals with a technique, called an *identification scheme* or entity authentication scheme, which allows one party to gain assurances that the identity of another is as declared, thereby preventing impersonation. Our proposed identification scheme is based on braid groups. Most of cryptosystems are based on commutative groups, but new cryptosystems based on non-commutative groups have been proposed. (*Braid cryptosystem* is one of them.) These systems are very difficult to analyze for their non-commutative properties. In the recent years, beginning with [44], several authors proposed to build secure cryptographical schemes using noncommutative groups, in particular Artin's braid groups [1, 29, 30, 34], a natural idea as, on the one hand, braid groups are more complicated than Abelian groups, but, on the other hand, they are not too complicated to be worked with. In particular, the conjugacy problem in braid groups is algorithmically difficult, and it consequently provides one-way functions.

In this thesis we construct two new interactive identification schemes. One is based on the conjugacy problem and another is based on decision Diffie-Hellman (DDH) assumption. The first scheme is the primary one based on conjugacy problem over a braid group. We prove that the scheme based on conjugacy problem is secure against active attacks if the k-simultaneous conjugator search problem (k-SCSP) is intractable. Our proof is based on the fact that the conjugacy search problem (CSP) is hard in braid group, on the other hand, the conjugacy decision problem (CDP) is easy in braid group by Ko *et al.*'s algorithm. Second scheme has some limitation for adversary. That is an adversary can view only k-times interactions. Under the DDH assumption and simulator's limitation, we prove the scheme is secure against impersonation attack. *i.e.* The impersonator has negligible advantage.

1.3 Outline of the Thesis

In this thesis, we deal with security concerns regarding identification schemes that guarantee provable security against various attacks.

The rest of this paper is organized as follows: We state some preliminaries in Chapter 2. In Chapter 3 we present our identification schemes. In Chapter 4 we formally state our definition of security and give a proof of security for our schemes and compare our schemes with previous identification schemes. Finally, we end with concluding remarks in Chapter 5.

Chapter 2

Preliminaries

2.1 Braid Cryptography

The braid group were first introduced to construct a key agreement protocol and a public-key encryption scheme at CRYPTO 2000 by Ko *et al.* [29]. Within the last years various attempts have been made to derive cryptographic primitives from problems originating in combinatorial group theory. As positive results are the discovery of a hard-core predicate for the conjugacy search problem in the braid group, and implementation of braid computation, and a conversion of the public-key encryption schemes into a provable one. But to the best of our knowledge, there is no identification scheme based on conjugacy problem over a braid group published in the open literature.

2.1.1 Braid Groups

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*. The set B_n of isotopy classes of braids of index n is naturally equipped with a group structure, called the n*-braid group*, where the product of two braids x and y is nothing more than laying down the two braids in a row and then matching the end of x to the beginning of y. We give a geometric definition of braid groups in Figure 2.1.

Any braid can be decomposed as a product of simple braids. One type of simple braids is the *Artin generator* σ_i that have a single crossing between i-th

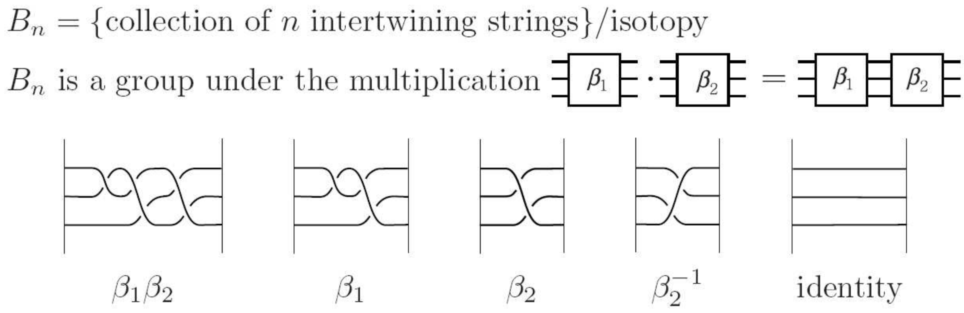
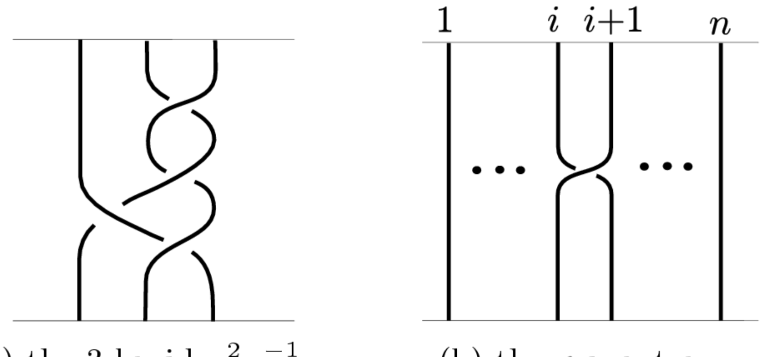

B_n = {collection of n intertwining strings}/isotopy
B_n is a group under the multiplication
β_1
β_2
β_1
β_2
$\beta_1\beta_2$
β_1
β_2
β_2^{-1}
identity

Figure 2.1: Geometric definition of braid groups

and $(i+1)$-th strand. B_n is presented with the Artin generators $\sigma_1, \ldots, \sigma_{n-1}$ and relations $\sigma_i\sigma_j = \sigma_j\sigma_i$ for $\mid i-j \mid> 1$ and $\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j$ for $\mid i-j \mid= 1$. When a braid a is expressed as a product of Artin generators, the minimum number of terms in the product is called the word length of a. An example of a braid and its generator is given in Figure 2.2



(a) the 3-braid $\sigma_2^2\sigma_1^{-1}\sigma_2$ (b) the generator σ_i

Figure 2.2: An example of braid and the generator

We have still other presentations. Let S_n be the symmetric group of an n-element set $I_n = \{1, 2, \ldots, n\}$. Let $Ref = \{(i, j) \mid 1 \leq i < j \leq n\}$ be the set of reflections (that interchange two elements and fix the other elements of I_n) in S_n and S the subset $\{(i, i+1) \mid 1 \leq i < n\}$ of Ref. We define $\ell(x)$ the *length of a permutation* x in S_n as

$$\ell(x) = min\{k \mid x_1 \cdots x_k \text{for } x_i \in S\}$$

B_n admits another presentation with generators $\{rx \mid x \in S_n\}$ and relations $r(xy) = (rx)(ry)$ if $\ell(xy) = \ell(x)+\ell(y)$. In this presentation, the longest permutation w_0 with $w_0(i) = n+1-i$ yields a braid Δ, which is called the *fundamental braid* or the *half-twist* depending on the authors. Let B_n^+ denote the submonoid of B_n generated by S_n. A braid in B_n^+ is said to be positive. A braid x is written uniquely, $x = \Delta^k x'$ where x' is in $B_n^+ - \Delta B_n^+$. This is called the normal form of x.

There is a partial order on $B_n^+ : x \leq y \Leftrightarrow y \in xB_n^+$. The ordering is inherited to S_n (We identify a permutation σ with the corresponding braid $r\sigma$ in B_n^+. We denote rS_n by Ω for simplicity reason. For a braid $x \in B_n^+$, the greatest element of the set $\{y \in \Omega \mid y \leq x\}$ is called the *left most factor* of x and denoted by $LF(x)$. A sequence of braids $(x_1, x_2, \ldots, x_k)$ in $\Omega - \{1\}$ is called the *greedy form* of x if $x_1 \cdots x_k = x$, $LF(x_i x_{i+1}) = x_i$ for all i. The above k in the greedy form is called the *Charney length* of x. This length function is easily extended to general braids using Thurston normal form.

2.1.2 Hard Problems on Braid Groups

Conjugacy Problems.

In a non-commutative group G, two elements x, y in G are *conjugate* each other, written $x \sim y$ if $y = a^{-1}xa$ for some $a \in G$. Here a or a^{-1} is called a *conjugator* and the pair (x, y) is said to be *conjugate*. Clearly $\sim$ is an equivalence relation. A simple and natural question to ask in a non-commutative group G is the conjugacy problem that can be described as a decision version and a computational version. The conjugacy decision problem(CDP) asks to determine whether $x \sim y$ for a given instance $(x, y) \in G \times G$. The conjugator search problem(CSP) asks to find $a \in G$ satisfying $y = a^{-1}xa$ for a given instance $(x, y) \in G \times G$ such that $x \sim y$. We have to be careful when we mention instances in an infinite group G. In the current information theory, it is hard to discuss a uniform distribution in G of elements described by randomly chosen information. To avoid any potential controversy, we always assume that instances to a problem are randomly chosen in a finite subset of an infinite group G restricted by system parameters.

We say a problem is *solvable* (*feasible*) if there is a deterministic finite (probabilistic polynomial-time) algorithm that outputs a solution that is accurate (accurate with non-negligible probability). The solvability is a mathematical notion and the complexity of an algorithm is not an issue as long as it is finite. A solvable problem is not necessarily feasible and vice versa.

The representation theory tells us that for any group G there are homomorphisms from G to rings that are invariant under conjugacy relation. Therefore CDP is always feasible although CDP may not be solvable. But the remaining question concerning CDP is how to construct an efficient algorithm to solve CDP with overwhelming probability.

On the other hand, there are many candidates for non-commutative groups where CSP is infeasible. However there is a normal form (such as Jordan form) of a conjugacy class in many matrix groups and so it is difficult to find a non-commutative group given as a subgroup of a matrix group that has an infeasible CSP. Therefore non-commutative groups with infeasible CSP are usually given by presentations.

We believe that CSP is infeasible in the braid groups B_n even though it is solvable. We will construct an efficient algorithm to give a solution to CDP with overwhelming accuracy. Unfortunately we do not know whether there is a polynomial-time algorithm that decides CDP.

- k-Simultaneous Conjugator Search Problem (k-SCSP)

Instance : k pairs $(x_1, x_1'), \ldots, (x_k, x_k') \in G \times G$ such that $x_i' = a^{-1}x_i a$ for all i.

Objective : Find $b \in G$ such that $x'_i = b^{-1}x_i b$ for all i

It is reasonable to believe that k-SCSP becomes easier as k increases. In particular a solution to CSP is almost unique for the braid groups and so k-SCSP is easier than CSP.

Matching Conjugacy Problems.

