A Thesis for the Degree of Master of Science

Threshold Password-Authenticated Key Retrieval Protocol Using Bilinear Pairings

Songwon Lee

School of Engineering

Information and Communications University

2004

Threshold Password-Authenticated Key Retrieval Protocol Using Bilinear Pairings

Threshold Password-Authenticated Key Retrieval Protocol Using Bilinear Pairings

Advisor : Professor Kwangjo Kim

by

Songwon Lee
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
Dec. 26. 2003
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

Threshold Password-Authenticated Key Retrieval Protocol Using Bilinear Pairings

Songwon Lee

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

Dec. 26. 2003

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae Choon Cha, Assistant Professor
School of Engineering

Committee Member
Dong Hoon Lee, Ph.D
National Security Research Institute

M.S. Songwon Lee
20022099

Threshold Password-Authenticated Key Retrieval Protocol Using Bilinear Pairings

School of Engineering, 2004, 41p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

These days many real world systems still rely on password authentication to verify the user's identity before allowing that user to be given certain network services. So far many protocols based on password have been proposed, even if there are many security concerns associated with password-based protocol. Thus many researchers have studied how to make the password-like weak key to be more stronger one.

On the other hand, with rapid development on the Internet, users can easily access the network anywhere and anytime, which is so called ubiquitous computing paradigm. Thus, we want to give users password-authenticated access to private keys from anywhere, while protecting the password and those private keys from being revealed to an attacker.

We present a new threshold password-authenticated key retrieval protocol that allows a *roaming user*, who accesses a network from different client terminals, to download a private key from remote servers with knowledge of only his identity and password information, assuming that the user does not carry the smart card storing user's private information.

We note that as a goal of a multi-server roaming system, the protocol has to allow a user to get his private key from the servers, even if some of the

servers are compromised. With this point of view, we give the first description of a *threshold password-only roaming protocol using bilinear pairings*. In this paper, we use (k,n)-*threshold scheme* in which only k honest servers or more are engaged to reconstruct a secret key. Our scheme is based on bilinear pairings which could be built from Weil pairing or Tate pairing.

Contents

List of Tables

List of Figures

List of Abbreviations

CDH Computational Diffie-Hellman

DH Diffie-Hellman

DDH Decisional Diffie-Hellman

DLP Discrete Logarithm Problem

GDH Gap Diffie-Hellman

IFP Integer Factorization Problem

IBE Identity-Based Encryption

ID Identity

MOV Menezes-Okamoto-Vanstone attack

OO Ohta-Okamoto

PKA Password-authenticated Key Agreement

PKR Password-authenticated Key Retrieval

PPT Probabilistic Polynomial Time

ROM Random Oracle Model

TTP Trusted Third Party

TPKR Threshold Password-authenticated Key Retrieval

$\mathcal{TPS}$ Threshold Password-authenticated key Sharing

$\mathcal{TPR}$ Threshold Password-authenticated key Retrieval

VSS Verifiable Secret Sharing

List of Notations

C a client terminal

$D_K(m)$ message m decrypted with symmetric key K

$\hat{e}$ a bilinear pairing

$E_K(m)$ message m encrypted with symmetric key K

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

$\mathbb{G}$ a cyclic group of a prime order

H_1, H_2, H_3 cryptographic hash functions

$\mathcal{IG}$ a parameter generator

k a security parameter

K_m a symmetric master key

L_i the i-th player

$\mathcal{O}$ point at infinity (on an elliptic curve)

π a password

P a generator of group $\mathbb{G}$

$\mathcal{P}$ a protocol

q the order of $\mathbb{G}$

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

SK a private key

U a user

$x \xleftarrow{R} \mathcal{S}$ an element x randomly

$\mathbb{Z}$ integers

$\mathbb{Z}_q$ integers modulo q

$\mathbb{Z}_q^*$ a group under multiplication modulo q

Chapter 1
Introduction

1.1 Roaming Protocol

With rapid development on the Internet users can easily access the network to securely retrieve private data and digitally sign critical transactions, such as stock trade or e-banking, from different client terminals – without being bound to a single terminal on which the user's electronic credentials reside. Credentials may consist of public/private key pairs, public key certificates or other private user data. Those are vulnerable to various attacks where the private key may be stolen or substituted, usually without user's even being aware of it. Furthermore, for *roaming users* who access a network from different client terminals, the terminal cannot store such user-specific data. So far, there have been two basic approaches to provide a secure roaming service – *portable hardware key storage such as smartcards* and *password-based mechanisms* [16, 32].

While the smartcard plays an important role in storing sensitive information, it is not currently practical in many real environments due mainly to inconvenience, for example, a user needs an external interface device to communicate with a smart card. Given the cost and availability problems of hardware storage devices, more reasonable approach is to use the password-based mechanisms. In this approach, a roaming users store their credentials at a central server and download temporary copies when needed to their local machine.