For a noncommutative group G, a pair $(x, x') \in G \times G$ is said to be CSP-hard if $x \sim x'$ and CSP is infeasible for the instance (x, x'). If (x, x') is CSP-hard, so is clearly (x', x). We now define two matching conjugacy problems in G that are equivalent and provide a foundation of our signature scheme.

- Matching Conjugate Search Problem (MCSP)
  Instance : A CSP-hard pair $(x, x') \in G$ and $y \in G$.
  Objective : Find $y' \in G$ such that $y \sim y'$ and $xy \sim x'y'$

- Matching Triple Search Problem (MTSP)
  Instance : A CSP-hard pair $(x, x') \in G$ and $y \in G$.
  Objective : Find a triple $(\alpha, \beta, \gamma) \in G \times G \times G$ such that $\alpha \sim x$, $\beta \sim \gamma \sim y$, $\alpha\beta \sim xy$, and $\alpha\gamma \sim x'y$

If CSP in G is infeasible, instances of MCSP or MTSP can be given as x, x', and $y \in G$ such that $x \sim x'$. In the description of the two matching problems, we do not want to exclude a group where CSP is partially infeasible, that is, the probability that a random conjugate pair (x, x') is CSP-hard is non-negligible. If a conjugate pair (x, x') is not CSP-hard, that is, an element $a \in G$ with $x' = a^{-1}xa$ can be known, then $y' = a^{-1}ya$ is a solution to MCSP and $(\alpha, \beta, \gamma) = (b^{-1}xb, b^{-1}yb, b^{-1}aya^{-1}b)$ is a solution to MTSP for any $b \in G$

and so the two matching conjugacy problems are feasible. These solutions are said to be *obvious*.

Theorem 1 *[30] In a non-commutative group G,* MCSP *is feasible if and only if* MTSP *is feasible.*

Proof. Suppose that MCSP is feasible. Let $\alpha = b^{-1}xb$ and $\beta = b^{-1}yb$ for some $b \in G$, and let γ be a solution to MCSP for the instance (x', α) and y. Then the triple (α, β, γ) is a solution to MTSP.

Suppose MTSP is feasible and MCSP is infeasible. Let (α, β, γ) is a solution to MTSP for a CSP-hard pair (x, x') and y. Since β is a solution to MCSP for a conjugate pair (x, α) and y and MCSP is infeasible, the pair (x, α) is not CSP-hard and so it is feasible to find $b \in G$ such that $\alpha = b^{-1}xb$. Similarly since γ is a solution to MCSP for a conjugate pair (x', α) and y, the pair (x', α) is not CSP-hard and so it is feasible to find $c \in G$ such that $\alpha = c^{-1}x'c$. Then $x' = cb^{-1}xbc^{-1}$ and this contradicts the fact that the pair (x, x') is CSP-hard. ■

2.1.3 Ko *et al.*'s Conjugacy Signature

Two braid-based signature schemes are introduced by Ko *et al.* in [30] : the second one is the scheme recommended by the authors, but the first is simpler and the common principle is more easily readable. Now we describe the signature schemes.

Let G be a non-commutative group where CSP is infeasible and CDP is feasible. We first give a simple conjugacy signature scheme on G and discuss its potential weakness and then we will improve it. Let $h : \{0,1\}^* \longrightarrow G$ be a hash function, that is, h is a collision-free one-way function that outputs an element of G expressed by a fixed amount of information. For example h can be given by a composition of a usual hash function of bit strings with a conversion from bit strings of a fixed length to elements of G.

Simple conjugacy signature scheme

Key generation: A public key is a CSP-hard pair (x, x') in G and a secret key is a for $x' = a^{-1}xa$.
Signing: Given a message m, a signature σ is given by a triple $\sigma = a^{-1}ya$ for $y = h(m)$.
Verifying: A signature σ is valid if and only if $\sigma \sim y$ and $x'\sigma \sim xy$.

The simple conjugacy signature scheme is a deterministic signature scheme and is clearly based on MCSP. But the secret key a is not zero-knowledge against many known message-signature pairs unless the following problem is infeasible.

Conjugacy signature scheme

Key generation: A public key is a CSP-hard pair (x, x') in G and a secret key is a for $x' = a^{-1}xa$.
Signing: Given a message m, choose $b \in G$ at random and let $\alpha = b^{-1}xb$ and $y = h(m\|\alpha)$, then a signature σ is given by a triple $\sigma = (\alpha, \beta, \gamma)$ where $\beta = b^{-1}yb$ and $\gamma = b^{-1}aya^{-1}b$.
Verifying: A signature σ is valid if and only if $\alpha \sim x$, $\beta \sim \gamma \sim y$, $\alpha\beta \sim xy$, and $\alpha\gamma \sim x'y$.

The conjugacy signature scheme is clearly based on MTSP that is equivalent to MCSP. In the conjugacy signature scheme, the secret key a is zero-knowledge unless 2-SCSP is feasible no matter how many message-signature pairs are known. Indeed b can be known from each message-signature pair if 2-SCSP is feasible and so many (y, aya^{-1}) pairs are known for the secret key a.

2.1.4 Random braids

Since braid groups are infinite and every cryptosystem has to run under finite resources, we first need to establish system parameters to confine the infinite group to a finite environment.

We first fix positive integers n, ℓ, d as system parameters. Let

$$B_n(\ell) = \{b \in B_n | 0 \leq \inf(b), \sup(b) \leq \ell\}.$$

Then $|Bn(\ell)| \leq (n!)^\ell$ and so it is finite. A *random braid generator* produces $b \in_R Bn(\ell)$ in $\mathcal{O}(\ell n)$ time using the random braid generator. A bit-string to braid conversion $c : \{0,1\}^N \to B_n(\ell)$ for $N = \ell \lfloor log2n! \rfloor$ can be done in $\mathcal{O}(\ell n)$ time as follows: For a bit string $r \in \{0,1\}^N$, cut r into of blocks $r_1 \| r_2 \| \cdots \| r_\ell$ of bit-length $\lfloor log2n! \rfloor$ and then for each $r_i \in [0, n! - 1]$, write $r_i = \sum_{k=1}^{n-1} a_k k!$ by recursively dividing r_i by 2 through $n-1$ so that $0 \leq a_k \leq k-1$ and then apply the random braid generator to the sequence $a_{n-1}, \dots, a_1$. We think that the values of our random braid generator and bit-string to braid conversion distribute almost uniformly in $B_n(\ell)$ for a small ℓ. We will suggest $\ell = 3$ and so the distribution will not cause much a problem. For a large ℓ, they can be replaced by slower algorithms with better distribution.

For $x, y \in B_n$ such that $x \sim y$, the distance $d(x, y)$ between x and y is defined by $\min\{\ell(b) | y = b^{-1}xb\}$. The distance behaves like a metric in a conjugacy class except the fact that $d(x, \tau(x)) = 0$. For example one can show $d(x, y) = d(y, x)$ by using $\inf(b^{-1}) = -\sup(b)$ and $\sup(b^{-1}) = -\inf(b)$.

Random Super Summit Braid Generator

From now on, we assume $x \in SSS(x)$ and $\inf(x) = 0$ and $\sup(x) = 0$. Then $SSS(x) \subset B_n(\ell)$. Define the *d-neighborhood* $S(x, d)$ of x in $SSS(x)$ as follows:

$$S(x, d) = \{y \in SSS(x) | d(x, y) \leq d\}.$$

For a randomly chosen $x' \in S(x, d)$, we will use a conjugate pair (x, x') as a public key. Thus the pair (x, x') must be CSP-hard. The cardinality

$|S(x,d)|$ seems an obvious choice for the security level and it will depend on all of n, ℓ, d, x and in particular on the location of x inside $SSS(x)$. Unfortunately we do not know how to estimate a lower bound for $|S(x,d)|$. A positive braid a is called a *minimal super summit conjugator* of $x \in SSS(x)$ if a is minimal among all positive braids b satisfying $b^{-1}xb \in SSS(x)$ with respect to the partial order '$\leq$'. Since $\Delta^{-1}x\Delta \in SSS(x)$, a minimal super summit conjugator is a permutation braid. Since any minimal super summit conjugator must be greater than or equal to at least one generator σ_i, there are at most $n-1$ minimal super summit conjugators of a given n-braid x. An algorithm to generate $SSS(x)$ is proposed using minimal super summit conjugators. The running time of the algorithm is obviously proportional to the size of $SSS(x)$. Consider the directed graph $\Gamma(x)$ where the super summit set $SSS(x)$ is the set of vertices and there is a directed edge from x_1 to x_2 if $x2 = a^{-1}x_1a$ for some minimal super summit conjugator a of x_1. We believe that the higher the out-going valency near x in the graph $\Gamma(x)$ is, the larger the d-neighborhood $S(x,d)$ of x is. It is not hard to write an heuristic algorithm to pick a good braid x by investigating valencies.

We now describe how to generate a random braid in $S(x,d)$. This procedure will be called a *random super summit braid generator* denoted by $RSSBG(x,d) = (x', a)$ where $x' \in_R S(x,d)$ and $a \in B_n(d)$ such that $x' = a^{-1}xa$. We first choose $b \in_R Bn(5\ell)$ if $\ell(x) = \ell$. Then we apply a random sequence of cyclings and decyclings to $b^{-1}xb$ until we obtain a braid $a^{-1}xa \in SSS(x)$. According to [5], the length of this sequence is at most in n^2 and it is much smaller in an average case. If $\ell(a) \leq d$, then $a^{-1}xa$ is the output. Otherwise we start over again by choosing new b. Our experiment shows the probability of success on each run is over 70% if $d = \ell + 1$.

2.2 Decision Diffie-Hellman Problem

Let p and q be large primes, such that q divides $p-1$. Let G be the subgroup of order q in $\mathbb{Z}_p^*$. Let $g \in G$ and $a, b \in \{0, 1, \ldots, q-1\}$ be randomly chosen. Then, the Diffie-Hellman assumption says that it is impossible to compute g^{ab} from g^a and g^b.

Let $g_1 = g^a$, $g_2 = g^b$ and g_3 be given. The *decision Diffie-Hellman problem* (DDH) is to decide if

$$g_3 = g^{ab}$$

This is equivalent to deciding whether

$$\begin{aligned} \log_g(g_3) &= \log_g(g_1)\log_g(g_2), \text{ or} \\ \log_{g_2}(g_3) &= \log_g(g_1). \end{aligned}$$

The *decision Diffie-Hellman assumption* says that no efficient algorithm exists to solve the decision Diffie-Hellman problem if a, b and g_3 (g_1, g_2, and g_3, respectively) are chosen at random (and independently).