The fundamental problems with passwords come from the fact that most users' passwords are drawn from a relatively small spaces and are easily memorable, which also means that the password may be easily guessed by an attacker. With mainly focused on this point, strong password-authenticated roaming protocols were presented by Perlman *et al.*[45], Ford *et al.*[16], and Jablon[27], *etc.* We refer to *roaming protocol* as a secure password-based protocol for remote retrieval of a private key from one or more credentials servers [27]. Using just an easily memorized password, and no other stored user credentials, the user authenticates to a *credentials server* and retrieves his private key for temporary use on any acceptable client machine.

1.2 Our Contributions

In the last few years, several roaming schemes have been proposed. Some of them used multiple servers to implement a roaming protocol that uses a weak secret key, user's password, to securely retrieve and reconstruct a strong private key that has been divided into pieces distributed among multiple servers. We note that as one of goals, a protocol has to allow a user to get his private key from the servers, even if some of the servers are compromised. With this point of view, we give the description of a threshold password-only roaming protocol.

In this paper we present a threshold password-authenticated key retrieval protocol for a roaming user. We make use of the (k,n)-*threshold scheme* in which only k honest servers or more are engaged to reconstruct a secret key. Our scheme is based on bilinear pairings that could be built from Weil pairing or Tate pairing over *Gap Diffie-Hellman(GDH)* group, which *Computational Diffie-Hellman(CDH)* problem is hard but *Decision Diffie-Hellman(DDH)* problem is easy. We also prove security of our construction in a formal way.

To the best of our knowledge, our proposed scheme is the first *threshold password-only roaming protocol using bilinear pairings.*

1.3 Outline of the thesis

The remainder of the thesis is organized as follows: In Chapter 2, we describe some underlying concepts on the bilinear parings and the cryptographic primitives used in our proposed scheme, and review several password-based protocols. Chapter 3 presents our proposed threshold roaming protocol along with not only cryptographic notions but also the security model, and Chapter 4 discusses its security, and evaluates performance of our scheme and compares with other scheme as well. Finally, we end with concluding remarks and suggestions for further work in Chapter 5.

Chapter 2
Preliminaries

2.1 Cryptographic Background

2.1.1 One-Way Functions

A one-way function is one of basic cryptographic primitives, is informally a function which is *easy* to compute but *hard* to invert. Any probabilistic polynomial time (PPT) algorithm attempting to invert the one-way function on an element in its range, will succeed with no more than *negligible* probability, where the probability is taken over the element in the domain of the function and the coin tosses of the PPT attempting the inversion [17].

Definition 2.1 *We call function* $\epsilon(k)$ negligible *if for every polynomial p(k), there exists* k_0 *such that* $\epsilon(k) < \frac{1}{p(k)}$ *for all* $k \geq k_0$.

The above definition considers the success probability of an algorithm to be *negligible* if as a function of the input length the success probability is bounded by any polynomial fraction. On the other hand, we say that a function ϵ is *non-negligible* if there exists a polynomial p such that for sufficiently large k it holds that $\epsilon(k) \geq \frac{1}{p(k)}$.

Definition 2.2 *A function* $f : \{0,1\}^* \rightarrow \{0,1\}^*$ *is* (strong) one-way *if:*

(1) there exists a PPT that on input x output f(x);

(2) for every PPT algorithm $\mathcal{A}$ there is a negligible function $\epsilon_{\mathcal{A}}$ such that for sufficiently large k,

$$\Pr\left[f(z) = y : x \stackrel{R}{\leftarrow} \{0,1\}^k; y \leftarrow f(x); z \leftarrow \mathcal{A}(1^k, y)\right] \leq \epsilon_{\mathcal{A}}(k).$$

Definition 2.3 *A function $f : \{0,1\}^* \to \{0,1\}^*$ is* weak one-way *if:*

(1) there exists a PPT that on input x output $f(x)$;

(2) there is a polynomial functions Q such that for every PPT algorithm $\mathcal{A}$, and for sufficiently large k,

$$\Pr\left[f(z) \neq y : x \stackrel{R}{\leftarrow} \{0,1\}^k; y \leftarrow f(x); z \leftarrow \mathcal{A}(1^k, y)\right] \geq \frac{1}{Q(k)}.$$

The difference between the two definitions is that whereas we only require some non-negligible fraction of the inputs on which it is hard to invert a weak one-way function, a strong one-way function must be hard to invert on all but a negligible fraction of the inputs.

2.1.2 Security Models

Standard Model and Random Oracle Model. There are two common formal methods which are used to prove security of cryptographic schemes. One is to use the standard model and the other is to use the random oracle model.

The Standard Model. This is the preferred approach of modern, mathematical cryptography. Here, one shows with mathematical rigor that any attacker who can break the cryptosystem can be transformed into an efficient algorithm to solve the underlying well-known problem that is widely believed to be very hard. Then, to show security of a cryptographic scheme turns out to verify that if there exists an attacker who can successfully attack the scheme. One can then construct an attacker who can break the presumed hardness.

The Random Oracle Model. The notion of a *random oracle model* (ROM) was introduced by Bellare and Rogaway [9]. The result of this approach is a reductionist proof, however the proof is only valid in a "parallel universe" where a "magic hash functions" exist—they *do not* exist in the "real world" of computation. We stress that the existence of magic hash functions is not a "hardness assumption," like IFP and DLP; they simply do not exist. Rather, they are a rough-and-ready *heuristic*, much like assuming the earth is flat, and that there is no wind resistance.

To analyze a protocol using ROM one replaces a real-world cryptographic hash function by a *black-box* that when queried outputs a *random bit string*, subjects to the restriction that it always outputs the same value on the same input. Having made this replacement, one then gives a reductionist security argument. The right way to view a proof of security in ROM is as a proof of security against a restricted class of adversaries that do not care if the hash function really is a black box.

2.1.3 Bilinear Pairings

We describe some basic concepts of bilinear maps and the relevant problems, with help of the Boneh-Franklin's work [2] (refer to the full version for details) which suggested Identity-Based Encryption from the Weil Pairing.

Let us consider an additive group $\mathbb{G}_1$ and a multiplicative group $\mathbb{G}_2$ of the same order q. Assume that the discrete logarithm problem is hard in both groups. Let P be a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ a bilinear map satisfying the following properties:

1. *Bilinear*: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$.
2. *Non-degenerate*: If $\hat{e}(P, Q) = 1$ for all $Q \in \mathbb{G}_1$, then $P = \mathcal{O}$. In other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$.

3. *Computable*: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

A bilinear map satisfying the three properties above is said to be an *admissible* bilinear map. To construct the bilinear pairing, we can use the Weil pairing and Tate pairing. $\mathbb{G}_1$ is a cyclic subgroup of the additive group of points of an elliptic curve $E/\mathbb{F}_p$ over a finite field while $\mathbb{G}_2$ is a cyclic subgroup of the multiplicative group associated to a finite field $\mathbb{F}^*_{p^2}$.

As mentioned in [2], the existence of the bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ as above has two direct implications to these groups.

The MOV reduction: Menezes, Okamoto, and Vanstone [39] showed that the discrete log problem in $\mathbb{G}_1$ is no harder than the discrete log problem in $\mathbb{G}_2$. To see this, let P and $Q \in \mathbb{G}_2$ be an instance of the discrete log problem in $\mathbb{G}_1$ where both P and Q have order q. We wish to find an $\alpha \in \mathbb{Z}_q$ such that $Q = \alpha P$. Let $g = \hat{e}(P, P)$ and $h = \hat{e}(Q, P)$. Then, by bilinearity of $\hat{e}$ we know that $h = g^\alpha$. By non-degeneracy of $\hat{e}$ both g, h have order q in $\mathbb{G}_2$. Hence, we reduced DLP in $\mathbb{G}_1$ to DLP in $\mathbb{G}_2$. It follows that for discrete log to be hard in $\mathbb{G}_1$ we must choose our security parameter so that discrete log is hard in $\mathbb{G}_2$.

Decision Diffie-Hellman is easy: The DDH problem [8] in $\mathbb{G}_1$ is to distinguish between the distributions $\langle P, aP, bP, abP\rangle$ and $\langle P, aP, bP, cP\rangle$ where a, b, and c are random in $\mathbb{Z}^*_q$ and P is random in $\mathbb{G}^*_1$. Joux and Nguyen [29] pointed out that DDH in $\mathbb{G}_1$ is easy. To see this, observe that given $\{P, aP, bP, cP\} \in \mathbb{G}^*_1$ we have

$$c = ab \bmod m \quad \iff \quad \hat{e}(P, cP) = \hat{e}(aP, bP).$$

The CDH problem in $\mathbb{G}_1$ can still be hard. Joux and Nguyen [29] gave examples of mappings $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ where CDH in $\mathbb{G}_1$ is believed to be hard even though DDH in $\mathbb{G}_1$ is easy.

With such groups $\mathbb{G}_1$ and $\mathbb{G}_2$, we can define the following hard cryptographic problems:

Discrete Logarithm(DL) problem: Given P and $P' \in \mathbb{G}_1$, find an integer n such that $P' = nP$ whenever such integer exists.

Computational Diffie-Hellman(CDH) problem: Given a triple $(P, aP, bP) \in \mathbb{G}_1$ for a and $b \in \mathbb{Z}_q^*$, compute abP.

Decision Diffie-Hellman(DDH) problem: Given a quadruple $(P, aP, bP, cP) \in \mathbb{G}_1$ for a, b, and $c \in \mathbb{Z}_q^*$, decide whether $c = ab \pmod{q}$ or not.

Gap Diffie-Hellman(GDH) problem: A class of problems where the CDH problems are hard but DDH problems are easy. That is, given a triple $(P, aP, bP) \in \mathbb{G}_1$ for a and $b \in \mathbb{Z}_q^*$, find the element abP with the help of the DDH oracle (which answers whether a given quadruple $(P, aP, bP, cP) \in \mathbb{G}_1$ is a DH one or not).