Definition 1 *For every polynomial Q and PPT algorithm A,*

$$|\Pr[A(g, g^x, g^y, g^{xy}) = \text{“}true\text{”}] - \Pr[A(g, g^x, g^y, g^c) = \text{“}true\text{”}]| < \frac{1}{Q(k)}$$

for all sufficiently large k. x, y, and c are chosen at random from Z_q, where q is a prime such that $q|p-1$.

2.3 Identification Schemes

2.3.1 Identification

We now describe the definition of identification scheme. An *identification* scheme $\mathcal{ID} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ is a triple of randomized algorithms. On input security parameter $k \in \mathbb{N}$, the *poly*(k)-time key-generation algorithm $\mathcal{G}$ returns a pair consisting of a public key *pk* and a matching secret key *sk*. $\mathcal{P}$ and $\mathcal{V}$ are polynomial-time algorithms that implement the prover and verifier, respectively. We require the natural correctness condition, namely that the boolean decision produced by $\mathcal{V}$, in the interaction in which $\mathcal{P}$ has input *pk*; *sk* and $\mathcal{V}$ has input *pk*, is one with probability one. This probability is over the coin tosses of both parties. We assume that the first and last moves in the interaction always belong to the prover.

Definition 2 *An identification scheme consists of two stages:*

1. *Initialization : In this stage, each user generates a secret key and a public key by using probabilistic polynomial-time generation algorithm* $\mathcal{G}$ *on input of the key size. A link between each user and its public key is established. Note that in some schemes a part of the public key can be commonly shared among all users as a system parameter.*

2. *Operation : In this stage, any user can demonstrate its identity to a verifier by performing some identification protocol related to its public key, where the input for the verifier is the public key. At the conclusion of this stage, the verifier either outputs* "accept" *or* "reject"*.*

2.3.2 Interactive Proof System

There are two participants in an interactive proof system, the *prover* and the *verifier*. Prover knows some fact (e.g. a secret key *sk* of a public-key cryptosystem or a square of a quadratic residue s), which we call *the secret of the prover.* In an *interactive proof of knowledge*, prover wishes to convince verifier that he/she knows the secret of the prover. Prover and verifier alternately perform *moves* consisting of:

1. Receive a message from the opposite party.

2. Perform some computation.

3. Send a message to the opposite party.

Usually, prover starts and verifier finishes the protocol. In the first move, prover does not receive a message. The interactive proof may consist of several *rounds*. This means that the protocol specifies a sequence of moves, and this sequence is repeated a specified number of times. Typically, a move consists of a challenge by verifier and a response by prover. Verifier accepts or rejects prover's proof, depending on whether prover successfully answers all of verifier's challenges.

Proofs in interactive proof systems are quite different from proofs in mathematics. In mathematics, the prover of some theorem can sit down and prove the statement by himself. In interactive proof systems, there are two computational tasks, namely producing a proof (prover's task) and verifying its validity (verifier's task). Additionally, communication between the prover and verifier is necessary. We describe the 3-way interactive protocol in Figure 2.3.

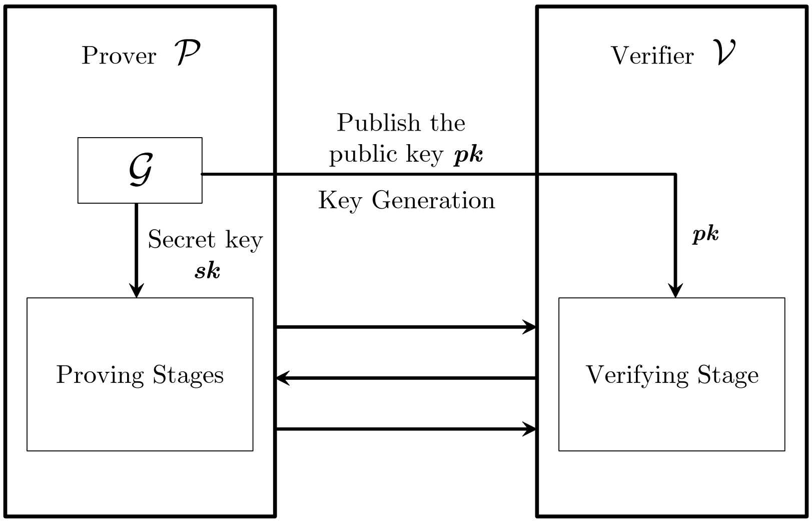


Figure 2.3: 3-way interactive protocol

We have the following requirements for interactive proof systems.

1. *(completeness).* If prover knows the prover's secret, then verifier will always accept prover's proof.

2. *(soundness).* If prover can convince verifier with reasonable probability, then he/she knows the prover's secret.

If the prover and the verifier of an interactive proof system follow the behavior specified in the protocol, they are called an *honest verifier* and an *honest prover*. A prover who does not know the secret of the prover an tries to convince the verifier is called a *cheating or dishonest prover*. Sometimes, the verifier can get additional information from the prover if he/she does not follow the protocol. Note that each prover (or verifier), whether he/she is honest or not, fulfills the syntax of the communication interface, because not following the syntax is immediately detected. She may only be dishonest in her private computations and the resulting data that he/she transmits.

Password Scheme

In a simple password scheme, $\mathcal{P}$ uses a secret password to prove his/her identity. The password is the only message, and it is sent from the prover to the verifier. $\mathcal{V}$ accepts prover's identity if the transmitted password and the stored password are equal. Here, only one message is transmitted, and obviously the scheme meets the requirements. If $\mathcal{P}$ knows the password, verifier accepts. If a cheating prover does not know the password, verifier does not accepts. The problem is that everyone who observed the password during communication can use the password.

Identity Based on Public-Key Encryption

First of all, we recall the basic scenario of identification scheme based on a public-key cryptosystem. Each prover has a secret key *sk* only known to him/her and a public key *pk* known to everyone. Suppose that everyone who can decrypt a randomly chosen encrypted message must know the secret key. This assumption should be true if the cryptosystem is secure. Hence, the secret key *sk* can be used to identify prover.

Prover proves his/her identity to verifier using the following steps:

1. Verifier chooses a random message m, encrypts it with the public key *pk* and sends the cryptogram c to prover.

2. Prover decrypts c with his/her secret key *sk* and sends the result m' back to verifier.

3. Verifier accepts the identity of prover if and only if $m = m'$.

Two messages are exchanged: it is a *two-move* protocol. The completeness of the scheme is obvious. On the other hand, a cheating prover who only knows the public key and a ciphertext should not be able to find the plaintext better than guessing at random. The probability that verifier accepts if the prover does not know the secret of the prover is very small. Thus, the scheme is also sound. This reflects verifier's security requirements. Suppose that an adversary observed the exchanged messages and later wants to impersonate prover. Verifier chooses m at random and computes c. The probability of obtaining the previously observed c is very small. Thus, adversary cannot take advantage of observing the exchanged messages. At first glance, everything seems to be all right. However, there is a security problem if verifier is not honest and does not follow the protocol in step 1. If, instead of a randomly chosen encrypted message, he/she sends a cryptogram intended for prover, then he/she lets prover decrypt the cryptogram. He/She thereby manages to get the plaintext of a cryptogram which he/she could not compute by himself. This violates prover's security requirements.

2.3.3 Zero-Knowledge

In the interactive proof system based on a public-key cryptosystem, which we discussed above, a dishonest verifier can decrypt prover's cryptograms by interacting with prover. Since verifier is not able to decrypt them without interaction, he/she learns something new by interacting with prover. He/She obtains *knowledge* from prover. This is not desirable, because it might violate prover's security requirements as our example shows. It is desirable that interactive proof systems are designed so that no knowledge is transferred from the prover to the verifier. Such proof systems are called zero-knowledge if whatever the verifier can efficiently compute after interacting with the prover, can be efficiently simulated without interaction. Below we define the zero-knowledge property more formally.

We denote the algorithm that the honest prover executes by $\mathcal{P}$, the algorithm of an honest verifier by $\mathcal{V}$ and the algorithm of a general (possibly dishonest) verifier by $\mathcal{V}^*$. The interactive proof system (including the interaction between $\mathcal{P}$ and $\mathcal{V}$) is denoted by $(\mathcal{P}, \mathcal{V})$. Prover knows a secret about some object x (e.g. as in the Fiat-Shamir example, the root of a square x). This object x is the common input to $\mathcal{P}$ and $\mathcal{V}$.

Each algorithm is assumed to have polynomial running time. It may be partly controlled by random events, that is, the algorithm has access to a source of random bits and thus can make random choices. Such algorithms are called probabilistic algorithms.

Let x be the common inputs of $(\mathcal{P}, \mathcal{V})$. Suppose, the interactive proof takes n moves. A message is sent in each move. For simplicity, we assume that the prover starts with the first move. We denote by m_i the message sent in the i-th move. The messages $m_1, m_3, \ldots$ are sent from the prover to the verifier and the messages $m_2, m_4, \ldots$ are sent from the verifier to the prover. The *transcript* of the joint computation of $\mathcal{P}$ and $\mathcal{V}^*$ on input x is defined by

$$tr_{\mathcal{P},\mathcal{V}^*}(x) := (m_1, \ldots, m_n)$$

where $tr_{\mathcal{P},\mathcal{V}^*}(x)$ is called an accepting transcript if $\mathcal{V}^*$ accepts after the last move. Note that the transcript $tr_{\mathcal{P},\mathcal{V}^*}(x)$ depends on the random bit that the algorithms $\mathcal{P}$ and $\mathcal{V}^*$ choose. Thus, it is not determined by the input x.