We assume through the thesis that CDHP and DLP are intractable, which means there is no polynomial time algorithm to solve CDHP or DLP with non-negligible probability. When the DDHP is easy but the CDHP is hard on the group $\mathbb{G}(= \mathbb{G}_1)$, we call $\mathbb{G}$ a *Gap Diffie-Hellman (GDH) group.*

GDH Parameter Generator. A polynomial time probabilistic algorithm $\mathcal{IG}$ is called a *GDH parameter generator* if for a given positive integer k, which plays the role of a security parameter, it outputs (descriptions of) a cyclic group $\mathbb{G}$ of prime order and a polynomial time algorithm $\mathcal{D}$ which solves DDHP in $\mathbb{G}$. We will always view $\mathbb{G}$ as an additive group. We denote the output of $\mathcal{IG}$ by $\mathcal{IG}(1^k)$.

GDH Assumption. Let $\mathcal{IG}$ be a GDH parameter generator, and let $\mathcal{A}$ be an algorithm whose input consists of a group $\mathbb{G}$ of prime order q, an algorithm

$\mathcal{D}$ solving DDHP, a generator P of $\mathbb{G}$, aP and bP (a and $b \in \mathbb{Z}_q$) and whose output is an element of $\mathbb{G}$ that is expected to be abP. As usual, the advantage of $\mathcal{A}$ with respect to $\mathcal{IG}$ is defined to be

$$\Pr\left[\mathcal{A}(\mathbb{G}, \mathcal{D}, P, aP, bP) = abP \;\middle|\; \begin{array}{l} \langle \mathbb{G}, \mathcal{D} \rangle \leftarrow \mathcal{IG}(1^k), \\ P \stackrel{R}{\leftarrow} \mathbb{G}^*, \\ a, b \stackrel{R}{\leftarrow} \mathbb{Z}_q \end{array}\right].$$

$\mathcal{IG}$ is said to satisfy the *GDH assumption* if any polynomial time algorithm $\mathcal{A}$ has advantage $\leq 1/f(k)$ for all polynomial f, that is, no polynomial time algorithm can solve CDHP with non-negligible advantage [12].

2.1.4 Threshold Scheme

The concept of a threshold scheme, called secret sharing scheme, was introduced in [48], and since then many researchers have investigated such schemes and their applications, *e.g.*, [44], [19], [41], [11], [37], [46], and [51]. Two main goals that motivate this research area are: (1) provide security to applications that are inherently distributed, namely, several parties are trying to accomplish some common task in the presence of an attacker, and (2) avoid single points-of-failure in a security system by distributing the crucial security resources [13].

A (k,n)-*threshold secret sharing scheme* is a protocol among n players in which the *dealer* distributes partial information (*share*) about a *secret* to n participant such that:

- Any group of fewer k participants can not obtain any information about the secret.
- Any group of at least k participants can compute the secret in polynomial time.

In this thesis, we use the verifiable secret sharing (VSS) scheme due to Pedersen [44], and which we denote by Pedersen-VSS. We now describe this Pedersen-VSS protocol briefly.

The parameters p and q denote large primes such that q divides $p-1$, $\mathbb{G}_q$ is the unique subgroup $\mathbb{Z}_p^*$ of order q, and g is a generator of $\mathbb{G}_q$. Let h be element of $\mathbb{G}_q$ such that nobody knows $\log_g h$.

1. The dealer, D, distribute a secret $s \in \mathbb{Z}_q$ as follows:

   (a) D publishes a commitment to s: $E_0 = g^s h^t$ for a randomly chosen $t \in \mathbb{Z}_q$.

   (b) D chooses two random polynomials $F(x)$ and $G(x)$ over $\mathbb{Z}_q$ of degree k-1:

   $$\begin{aligned} F(x) &= s + a_1 x + \cdots + a_{k-1} x^{k-1}, \\ G(x) &= t + b_1 x + \cdots + b_{k-1} x^{k-1}. \end{aligned}$$

   D broadcasts $E_i = g^{a_i} h^{b_i} \mod q$ for $i = 1, 2, \ldots, k-1$.

   (c) Let $s_i = F(i), t_i = G(i) \mod q$ for $i = 1, 2, \ldots, n$. Then D sends (s_i, t_i) secretly to P_i for $i = 1, 2, \ldots, n$.

2. When P_i has received his share (s_i, t_i) he verifies that

   $$g^{s_i} h^{t_i} = \prod_{j=0}^{k-1} (E_j)^{i^j}. \tag{2.1}$$

3. Let $S \subset \{1, 2, \ldots, n\}$ be a set of k participants such that Eq. (2.1) holds for these k parties. The member in S can find the secret s by the formula

   $$s = \sum_{i \in S} l_i s_i \quad \text{where} \quad l_i = \prod_{i \in S, i \neq j} \frac{i}{i-j}.$$

   Note that they can also find t by the formula

   $$t = \sum_{i \in S} l_i t_i.$$

Some of the main properties of Pedersen-VSS are summarized in the next Lemma which comes from [19], and are used in the analysis of our protocol in Chapter 4.