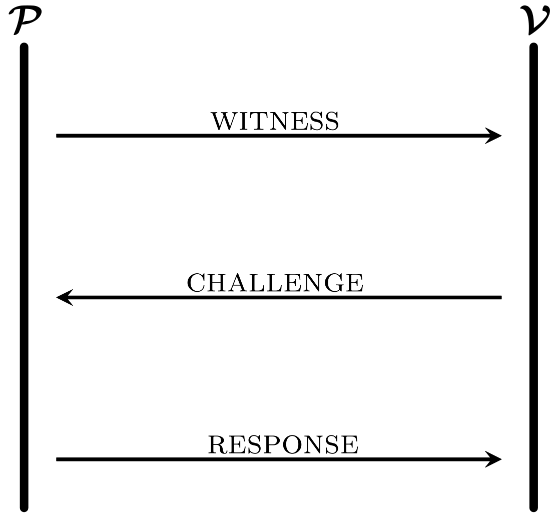


Figure 2.4: Zero-knowledge identification scheme

Definition 3 *An interactive proof system $(\mathcal{P},\mathcal{V})$ is (perfect) zero-knowledge if there is a probabilistic simulator $\mathcal{S}(\mathcal{V}^*,x)$, running in expected polynomial time, which for every verifier $\mathcal{V}^*$ (dishonest or not) outputs on input x an accepting transcript t of $\mathcal{P}$ and $\mathcal{V}^*$, such that these simulated transcripts are distributed in the same way as if they were generated by the honest prover $\mathcal{P}$ and $\mathcal{V}^*$.*

Remark 1 *The definition of zero-knowledge included all verifiers (also the dishonest ones). Hence, zero-knowledge is a property of the prover $\mathcal{P}$. It captures the prover's security requirements against attempts to gain "knowledge" by interacting with him.*

To understand the definition, we have to clarify how a simulator works. A simulator $\mathcal{S}$ is an algorithm which, given some verifier $\mathcal{V}^*$, honest or not, generates valid accepting transcripts for $(\mathcal{P}, \mathcal{V}^*)$, without communicating with the real prover $\mathcal{P}$. In particular, $\mathcal{S}$ does not have any access to computations that rely on the secret of the prover. Trying to produce an accepting transcript, $\mathcal{S}$ plays the role of $\mathcal{P}$ in the protocol and communicates with $\mathcal{V}^*$. Thus, he obtains outgoing messages of $\mathcal{V}^*$ which are compliant with the protocol. His/Her task is to fill into the transcript the messages going out from $\mathcal{P}$. Since $\mathcal{P}$ computes these messages by use of his/her secret and $\mathcal{S}$ does not know this secret, $\mathcal{S}$ applies his/her own strategy to generate the messages. Necessarily, his probability of obtaining a valid transcript in this way is significantly less than 1. Otherwise, with high probability, $\mathcal{S}$ could falsely convince $\mathcal{V}^*$ that he knows the secret, and the proof system is not sound. Thus, not every attempt of $\mathcal{S}$ to produce an accepting transcript is successful; he/she fails in many cases. Nevertheless, by repeating his/her attempts sufficiently often, the simulator is able to generate a valid accepting transcript. It is required that the expectation value of the running time, which $\mathcal{S}$ needs to get an accepting transcript, is bounded by a polynomial in the binary length $|x|$ of the common input x.

To be zero-knowledge, the ability to produce accepting transcript by a simulation is not sufficient. The generation of transcripts, real or simulated, includes random choices. Thus, we have a probability distribution on the set of accepting transcripts. The last condition in the definition means that the probability distribution of the transcripts, which are generated by the simuator $\mathcal{S}$ and $\mathcal{V}^*$, is the same as if they were generated by the honest prover $\mathcal{P}$ and $\mathcal{V}^*$.

2.3.4 Attack Types

In general, an identification scheme is said to *be broken if an adversary succeeds in an impersonation attempt* (making the verifier accept with non-negligible probability). The methods an adversary may employ in an attempt to defeat identification protocol are summarized in Table 2.1 [35]. We can divide them into two types–passive attack and active attack–according to the interaction allowed to the adversary before an impersonation attempt [46, 35].

The weakest form of attack is a *passive attack*, where the adversary is not allowed to interact with the system at all before attempting an impersonation; the only available information to the adversary is the public key of the prover. Other attacks of intermediate level such as *eavesdropping attack* or *honest-verifier attack* are essentially equivalent to the passive attack.

The strongest form of attack is an *active attack*, in which the adversary is allowed to interact with $\mathcal{P}$ several times, posing as $\mathcal{V}$. We may consider active attacks as adaptive chosen ciphertext attacks. We should note that active attacks are quite feasible in practice.

2.3.5 The Schnorr scheme and its variants

The Schnorr protocol [43] is an alternative of the FS and GQ protocols whose security is based on the intractability of DLP. The design allows pre-computation, reducing the real-time computation for the prover to one multiplication modulo a prime q; it is particularly suitable for provers of limited computational ability. A further important computational efficiency results from the use of a subgroup of order q of the multiplicative group of integers modulo p, where $q|(p-1)$; this also reduces the required number of transmitted bits. Finally, the protocol was designed to require only three passes. The Schnorr protocol is depicted on Figure 2.5. Brickell and McCurley [9] propose a modification of Schnorr's identification scheme, in which q is kept secret and exponent computations are reduced modulo $p-1$ rather than q. A major drawback is that almost 4 times as much computation is required by the prover. Another variant of Schnorr's scheme by Girault [21] was the first identity-based identification scheme based on DLP. A further variation of Schnorr's identification protocol by Okamoto [37] is provably secure; it does,

Table 2.1: Types of attacks on identification protocols

Types of attacks	Descriptions
impersonation	a deception whereby one entity purports to be another.
replay attack	an impersonation or other deception involving use of information from a single previous protocol execution, on the same time or a different verifier.
interleaving attack	an impersonation or other deception involving selective combination of information from one or more previous or simultaneously ongoing protocol executions, including possible origination of one or more protocol executions by an adversary itself.
reflection attack	an interleaving attack involving sending information from an ongoing protocol execution back to the originator of such information.
forced delay	a forced delay occurs when an adversary intercepts a message, and relays it at some later point in time.
chosen-text attack	an attack on a challenge-response protocol wherein an adversary strategically chooses challenges in an attempt to extract information about the prover's long-term key.

however, involve some additional computation. Popescu [40] shows how the interactive identification scheme based on the elliptic curve discrete logarithm problem (ECDLP) is constructed.

Aside from the above protocols based on the computational intractability of the standard number-theoretic problems, a number of very efficient identification protocols have more recently been proposed based on **NP**-hard problems. Stern [47] proposed a practical zero-knowledge identification scheme based on the **NP** hard *syndrome decoding* problem. Stern [48] proposed another practical identification scheme based on an **NP** hard combinatorial *constrained linear equations* problem, offering a very short key length, which is of particular interest in specific applications. Pointcheval [39] proposed

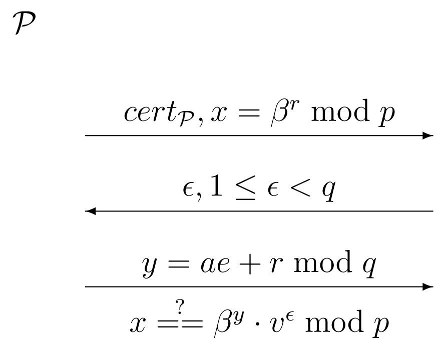

$\mathcal{P}$
$\mathcal{V}$
$cert_{\mathcal{P}}, x = \beta^r \bmod p$
$\epsilon, 1 \leq \epsilon < q$
$y = ae + r \bmod q$
$x \overset{?}{==} \beta^y \cdot v^\epsilon \bmod p$

Figure 2.5: The Schnorr identification protocol

another such scheme based on the **NP**-hard *perceptrons problem*: given an $m \times n$ matrix M with entries ±1, find an n-vector y with entries ±1 such that $M_y \geq 0$.

2.3.6 The Kim-Kim Scheme

The Basic Scheme.

For a security parameter k, a pair of secret and public parameters is generated as follows:

Key generation.
On input k, the key generation algorithm $\mathcal{G}$ works as follows:

1. Generate two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order m for some large prime m and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

2. Generate an arbitrary generator $P \in \mathbb{G}_1$.

3. Choose randomly $a, b, c \in \mathbb{Z}_m^*$ and compute $v = \hat{e}(P, P)^{abc}$.

4. The public parameter is $\mathsf{Pub} = \langle \mathbb{G}_1, \mathbb{G}_2, P, aP, bP, cP, \hat{e}, v \rangle$, and the secret parameter is $\mathsf{Sec} = \langle a, b, c \rangle$. And then publish them.

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
As is the case for other identification schemes, this scheme consists of several rounds. The protocol executes just once the following:

1. $\mathcal{P}$ chooses $r_1, r_2, r_3 \in \mathbb{Z}_m^*$ at random, computes $x = \hat{e}(P, P)^{r_1 r_2 r_3}$, $Q = r_1 r_2 r_3 P$, and sends $\langle x, Q \rangle$ to $\mathcal{V}$.
2. $\mathcal{V}$ picks $\omega \in \mathbb{Z}_m^*$ at random, and sends $R = \omega P$ to $\mathcal{P}$.
3. On receiving R, $\mathcal{P}$ sets $S = r_1 r_2 r_3 R$, computes $Y \in \mathbb{G}_1$ such that

$$Y = abcP + (a + b + c)S,$$

and sends it to $\mathcal{V}$; $\mathcal{V}$ accepts $\mathcal{P}$'s proof of identity if both $x = \hat{e}(P, Q)$ and $\hat{e}(Y, P) = v \cdot \hat{e}(aP + bP + cP, Q)^{\omega}$, and rejects otherwise.

This protocol is represented graphically in Figure 2.6. Once after this protocol can be proved to be secure against active adversaries, it can be extended to a generalized protocol.

Generalized scheme.

We now describe a generalized version KK identification scheme. The generalized identification scheme extends the basic scheme in Section 2.3.6 using k random numbers. The key generation algorithm $\mathcal{G}$ is similar to that of the basic scheme except generating k random numbers.

Key generation.
On input k, the key generation algorithm $\mathcal{G}$ works as follows:

$$\mathcal{P} \qquad\qquad\qquad\qquad\qquad\qquad \mathcal{V}$$

$$\xrightarrow{\quad x = \hat{e}(P,P)^{r_1 r_2 r_3}, Q = r_1 r_2 r_3 P \quad}$$

$$\xleftarrow{\quad R = \omega P,\text{where } \omega \in \mathbb{Z}_m^* \quad}$$

$$\xrightarrow[\hat{e}(Y,P) \overset{?}{==} v \cdot \hat{e}(aP + bP + cP, Q)^{\omega}]{\quad Y = abcP + (a+b+c)S, \text{ where } S = r_1 r_2 r_3 R \quad}$$

Figure 2.6: The SAA identification protocol

1. Generates two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order m for some large prime m and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

2. Generates an arbitrary generator $P \in \mathbb{G}_1$.

3. Chooses randomly $a_1, \dots, a_{3\ell} \in \mathbb{Z}_m^*$ and computes $v_1 = \hat{e}(P,P)^{a_1 a_2 a_3}, \cdots, v_\ell = \hat{e}(P,P)^{a_{3\ell-2} a_{3\ell-1} a_{3\ell}}$.