Lemma 2.4 ([44]) Pedersen-VSS *satisfies the following properties in the presence of an adversary that corrupts at most k-1 parties and which cannot compute* $\log_g h$*:*

1. *If the dealer is not disqualified during the protocol then all honest players hold shares that interpolate to a unique polynomial of degree k-1. In particular, any k of these shares suffice to efficiently reconstruct (via interpolation) the secret s.*

2. *The protocol produces information (the public values* E_i *and private value* s_i*) that can be used at reconstruction time to test for the correctness of each share; thus, reconstruction is possible, even in the presence of malicious players, from any subset of shares containing at least k correct shares.*

3. *The view of the adversary is independent of the value of the secret s, and therefore the secrecy of s is unconditional.*

In the Chapter 3, we describe our proposed password-based roaming protocol making use of the (k,n)-*threshold scheme*, Pedersen-VSS, in which a user distributes secrets to multiple servers, assuming $n \geq 2k - 1$ [44, 19, 37].

2.2 Related Works

After EKE [5] being resistant to off-line attacks was introduced in 1992 by Bellovin and Merritt, a lot of password-based protocols followed such as A-EKE [6], SPEKE [25], B-SPEKE [26], and PAK [7]. A formal model of security was presented by Halevi and Krawczyk in [22], where they also proposed

a provably secure protocol for the case in which the authentication server has a certified public key known to the client. Ford and Kaliski [16] introduced the idea of sharing the password information among several servers in order to prevent leaking the passwords to an attacker, but they did not give a formal proof of security. Since then, k-out-of-n threshold password authentication protocols have been presented by MacKenzie, Shrimpton and Jakobsson [41], and by Raimondo and Gennaro [46], respectively. The former is provably secure in the random oracle model, and the latter is in the standard model.

Those password-based techniques can be categorized into three classes as follows [24]:

1. *Balanced Password-authenticated Key Agreement (BPKA) schemes*, in which two parties use a shared password to negotiate one or more shared ephemeral keys such that the shared keys are established if and only if they use the same password. The shared keys may then be used for password-based entity authentication or symmetric cryptography.

2. *Augmented Password-authenticated Key Agreement (APKA) schemes*, in which two parties (denoted Client and Server) use related password-derived values to negotiate one or more shared ephemeral keys such that the shared keys are established if and only if they use values that correspond to the same password. The Server uses password verification data that is derived from a one-way function of the Client's password data. The shared keys may then be used for password-based entity authentication or symmetric cryptography.

3. *Password-authenticated Key Retrieval (PKR) schemes*, in which a Client determines one or more static keys that are derived from a password in a negotiation with a Server that knows data associated with the password. The static keys may then be used for entity authentication or symmetric cryptography.

All these schemes require one or more parties to use specific password-related data to make the protocol succeed. But when the participant does not use the correct password-related data, then the protocol is designed to fail in a way that does not reveal the password to those that don't already know the password.

2.2.1 BPKA Scheme

The EKE [5] was the first password-authenticated key agreement protocol. The idea of EKE was to use the password to symmetrically encrypt the protocol messages of a standard key exchange (*e.g.*, Diffie-Hellman [15]). Following EKE, many protocols for password-authenticated key agreement were proposed, such as SPEKE [25], PPK [7], and PAK [7].

We here briefly describe PPK which requires only two rounds of communication having implicit authentication.

Let k and l denote security parameters, where k can be considered to be a general security parameter for hash functions and secret key, and $l > k$ can be considered to be a security parameter for discrete-log-based public keys. Let q of size at least k and p of size l be primes such that $p = rq + 1$ for some value r co-prime to q. Let g be a generator of a subgroup of $\mathbb{Z}_q^*$ of size of q.

Define hash function $H_1, H_1' : \{0,1\}^* \rightarrow \{0,1\}^\eta$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$, where $\eta \geq l + k$, assuming that H_1, H_1', and H_2 are independent random functions. We let π denote the function assigning passwords to pair of users. Figure 2.1 depicts the PPK protocol, with $\pi = \pi[A, B]$. The resulting session key is K.

2.2.2 APKA Scheme

A-EKE [6] was the first verifier-based protocol to resist a password file compromise. Following A-EKE, many APKA protocols have been developed, *e.g.*,

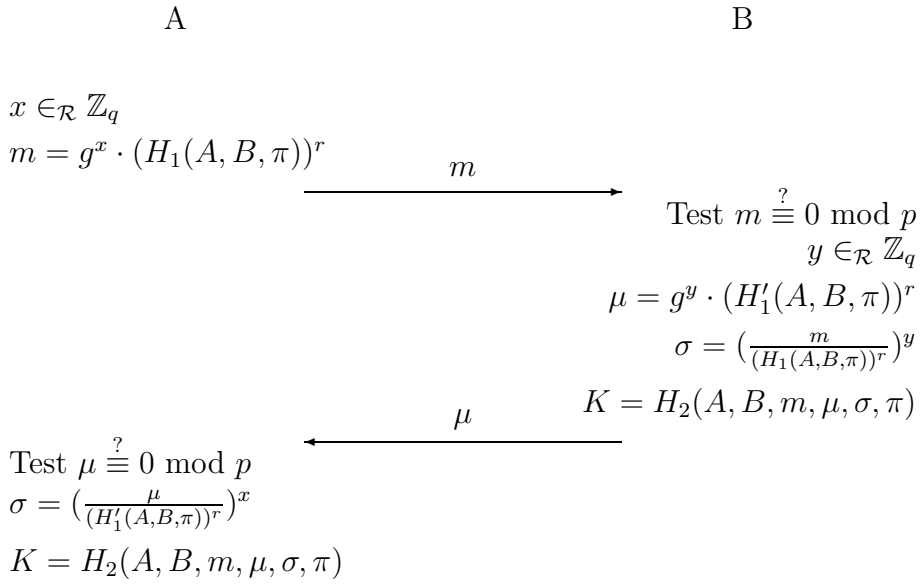

Figure 2.1: The PPK protocol

BSPEKE [26], SRP [52], PAK-X [7], and AMP [35].

We here briefly review SRP. Figure 2.2 shows details of the protocol. In SRP, all computations are performed in a finite field $GF(n)$. To establish a password π with the server, the client picks a random salt s, and computes $x = H(s, \pi)$ and $v = g^x$, where $H()$ is one-way hash function. The server stores v and s as client's password verifier and salt. And then, x is discarded because it is equivalent to the password π. As shown in Figure 2.2, the resulting session key is K.

2.2.3 PKR Scheme

A PKR scheme differs from a PKA scheme in two ways: (1) A PKR scheme derives a password-based *static* key, whereas a PKA scheme derives an *ephemeral key*, and (2) A PKR scheme derives a key for Client that is not (necessarily) derivable by Server, whereas a PKA scheme derives a key shared by both

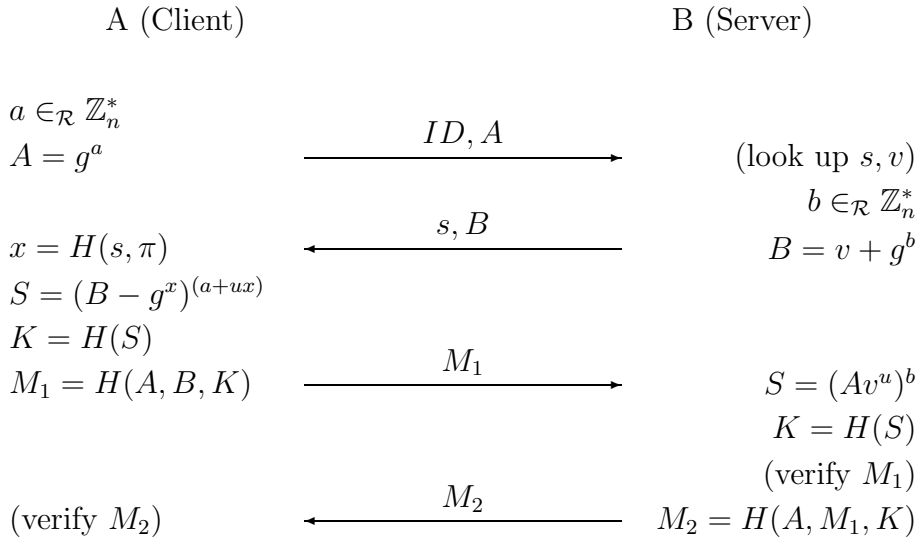

Figure 2.2: The SRP protocol

parties.

Perlman and Kaufman presented protocols [45] that one can securely retrieve a private key and use this to download the rest of one's security context. Ford and Kaliski proposed the idea of sharing the password information among several servers in order to prevent leaking the passwords to an attacker. Jablon in [27] improved on the [16]. Both of them made use of the multiple servers to gain the goal of the protocol.

When some of the servers are compromised, the user can not obtain valid secret key no matter what the user has a method to verify the key. In our proposed scheme, we mainly address this problem.

Now, let's briefly review the protocol proposed in [27] which we refer to as the *Jab01* protocol.

In this protocol two security parameters are defined, j which represents the desired bit-strength for symmetric functions, and k representing the number of bits required for the modulus of asymmetric functions. The protocol operates in a subgroup of order q in $\mathbb{Z}_p^*$, where p, q and r are odd primes,

$p = 2rq + 1, 2^{k-1} < p < 2^k, r \neq q$, and $2^{2^j-1} < q < 2^{2^j}$.

At first, the user, Alice, selects a password π, computes $g_\pi = h(\pi)^{2r}$, and creates a private key SK. For each $i \in [1, n]$, she computes a secret key share $S_i = {g_\pi}^{y_i}$ using randomly chosen $y_i \in_{\mathcal{R}} [1, q-1]$. She then creates her master j-bit symmetric key with $K_m = h(S_1 \parallel \cdots \parallel S_n) \bmod 2^j$, creates her encrypted private key as $U_K = E_{K_m}(SK)$, and then creates her key verifier $V_{K_m} = h(K_m \parallel g)$. They must perform these actions using an authenticated communication method that assures the proper identity of ID_A:

1. *Client*: send $\{ID_A, y_i, U_K, V_{K_m}\}$ to each server L_i for all $i \in [1, n]$.

2. *Servers*: store them in a list C_i maintained on each server.