4. The public parameter is $\mathsf{Pub} = \langle \mathbb{G}_1, \mathbb{G}_2, P, a_1 P, \dots, a_{3\ell} P, \hat{e}, v_1, \cdots, v_\ell \rangle$, and the secret parameter is $\mathsf{Sec} = \langle a_1, \dots, a_{3\ell} \rangle$. And then publishes them.

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
The generalized scheme is similar to the basic scheme, however, each round is performed in parallel as follows:

1. $\mathcal{P}$ chooses $r_1, r_2, r_3 \in \mathbb{Z}_m^*$ at random, computes $x = \hat{e}(P,P)^{r_1 r_2 r_3}$, $Q = r_1 r_2 r_3 P$, and sends $\langle x, Q \rangle$ to B.

2. $\mathcal{V}$ picks $\omega_1, \dots, \omega_\ell \in \mathbb{Z}_m^*$ at random, and sends $R_1 = \omega_1 P, \dots, R_\ell = \omega_\ell P$ to $\mathcal{P}$.

3. On receiving ℓ random values, $\mathcal{P}$ sets

$$S_1 = r_1r_2r_3R_1, S_2 = r_1r_2r_3R_2, \ldots, S_\ell = r_1r_2r_3R_\ell,$$

computes Y such that

$$Y = \sum_{i=1}^{\ell} a_{3i-2}a_{3i-1}a_{3i}P + \sum_{i=1}^{\ell}(a_{3i-2} + a_{3i-1} + a_{3i})S_i$$

and sends it to $\mathcal{V}$; $\mathcal{V}$ accepts if both $x = \hat{e}(P, Q)$ and $\hat{e}(Y, P) = \prod_{i=1}^{\ell} v_i \cdot \hat{e}(a_{3i-2}P + a_{3i-1}P + a_{3i}P, Q)^{\omega_i}$, and rejects otherwise.

The KK scheme is more efficient than the Schnorr scheme and the Okamoto scheme with respect to preprocessing of prover and on-line processing overhead of both parties (prover and verifier). At the same time, security of the KK scheme is higher than or equal to previous schemes. The authors prove that the KK scheme is secure against active attacks as well as passive attacks if the bilinear Diffie-Hellman problem is intractable. The proof is based on the fact that the computational Diffie-Hellman problem is hard in the additive group of points of an elliptic curve over a finite field, on the other hand, the decisional Diffie-Hellman problem is easy in the multiplicative group of the finite field mapped by a bilinear map.

Chapter 3
Our Proposed Scheme

In this chapter we propose two identification schemes. The first one is based on conjugacy problem over a braid group and another is based on DDH problem.

3.1 Scheme I : based on Conjugacy Problem

In this section, we introduce an identification scheme based on conjugacy problem over a braid group. Let B_n be a braid group where CSP is infeasible and CDP is feasible. Let $h : \{0,1\}^* \longrightarrow B_n$ be a hash function, that is, h is a collision-free one-way function that outputs an element of B_n expressed by a fixed amount of information. For example h can be given by a composition of a usual hash function of bit strings with a conversion from bit strings of a fixed length to elements of B_n. We recommend the security parameter $n = 20,\ n = 24$ and $n = 28$ depending on a use of our identification scheme.

Our proposed identification scheme consists of two stages, key generation and protocol actions between prover and verifier. We will describe the details.

Key generation. On input k, the key generation algorithm $\mathcal{G}$ works as follows:

1. Generate a braid group B_n.

2. Generate a CSP-hard pair $(x, x') \in B_n \times B_n$ such that $x' = a^{-1}xa$.

3. The public parameter is $\mathsf{Pub} = \langle B_n, (x, x')\rangle$, and the secret parameter is $\mathsf{Sec} = \langle a\rangle$. And then publish them.

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
As is the case for other identification schemes, our protocol consists of $\mathit{\Delta}$-times challenge-response protocol where Δ is a security parameter as usual identification protocol. The 1 round challenge-response protocol is described as follows:

1. $\mathcal{P}$ chooses $s \in B_n$ at random, computes $X = s^{-1}xs$ and $X' = a^{-1}Xa$, and sends $\langle X, X'\rangle$ to $\mathcal{V}$.
2. $\mathcal{V}$ picks $r \in B_n$ at random, and sends r to $\mathcal{P}$.
3. On receiving r, $\mathcal{P}$ computes $\alpha, y, \beta,$ and γ such that

$$\begin{aligned} \alpha &= r^{-1}Xr \\ y &= h(X\|\alpha) \\ \beta &= r^{-1}yr \\ \gamma &= r^{-1}aya^{-1}r \end{aligned}$$

and sends them to $\mathcal{V}$; $\mathcal{V}$ accepts $\mathcal{P}$'s proof of identity if and only if all of the followings are satisfied and otherwise rejects.

$$\mathcal{V} \text{ outputs } \mathsf{accept} \text{ when : } \begin{cases} \alpha = r^{-1}Xr \\ Xx \sim X'x' \\ \alpha \sim X \\ \beta \sim \gamma \sim y \\ \alpha\beta \sim Xy \\ \alpha\gamma \sim X'y \end{cases}$$

Our proposed scheme repeats Δ-times of the Protocol actions between $\mathcal{P}$ and $\mathcal{V}$. This identification scheme is represented graphically in Figure 3.1. Once after this scheme can be proved to be secure against passive adversaries.

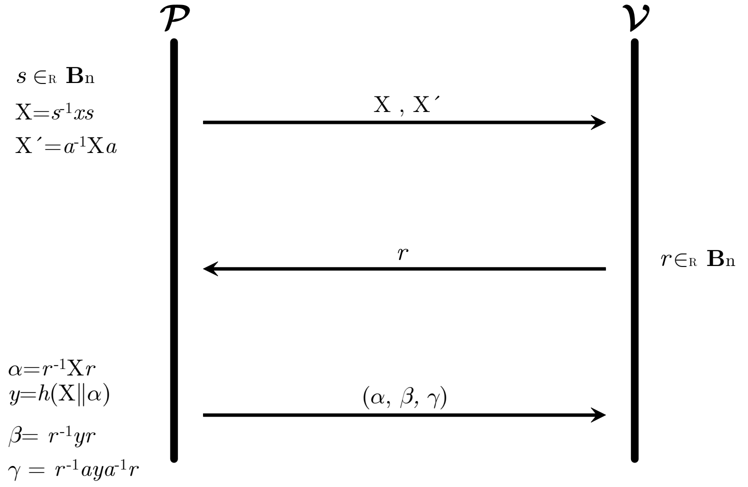


Figure 3.1: Proposed Scheme

3.2 Scheme II : based on DDH assumption

Now we introduce a new identification scheme based on DDH problem.

Key generation. On input k, the key generation algorithm $\mathcal{G}$ works as follows:

It is almost the same as in the basic scheme except that in this scheme we use two generators. This is accomplished selecting a random element g_1 of order q modulo p. The group G is then set to be the subgroup of $\mathbb{Z}_p^*$ generated by g_1, *i.e.* $G = \{g_1^i \bmod p : i \in \mathbb{Z}_q\} \subset \mathbb{Z}_p^*$. A random $w \leftarrow_R \mathbb{Z}_q$ is then chosen and compute $g_2 = g_1^w$. Then, two random k-degree polynomials $p_1 = \sum_{t=0}^{k} d_t x^t$ and $p_2 = \sum_{t=0}^{k} d'_t x^t$ are chosen over $\mathbb{Z}_q$. Next the algorithm computes $D_0 = g_1^{d_0} g_2^{d'_0}, \ldots, D_k = g_1^{d_k} g_2^{d'_k}$. Finally, it outputs

$\mathsf{Pub} = \langle g_1, g_2, D_0, \ldots, D_k \rangle$ and $\mathsf{Sec} = \langle p_1, p_2 \rangle$, and publishes the Pub.

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
Our challenge-response protocol is described as follows:

1. $\mathcal{P}$ chooses $x, \mu_0, \text{and } \mu_1 \in \mathbb{Z}_q$ at random, and sends $x, \mu_0, \text{and } \mu_1$ to $\mathcal{V}$.

2. $\mathcal{V}$ computes the followings:

$$\begin{aligned} r_1 &\leftarrow_R \mathbb{Z}_q && (3.1) \\ u_1 &= g_1^{r_1} && (3.2) \\ u_2 &= g_2^{r_1} && (3.3) \\ D_x &= \prod_{t=0}^{k} D_t^{x^t} && (3.4) \\ D &= D_x^{r_1} && (3.5) \\ \mu_i &\leftarrow_R \{\mu_0, \mu_1\} && (3.6) \\ c &= \mu_i D && (3.7) \end{aligned}$$

   And sends $u_1, u_2, \text{and } c$ to $\mathcal{P}$.

3. On receiving $u_1, u_2, \text{and } c$, $\mathcal{P}$ computes $D' = u_1^{p_1(x)} u_2^{p_2(x)}$ and sends $\mu_{i^*}, D', p_1(x), \text{and } p_2(x)$ to $\mathcal{V}$;
   $\mathcal{V}$ accepts $\mathcal{P}$'s proof of identity if and only if both of the condition $\mu = cD'^{-1}$ and $\mu_{i^*} = \mu_i$ and otherwise rejects.

In this scheme, we can transfer the scheme as ID-based variant. ID-based version of scheme II needs the **Setup** stage.
Setup. A probabilistic algorithm used by the center to set up all the parameters of the scheme. The algorithm $\mathcal{S}$ takes as input 1^s and generates the global system parameters param and the master-key. The system parameters will be publicly known while the master-key will be known to the center only. And a probabilistic algorithm used by the center to extract a private

key corresponding to a given identity. The algorithm $\mathcal{S}$ receives as input the $\mathsf{master\text{-}key}$ and a public identity ID, it returns the corresponding private key Sec.