To retrieve her master key at a later time, the client and servers perform the protocol as in Figure 2.3.

Client		Servers(L_i)
$x \in [1, q-1]$ $X = {g_\pi}^x \bmod p$	$\xrightarrow{ID_A, X}$	
		Retrieve $\{ID_A, y_i, U_K, V_{PK_m}\}$ $Y_i = X^{y_i}$
	$\xleftarrow{Y_i, U_K, V_{K_m}}$	
$S_i = {Y_i}^{1/x} \bmod p$, for each $i \in [1, n]$ $K'_m = h(S_1 \parallel S_2 \parallel \cdots \parallel S_n)$ If $V_{K_m} \neq h(K'_m \parallel g)$, abort. $SK = D_{K'_m}(U_K)$		

Figure 2.3: Authenticated Retrieval Protocol in Jab01

Chapter 3
The Proposed scheme

3.1 Model

Our model for multi-server roaming protocol is similar to that of [27], but with some different features.

First, our scheme employs the concept of threshold, where the user plays the role of a dealer to distribute secret shares to n servers. To do this, we make use of the Pedersen-VSS protocol [44] in a different way that only the user who knows an extra information, *password*, can obtain the secret value derived from the password in collaboration with threshold servers. While the protocol in [27] uses a n-out-of-n solution, *i.e.* the password information is shared among n servers and they *all* must cooperate to authenticate the user, the protocol in our model uses k-out-of-n solution. In addition, even if an adversary compromises k or more servers, she cannot reconstruct the secret value, without knowing user's password.

Second, our scheme is based on bilinear pairings that could be built from Weil pairing or Tate pairing over *GDH* group, which *CDH* problem is hard but *DDH* problem is easy.

On the other hand, although we don't consider *forgiveness protocol* introduced in [27] by which user's honest mistakes are forgiven by sending evidence of recent prior invalid access attempts after each successful authentication, this forgiveness can be adapted in our system.

Figure 3.1 depicts the concept of our model.

There are two phases: (1) Enrollment phase — The user enrolls his credentials in the servers upon his own client terminal which may hold user's private information. (2) Retrieval phase — The user may move to other place where he is just able to use different client terminal which does not hold any user-specific information. The protocol, however, allows the user to download a private key from remote servers with knowledge of only his identity and password.

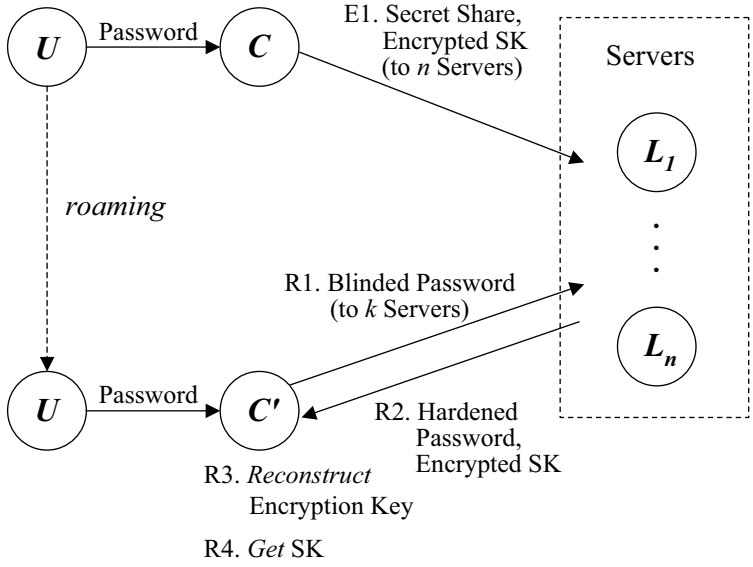

Figure 3.1: The concept of our model

Enrollment [$\mathcal{TPS}$ Protocol]. The client constructs (k,n)-threshold system in the similar way as in Pedersen-VSS [44]. The client creates n shares by using the Pedersen-VSS scheme. k shares will contribute to reconstruct the master symmetric key K_m which is derived from a user's password π. Then, the client have user's private key SK encrypted with the symmetric key K_m. Finally, he creates a proof value V that links the password to his master key.

The client sends secretly share Y_i, encrypted private key U_K, and the proof value V to each server.

As in [27], the enrollment protocol flow has to be performed through a secure channel that authenticates the identity of the user to each i^{th} server L_i.

Authenticated Retrieval [$\mathcal{TPR}$ Protocol]. When at any available client terminal, the user wants to download his private key stored in the server, the client first performs the threshold protocol with at least k servers. Note here that no client terminal has a user's information created at enrollment time.

The client randomly chooses at least k servers and sends them a randomly blinded form of the password to each server. On receiving the request, each server in turn responds with a blinded reply consisting of the blinded password. At least one of the servers also sends the client the encrypted private key U_K and proof value V.

The client reconstructs user's master key K_m using the shares and user's password, and then verifies whether the master key is correct using the proof value V and the master key K_m. Finally, if the master key is correct, the user gets his private key SK by decrypting U_K with the master key.