Chapter 4 Security Analysis

4.1 Security of Identification Schemes

We define a secure identification scheme based on the definition given by Feige *et al.* [18]

Definition 4 *A prover $\mathcal{P}$ (resp. verifier $\mathcal{V}$) is a "good" prover denoted by $\overline{\mathcal{P}}$ (resp. "good" verifier denoted by $\overline{\mathcal{V}}$), if it does not deviate from the protocols dictated by the scheme. Let $\widetilde{\mathcal{P}}$ be a fraudulent prover who does not complete the Initialization stage of where Definition 2 as $\mathcal{P}$ and may deviate from the protocols (so another person/machine can simulate $\widetilde{\mathcal{P}}$). $\widetilde{\mathcal{V}}$ is not a good $\mathcal{V}$. $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{V}}$ are assumed to be polynomial time bounded machines, which may be nonuniform.*

An identification scheme $(\mathcal{P}, \mathcal{V})$ is secure if

1. *$(\overline{\mathcal{P}}, \overline{\mathcal{V}})$ succeeds with overwhelming probability.*
2. *There is no coalition of $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{V}}$ with the property that, after a polynomial number of executions of $(\overline{\mathcal{P}}, \widetilde{\mathcal{V}})$ and relaying a transcript of the communication to $\widetilde{\mathcal{P}}$, it is possible to execute $(\widetilde{\mathcal{P}}, \overline{\mathcal{V}})$ with non-negligible probability of success. The probability is taken over the distribution of the public key and the secret key as well as the con tosses of $\overline{\mathcal{P}}$, $\widetilde{\mathcal{V}}$, $\widetilde{\mathcal{P}}$, and $\overline{\mathcal{V}}$, up to the time of the attempted impersonation.*

When an identification scheme is "witness hiding" [20] and an interactive proof of "knowledge" [18], this scheme is secure in the sense of Definition 4.

This is roughly because if there exists $(\widetilde{\mathcal{P}}, \widetilde{\mathcal{V}})$ with non-negligible probability of success, we can construct a knowledge extractor (from "knowledge soundness"), which leads to contradiction with "witness hiding". Thus there are two ways to prove the security of Definition 4: One is to prove it directly as in [18, 36], and the other way is to prove that a scheme is "witness hiding" and an interactive proof of "knowledge". Some scheme such as [36] seems to be proven only in the former way, since the knowledge soundness is sometimes hard to prove.

4.2 Analysis of Scheme I

Our identification scheme is clearly based on MTSP that is equivalent to MCSP. Now we analyze the scheme I by using proof of knowledge method.

Our propsed protocol has zero-knowledge property unless k-SCSP is feasible no matter how many witness-response pairs are known.

Theorem 2 *The proposed scheme is exhibits a zero-knowledge proof of knowledge.*

Proof.
Completeness. Prover who knows the secret value a can convince the verifier of his identity with probability 1. Honest prover can compute the values, $X, X', \alpha, y, \beta,$ and γ for any random challenge value r from verifier. After receiving the (α, β, γ), verifier outputs the 'accept' with probability 1. Because the verifier always check the verifying equation easily by using the conjugacy decision algorithm.

$$\alpha = r^{-1}Xr \ ; \quad \text{so, } \alpha \sim X$$

$$X'x' = a^{-1}Xaa^{-1}xa = a^{-1}Xxa, \quad Xx \sim X'x'$$

$$\beta = r^{-1}yr \ ; \quad \text{so, } \beta \sim y$$

$$\gamma = r^{-1}aya^{-1}r = (a^{-1}r)^{-1}y(a^{-1}r) \ ; \quad \text{so, } \gamma \sim y$$

$$\alpha\beta = r^{-1}Xrr^{-1}yr = r^{-1}Xyr \ ; \quad \text{so, } \alpha\beta \sim Xy$$

$$\begin{aligned} \alpha\gamma &= r^{-1}Xrr^{-1}aya^{-1}r \\ &= r^{-1}Xaya^{-1}r \\ &= r^{-1}aX'ya^{-1}r \\ &= (a^{-1}r)^{-1}X'y(a^{-1}r) \end{aligned}$$

So, $\alpha\gamma \sim X'y$.
Above equation is alway successful. So, this shows the completeness of the proposed scheme.

Soundness. First, we define the adversary, $\mathcal{A}$. $\mathcal{A}$ works as follows:

1. $\mathcal{A}$ runs the protocol for several times as verifier. This means that $(\mathcal{P}, \mathcal{A})$ works. $\mathcal{A}$ takes the data from the $(\mathcal{P}, \mathcal{A})$ in his memory.
2. $\mathcal{A}$ runs the protocol for several times as prover. This means that $(\mathcal{A}, \mathcal{V})$ works. In this stage, $\mathcal{A}$ tries to impersonate the prover.

If the success probability of $\mathcal{A}$ is negligible, we can obtain the soundness of our proposed scheme.

After the stage 1, $\mathcal{A}$ gets the data $D_1, D_2, \ldots, D_k$.

$$(D_i = \{X_i, X'_i, r_i, \alpha_i, y_i, \beta_i\gamma_i\})$$

On stage 2, $\mathcal{A}$ sends X_t, X'_t ($X_t \in D_t$ $(1 \leq t \leq k)$) to verifier and gets a random challenge r from verifier. For impersonating the prover, $\mathcal{A}$ must compute the value $\gamma = r^{-1}ah(X_t \| r^{-1}X_t r)a^{-1}r$ without knowing the secret value a. Because it is impossible that find other solution which satisfies $\beta \sim \gamma$, $\gamma \sim y$, $\alpha\gamma \sim X'y$. From the infeasibility of k-SCSP, the success probability is negligible. This means that it is infeasible to get a from any number of pairs $(r_i\gamma_i r_i^{-1}, y_i) = (ay_i a^{-1}, y_i)$. Therefore there is no dishonest prover who can impersonate with non-negligible probability.

This completes the theorem. ■

4.3 Analysis of Scheme II

In this section, we show that the impersonator $\mathcal{I}$ of proposed scheme has negligible advantage under the DDH assumption.

In scheme II, the *completeness* that honest prover is always accepted by verifier with probability 1. Now we prove the *soundness* when adversary is passive.

Theorem 3 *The proposed scheme is secure against impersonation attack with maximum up to k-times view under the* DDH *assumption.*

Proof. It is clear that our proposed scheme is impersonated when $\mathcal{I}$ can distinguish μ_0 and μ_1. Now we shall define a sequence of "indistingushable" modified games $\mathbf{G_0}, \mathbf{G_1}, \mathbf{G_2},$ and $\mathbf{G_3}$ where $\mathbf{G_0}$ is original game and the last game clearly gives no advantages to the impersonator.

Game $\mathbf{G_0}$. In game $\mathbf{G_0}$, the impersonator receives the public information $\mathsf{Pub} = \langle g_1, g_2, D_0, \ldots, D_k \rangle$ and eavesdrop for a maximum of k-times and get all information of the interactions. Then, she receives a target commitment (x, μ_0, μ_1) and challenge (u_1, u_2, c). At this point, $\mathcal{I}$ outputs her guess μ_{i^*}. Let T_0 be the event that $\mu_{i^*} = \mu_i$ in game $\mathbf{G_0}$.

Game $\mathbf{G_1}$. Game $\mathbf{G_1}$ is identical to game $\mathbf{G_0}$, except for a small modification to the verifier's challenge-choosing oracle. In game $\mathbf{G_1}$, steps (4) and (5) are replaced with the following single step:

$$(3.5)'. \quad D = u_1^{p_1(x)} u_2^{p_2(x)}$$

It is clear that step (5)' computes the same value as step (5). The point of this change is to make explicit any functional dependency of the above quantity on u_1 and u_2. Let T_1 be the event that $\mu_{i^*} = \mu_i$ in game $\mathbf{G_1}$. Clearly, it holds that $\mathbf{Pr}[T_1] = \mathbf{Pr}[T_0]$.

Game $\mathbf{G_2}$. To turn game $\mathbf{G_1}$ into game $\mathbf{G_2}$, we make another change to the verifier's challenge-choosing oracle. We replace steps (1) and (3) with the following, respectively:

$$\begin{array}{ll} (3.1)'. & r_1 \leftarrow_R \mathbb{Z}_q, \quad r_2 \leftarrow_R \mathbb{Z}_q / r_1 \\ (3.3)'. & u_2 = g_2^{r_2} \end{array}$$

Let T_2 be the event that $\mu_{i^*} = \mu_i$ in game $\mathbf{G_2}$. Notice that while in $\mathbf{G_1}$ the values u_1 and u_2 are obtained using same value r_1, in game $\mathbf{G_2}$ they are independent subject to $r_1 \neq r_2$. Therefore, using a standard reduction argument, any non-negligible difference in behavior between $\mathbf{G_1}$ and $\mathbf{G_2}$ can be used to construct a PPT algorithm $\mathcal{A}$ that can distinguish Diffie-Hellman tuples from totally random tuples with non-negligible advantage. Hence $|\mathbf{Pr}[T_2] - \mathbf{Pr}[T_1]| \leq \epsilon$ for some negligible ϵ.

Game $\mathbf{G_3}$. In this game, we again modify the verifier's challenge-choosing oracle as follows:

$$(3.7)'. \quad e \leftarrow_R \mathbb{Z}_q, \quad c \leftarrow g_1^e$$

Let T_3 be the event that $\mu_{i^*} = \mu_i$ in game $\mathbf{G_3}$. Due to this last change, the challenge no longer contains μ_i, nor does any other information in the impersonator's view; therefore, we have that $\mathbf{Pr}[T_3] = \frac{1}{2}$. Moreover, we can prove that the impersonator has same chances to guess μ_i in both game $\mathbf{G_2}$ and $\mathbf{G_3}$, *i.e.* $\mathbf{Pr}[T_3] = \mathbf{Pr}[T_2]$.

Finally, combining all the intermediate results, we can conclude that impersonator $\mathcal{I}$'s advantage is negligible, more precisely less than ϵ. ■

Thus, for sufficiently large k, adversary can impersonate prover's identity with negligible probability ($\frac{1}{2^k}$).

4.4 Comparison

In this section, we compare our proposed schemes with the prior schemes in terms of their security,ID-based variant and 1-round running time of prover and verifier. Table 4.1 shows the comparison of identification schemes.

We can compute the 1-round running time from [32, 30, 28, 35]. The modular multiplication speed on Pentium 3 866MHz is 0.115 ms in [28]. The braid implementation result is from [30]. In [32, 37], they give the numbers of modular multiplications and point additions of previous identification schemes and we can estimate that A costs less than or equal to two times M, i.e., $\mathsf{A} \leq 2\mathsf{M}$. KK scheme takes $140\mathsf{A}+2\mathsf{M}$ for prover's processing and $141\mathsf{M}$ for verifier's. So, estimating time for prover and verifier's processing time is 32.44 and 16.22, respectively. Other schemes can be derived from the comparison table in [32]. In braid identification scheme we can estimate the processing time from [30]. Our identification scheme in Table 4.1 takes 28 braids.

Table 4.1: Comparison of identification schemes

Comparison		KK	Schnorr	Okamoto	FFS	GQ	Scheme I	Scheme II
Security proof		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Public Key Size (bits)		512	512	512	10,240	1,024	591	512
Security against active attack		Yes	No	Yes	Yes	No	Yes	No
Cryptographic problem		BDH	DLP	DLP	IFP	IFP	MTSP	DDH
1-Round running time	(Prover)	32.44	24.17	28.18	1.26	7.02	25.77	7.25
(ms)	(Verifier)	16.22	24.17	28.53	1.26	3.94	36.08	11.62
Extention to ID-based system		Possible	Possible	Possible	Possible	Possible	Impossible	Possible

So the public key size is 591 bit. For prover's processing is same as braid signature's signing algorithm. So, it takes 25.77 ms. And verifier have 2 conjugacy decision than signature verifying of [30]. Conjugacy decision algorithm takes 5.154 ms. So, verifier's processing time takes $36.08 (= 25.77 + 10.31)$

ms. Finally we estimate DDH identification scheme. Scheme II takes $2\mathsf{E} + k + 1\mathsf{M}$ for prover and $3\mathsf{E}$+$3\mathsf{M}$ for verifier. From [35, 32], we know that E is costs less than or equal to 30 times M. So, scheme II takes processing time of prover and verifier as 7.25 and 11.62, respectively. The processing time of verifying stage is depend on the security parameter k. In this comparison we define the k as 40.

Chapter 5
Conclusion

In this thesis, we have studied the design and analysis of secure identification schemes against passive adversaries. We have reviewed previous works and presented current concerns on the identification schemes. And then we have presented our suggestions to solve the problems.

We have presented a construction of a new identification scheme based on the conjugacy problem on the braid group. The identification scheme is typical three-round (canonical) identification. In the open literature, there is no identification scheme based on conjugacy problem over a braid group. For the constructing scheme I, we transferred Ko *et al.*'s braid signature scheme to Δ-times zero-knowledge protocol. We have settled the the proposed model of our apporach. Then we prove that our identification scheme is secure against passive impersonation attacks. The proof is from that the hardness of MTSP is same as our proposed scheme. In other words, we have showed that any attacker that can break proposed identification scheme can be transformed into an efficient algorithm to solve the underlying problem, matching triple search problem (MTSP). And this scheme is the first identification scheme based on conjugacy problem over a braid group. We hope that our scheme can open the new genre of braid cryptosystem.

And we have presented another construction of a new identification scheme based DDH problem. The identification scheme is also typical three-round (canonical) identification. We prove that our identification scheme is secure against impersonation attacks when adversary has some limitation of view.

The proof gives that impersonator has negligible advantage for successful cheating. The most important point of scheme II is the first identification scheme which is based on DDH problem and proved the security against impersonation in standard model. In addition, our proposed scheme can be transferred to ID-based version by changing public information g_1 as $h_1(\mathsf{ID})$ and g_1 as $h_2(\mathsf{ID})$, h_1 and h_2 are collision-resilient one-way hash functions which are from binary string to a group G.

Our future works is as follows: (1)We modify our schemes secure against active attacks and efficient. (2)We upgrade the scheme II is secure without adversary's limitation.

공액문제와 **DDH** 문제에 기반한 안전성이 증명가능한 개인식별기법

김진

빌딩의 입구나 공항의 출입국관리 게이트에서 상대의 신원을 검증하기 위해서 운전면허증과 패스포트 등의 신분증명서를 제시하여 증명을 한다. 그러나, 이것들은 비교적 위조가 간단하다는 결점을 갖는다. 그래서 이후에는 위조가 어려운 IC카드 등의 전자매체를 이용한 인증시스템이 보급될 것이다. 또한 네트워크를 통한 각종 서비스에서 과금의 대상이 되는 이용자의 정당성을 확인하는 개인식별은 필수적이다. 식별은 서로 통신하는 송,수신자의 한쪽이 다른 한쪽의 신원을 어떤 프로토콜을 사용하여 확인하는 방법 또는 과정이며, 이때 사용되는 기법을 식별 프로토콜이라 한다.

식별 프로토콜은 기본적으로 증명자와 검증자로 구성된다. 증명자가 검증자에게 자신의 신원을 확인시키는 동시에 공격자가 자신의 신원을 위장하는 것을 어렵게 하는 것이 식별 프로토콜의 목표이다. 검증자의 입장에서 볼 때, 식별 프로토콜의 결과는 증명자가 주장하는 신원을 올바른 것으로 수락하거나 거절하는 것이다. 구체적으로 식별기법이 추구하는 바는 다음과 같다.

1. 증명자와 검증자가 모두 적법한 사용자일 경우에 증명자는 검증자에게 자신의 신원을 확인시킬 수 있어야 한다. 즉, 검증자는 식별프로토콜의 실행결과로 증명자의 신원이 올바른 것으로 수락한다.

2. 정보양도에 대한 안전성 : 검증자가 증명자의 신원을 확인하는 과정에 사용된 정보를 이용하여 제 3의 사용자에게 검증자가 증명자의 신원을 위장할 수 없어야 한다.

3. 위장에 대한 안전성 : 증명자가 아닌 공격자가 검증자에게 자신이 증명자인 것으로 위장할 수 있는 확률은 무시할 수 있어야 한다. 무시할 수 있다는 의

미는 위장할 수 있는 확률이 너무 작아서 실질적인위협 요소가 되지 못한다는 것이다.

4. 정보 양도 및 위장에 대하여 다음의 상황을 가정하여도 안전하여야 한다.
   - 증명자와 검증자간의 이전의 식별 과정은 관찰 가능하다.
   - 공격자는 증명자 또는 검증자로서 이전의 프로토콜 실행에 참여하였다.
   - 공격자의 주도하에 식별 프로토콜이 동시에 여러번 수행된다.

영지식 식별 프로토콜은 프로토콜의 실행 과정에서 공격자가 위장하는데 도움이 되는 어떠한 부분 정보도 나타내지 않는 것을 아이디어로 한다.

식별의 가장 주된 응용 분야는 자원에 대한 접근제어이다. 실례로 특정 개체마다 접근 권한이 주어진 환경으로, 컴퓨터 계정에 대한 지역 또는 원거리 접근, 현금 자동 지급기에서의 현금 인출, 특정 통신 포트를 사용할 수 있는 통신 권한, 소프트웨어에 대한 접근 등이 있다. 대부분의 응용 환경에서 식별의 역할은 신원이 확인된 개체에게만 자원의 사용을 허가하겠다는 것이다.

본 논문은 우선 증명가능한 안전성을 지니는 개인식별기법에 관한 요구사항을 논한다. 식별기법에 대한 다양한 공격방법과 영지식대화증명을 통한 식별기법의 안전성을 논한 뒤, 수동적 공격자의 신원위조 공격에 안전한 개신식별기법으로 비가환군의 일종인 땋임군에 기반한 새로운 개인식별기법과 결정적 디피헬만 가정에 기반한 개인식별기법을 제안한다. 이중 땋임군에서의 개인 식별기법은 군이 갖는 비가환 특성때문에 기존의 안전성 분석기법이 적용되기 어려운 이유로 그 의미를 찾을 수 있다. 설계한 제안 기법의 안전성은 땋임군에서의 k연립 공액찾기 문제와 매칭 세쌍 찾기 문제의 어려움에 기반하고 있으며, 이 안전성을 지식의 영지식 증명을 통해서 증명한다. 본 제안 기법은 땋임군 중 공액찾기 문제는 해결 가능성 없는 문제이고, 공액 결정 문제는 해결 가능성 있는 문제를 제공하는 비가환군에서 정의되었다.

결정적 디피헬만 가정에 기반한 기법은 공격자의 능력에 제한을 두고 그 안전성을 증명하였다. 본 기법은 최대 k회의 인터렉션만을 허용한다. 이 기법은 우선 랜덤오라클 모델이 아닌 스탠더드 모델에서의 안전성을 제공한다는 장점을 지닌다. 제안 기법에서 위장공격을 하고자 하는 공격자가 검증자로 부터 신원을 수락

받는 어드밴티지는 무시해도 좋을(negligible)한 양으로 측정된다. 적당한 회수의 라운드를 반복해서 옳바른 증명자의 신원을 확인할 수 있다.

References

1. I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, "New key agreement schemes in braid group cryptography," *RSA2001*, 2001

2. E. Artin, "Theory of Braids," *Ann. of Math.* 48 pp. 101–126, 1947

3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes", *Advances in Cryptology – Crypto 1998*, LNCS 1462, Springer-Verlag, pp. 26–45, 1998.

4. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols", *ACM Conference on Computer and Communications Security*, pp. 62–73, 1993.

5. J. S. Birman, K. H. Ko and S. J. Lee, "The infimum, supremum and geodesic length of a braid conjugacy class," to appear in Advances in Mathematics.

6. M. Bellare and P. Rogaway, "Optimal asymmetric encryption – How to encrypt with RSA", *Advances in Cryptology – Crypto 1994*, LNCS 950, Springer-Verlag, pp. 92–111, 1994.

7. M. Bellare and S. Miner, "A forward-secure digital signature scheme," *Advances in Cryptology – CRYPTO '99*, LNCS 1666, Springer-Verlag, pp.( ), 1999.

8. M. Bellare and A. Palacio, "GQ and Schnorr identification schemes ; proofs of security against impersonation under active and concurrent attacks," *Advances in Cryptology – CRYPTO 2002*, LNCS 2442, Springer-Verlag, pp. 162–177, 2002.

9. E. F. Brickell and K. S. McCurley, "An interactive identification scheme based on discrete logarithms and factoring", *Jounal of Cryptology* 5: 29–39, 1992.

10. R. Canetti, "Towards realizing random oracles: Hash functions that hide all partial information", *Advances in Cryptology – Crypto 1997*, LNCS 1295, Springer-Verlag, pp. 455–469, 1997.

11. R. Canetti, O. Goldreich, and S. Halevi, "The ramdom oracle methodology, revisited", *ACM Symposium on the Theory of Computing*, ACM Press, pp. 209–218, 1998.

12. R. Canetti, D. Micciancio, and O. Reingold, "Perfectly one-way probabilistic hash functions", *ACM Symposium on the Theory of Computing*, pp. 131–140, 1998.

13. J.-S. Coron, "On the security of full domain hash", *Advances in Cryptology – Crypto 2000*, LNCS 1880, Springer-Verlag, pp. 229–235, 2000.

14. R. Cramer and V. Shoup, "A Practical Public Key Cryptosystem Provably Secure agaisnt Adaptive Chosen Ciphertext Attack", *Advances in Cryptology – Crypto 1998*, LNCS 1462, Springer-Verlag, pp. 13–25, 1998.

15. R. Cramer and V. Shoup, "Signature schemes based on the strong RSA assumption," *In 5th ACM Conference on Computer and Communications Security*, pp. 46–51, Singapore, Nov. ,ACM Press, 1999.

16. W.Diffie and M.E.Hellman, "New directions in cryptography," *IEEE trans, Inform. Theory*, IT - 22:644–654, Nov 1976.

17. N. Franco and J. Gonzales-Meneses, "Conjugacy problem for braid groups and Garside groups," `http://xxx.lanl.gov/abs/math.GT/0112310`, 2001

18. U. Feige, A. Fiat, and A. Shamir, “Zero-knowledge proofs of identity”, *Journal of Cryptology* 1: 77–94, 1988.

19. A. Fiat and A. Shamir, “How to prove yourself: pratical solutions to identification and signature problems”, *Advances in Cryptology – Crypto 1986*, LNCS 263, Springer-Verlag, pp. 186–194, 1987.

20. U. Feige and A. Shamir, “Witness Indistinguishavle and Witness Hiding Protocols,” Proceedings of STOC, pp. 416–426, 1990.

21. M. Girault, “An identity-based identification scheme based on discrete logarithms modulo a composite number”, *Advances in Cryptology – Eurocrypt 1990*, LNCS 473, Springer-Verlag, pp. 481–486, 1991.

22. O. Goldreich, *Foundation of Cryptography–Fragments of a Book*, available from `http://theory.lcs.mit.edu/~oded/` (1995).

23. O. Goldreich, *Intoduction to Complexity Theory*, available from `http://www.wisdom.weizmann.ac.il/~oded/` (1999).

24. S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems”, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC'85)*, Providence Rhode Island, pp. 291–304, May 1985.

25. O. Goldreich and H. Krawczyk, “On the composition of zero-knowledge proof systems”, *Proc. of the 17th ICALP*, LNCS 443, Springer-Verlag, pp. 268–282, 1990.

26. S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems”, *SIAM J. Comput.*, 18: 186–208, 1989.

27. L. Guillou and J. Quisquater, “A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory”,

Advances in Cryptology – Eurocrypt 1988, LNCS 330, Springer-Verlag, pp. 123–128, 1989.

28. Cetin Kaya Koc, Tolga Acar and S. Kaliski Jr. “Analyzing and Comparing Montgomery Multiplication Algorithms,” *IEEE Micro*, 16(3):26–33, 1996.

29. K. Ko, S. Lee, J. Cheon, J. Han, J. Kang, C. Park, “New Pulic-key Cryptosystem using Braid Groups,” *Advances in Cryptology – Crypto 2000*, LNCS 1880, Springer-Verlag, pp. 166–183, 2000.

30. K.H. Ko, D.H. Choi, M.S. Cho, and J.W. Lee, “ New signature scheme using conjugacy problem,” *Preprint; http://eprint.iacr.org/2002/168*, 2002.

31. K. Kurosawa and S.-H. Heng, “From Digital Signature to ID-Based Identification/Signature,” *To appear in PKC 2004*, 2003.

32. K. Kim and K. Kim, “A new identification scheme based on bilinear Diffie-Hellman problem,” *7th Australasian Conference on Information Security and Privacy – ACISP ’02*, LNCS vol.2384 , Springer-Verlag, pp. 362–378, 2002.

33. M. Luby, “Pseudorandomness and Cryptographic Applications”, Princeton University Press, 1996.

34. E.K. Lee, S.J. Lee, and S.G. Hahn, “Pseudorandomness from braid groups,” Crypto2001, 2001

35. A. J. Manezes, P. C.van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.

36. K. Ohta and T. Okamoto, “A modification of the Fiat-Shamir scheme”, *Advances in Cryptology – Crypto 1988*, LNCS 403, Springer-Verlag, pp. 232–243, 1990.

37. T. Okamoto, “Provably secure and practical identification schemes and corresponding signature schemes”, *Advances in Cryptology – Crypto 1992*, LNCS 740, Springer-Verlag, pp. 31–53, 1993.

38. H. Ong and C.P. Scnorr, “Fast signature generation with a Fiat Shamir-like scheme”, *Advances in Cryptology – Eurocrypt 1990*, LNCS 473, Springer-Verlag, pp. 432–440, 1991.

39. D. Pointcheval, “A new identification scheme based on the perceptrons problem”, *Advances in Cryptology – Eurocrypt 1995*, LNCS 921, Springer-Verlag, pp. 319–328, 1995.

40. C. Popescu, “An identification scheme based on the elliptic curve discrete logarithm problem”, *IEEE High Performance Computing in the Asia-Pacific Region*, Volume: 2, pp. 624–625, 2000.

41. R. L. Rivest, A. Shamir, and L. M. Adleman, ”A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, 21(2):120–126, Feb. 1978.

42. A.D. Santis, S. Micali, and G. Persiano, “Non-interactive zero-knowledge proof systems”, *Advances in Cryptology – Crypto 1987*, LNCS 293, pp. 52–72, 1988.

43. C.P. Schnorr, “Security of 2^t-root identification and signatures”, *Advances in Cryptology – Crypto 1996*, LNCS 1109, Springer-Verlag, pp. 143–156, 1996.

44. V.M. Sidelnikov, M.A. Cherepnev, V.Y. Yashcenko, “Systems of open distribution of keys on the basis of noncommutative semigroups,” Russian Acad. Sci. Dokl. Math. 48-2 (1194) pp. 384–386, 1993.

45. V. Shoup, “Why chosen ciphertext security matters”, IBM Research Report RZ3076(#93122), 1998.

46. V. Shoup, “On the security of a practical identification scheme”, *Journal of Cryptology* 12: 247–260, 1999.

47. J. Stern, “A new identification scheme based on syndrome decoding”, *Advances in Cryptology – Crypto 1993*, LNCS 773, Springer-Verlag, pp. 13–21, 1994.

48. J. Stern, “Designing identification schemes with keys of short size”, *Advances in Cryptology – Crypto 1994*, LNCS 839, Springer-Verlag, pp. 164-173, 1994.

49. D.R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.

Acknowledgements

First, I would like to express my gratitude to Prof. Kwangjo Kim, my thesis advisor, for his constant direction and support. Without his guidance, I could never have carried out my research. Special thanks are also due to Prof. Jae Choon Cha and Dr. Dae Sung Kwon for their generosity and agreeing to serve as advisory committee members.

I would like to mention the close relationship of my colleagues in C&IS lab members. Their kind and sincere support lead me to carry out this thesis and successful ICU life.

Most of all, I have to give my thanks and love to my parents for their endless encourage and devotion. I dedicate this work to them.

Curriculum Vitae

Name : Zeen Kim

Date of Birth : Aug. 30. 1977

Sex : Male

Nationality : Korean

Education

2001.2 Mathematics
Korea Advanced Institute of Science and Technology (B.S.)

2004.2 Computer Science
Information and Communications University (M.S.)

Projects

2001.6–2001.11 Graduate Research Assistant
A Study on the Policy for Information Warfare
NSRI

2001.6–2003.2 Graduate Research Assistant
Development of IT Manpower
The Ministry of Information and Communications

2001.6–2001.11 Graduate Research Assistant
Development of IT Manpower
The Ministry of Information and Communications

2001.6–2001.11 Graduate Research Assistant
A Study on Cryptanalysis of the Block Cipher Candidates for AES, NESSIE and CRYPTREC
KISA

2002.4–2002.11 Graduate Research Assistant
Development of secure electronic trading system for online game items
Semtlo Inc.

2002.6–2002.11 Graduate Research Assistant
A Study on the Trend of Informaton Warfare Technology
NSRI

2002.6–2002.11 Graduate Research Assistant
Research on ID-based PKI and its Application
ETRI

2003.3–2004.2 Graduate Research Assistant
Support for running the International Research center for Information Security
The Ministry of Information and Communications

Career

2002.1–2002.12 Graduate Research Assistant
Development of Gifted on IT field
Education Center for IT Gifted, ICU

2002.8–2003.2 Apprentice Researcher
Network Security Structure Research Team,
Information Security Research Division,
Electronics and Telecommunications Research Institute(ETRI)

2003 Spring Graduate Teaching Assistant
ICE0100 University Mathematics

2003 Fall Graduate Teaching Assistant
ICE0102 Linear Algebra

Publications

(1) 2002.8 A Study on Discrete Logarithm Related Problems, CISC 2002 Summer, (with J. Cheon)

(2) 2003.8 A Study on Next Generation Key Management Protocol for IPsec, CISC 2003 Summer, (with S. Lee, C. Choi, and K. Kim)

(3) 2003.8 Tree-based Authenticated Group Key Agreement Protocol, CISC 2003 Summer, (with S. Lee and K. Kim)

(4) 2003.10 Schnorr Signature Scheme with Restricted Signing Capability, CCS 2003, (with C. Choi and K. Kim)

(5) 2003.12 A New Identification Scheme Based on Conjugacy Problem, CISC 2003 Winter, (with K. Kim)

(6) 2004.1 Provably Secure Identification Scheme based on Braid Group, SCIS 2004, (with K. Kim)