3.2 Definitions

Communication Model. We assume that our computation model is composed of a set of n players $\{L_1, \ldots, L_n\}$. They are connected by a complete network of secure point-to-point channels. The players have access to a dedicated broadcast channel in which when a player sends a broadcast message all other players can receive the message and know exactly from whom the message sent.

The Adversary. We assume that there exists an adversary, $\mathcal{A}$, who can corrupt up to $k-1$ of the n servers in the network, where $n \geq (2k-1)$. We consider a malicious adversary that may cause corrupted players to divert from the specified protocol in any way. We assume that the computational power of the adversary is adequately modeled by a probabilistic polynomial time Turing machine. Our adversary is *static*, *i.e.* chooses the corrupted players at the beginning of the protocol.

Now, we will give some definitions which are similar to that of [18, 19], upon which we will analyze our protocol.

In the following we assume that there are a dealer C and n players $\{L_1, \ldots, L_n\}$. We say that C `broadcasts` a message m, when she puts m on the broadcast channel for everybody to hear it. In particular $\mathcal{A}$ can hear the message too. We say that C `distributes` a message if she puts m on the private channels connecting her to all the other players. Notice that $\mathcal{A}$ can see m only if somebody has been corrupted.

Let $\mathcal{P}$ be a pair of protocols where the second is always executed after the first one, $\mathcal{P}$=`(Share-Verify, Recover)` for the players $\{L_1, \ldots, L_n\}$ and a dealer C.

Definition 3.1 (View) *The* adversary view, $\text{View}^{\mathcal{P}}_{Network,\mathcal{A}}(\cdot)$ *during protocol* $\mathcal{P}$ *is the probability distribution over the set of computational histories (traffic and coin tosses) of the bad players.*

Sometimes we accompany some distributed protocol $\mathcal{P}$ we propose by a description of a *simulator* `Sim` which is needed in an analysis of the security of this protocol. `Sim` is an algorithm that plays the role of the honest players. $\mathcal{A}$ interacts with `Sim` as if she was interacting with the network. `Sim` tries to create a view for $\mathcal{A}$ that is indistinguishable form the real one. That is, the process of simulation is a computation of two interactive algorithms, `Sim` and $\mathcal{A}$, where the simulator controls the uncorrupted players, and $\mathcal{A}$ controls the

corrupted players. Therefore a description of a simulation process is similar to a description of the protocol itself [28].

Definition 3.2 *The protocol $\mathcal{P}$ is called k-secure (or secure with threshold k) if in the presence of an attacker that corrupts at most k-1 parties the following requirements for correctness and secrecy are satisfied:*

Correctness:

1. *All subsets of k shares provided by honest players define the same unique secret value.*
2. *Secret value is uniformly distributed.*

Secrecy: (Simulatability) *For every (probabilistic polynomial-time) adversary $\mathcal{A}$, there exists a (probabilistic polynomial-time) simulator* `Sim`*, such that on input an element Y, produces an output distribution which is polynomially indistinguishable from $\mathcal{A}$'s view of a run of $\mathcal{P}$ that ends with Y as its output, and where $\mathcal{A}$ corrupts up to k-1 parties. That is, the adversary view* $\mathrm{View}^{\mathcal{P}}_{Network,\mathcal{A}}(\cdot)$ *is identical with* $\mathrm{View}^{\mathcal{P}}_{Sim,\mathcal{A}}(\cdot)$ *which is the adversary view of the simulated execution of the protocol.*

A simulator of each subprotocol exhibits a secrecy property of this subprotocol, which states that the adversary learns nothing from the protocol beyond the public inputs and outputs of this protocol, or in other words, that the adversary learns as much by participating in the threshold computation as he would learn from observing this operation as a block-box.

We now come up with the following definition of *the secure threshold password-authenticated key retrieval protocol* (TPKR for short).

Definition 3.3 *In our* $\mathsf{TPKR} = (\mathcal{TPS}, \mathcal{TPR})$*, Let two kinds of* (`static`) *adversaries exist as follows:*

1. *Strong Adversary: the adversary is able to corrupt and to control k or more servers if he desires.*

2. *Weak Adversary: Not strong adversary, i.e. the adversary is just able to corrupt at most k-1 servers.*

Definition 3.4 (**Secure** TPKR) *Let TPKR be the (k,n)-threshold password-authenticated key retrieval protocol, where 2k-1≤n. TPKR is a strongly secure protocol if:*

1. *The protocol is k-secure satisfying Definition 3.2 in the presence of the weak adversary of Definition 3.3.*

2. *No adversary, even strong one, without knowing user's password is able to reconstruct the master symmetric key K_m, and is thus able to obtain the private key SK.*

3.3 Detailed Protocol

We let ℓ be the security parameter given to the setup algorithm. We let $\mathcal{G}$ be some GDH parameter generator.

3.3.1 System Setup

Given a security parameter ℓ, the algorithm $\mathcal{G}$ works as follows:

1. Run $\mathcal{G}$ on input ℓ to generate a prime $q \geq 2^{\ell}$, two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same order q and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

2. Choose two arbitrary generators P and $P' \in \mathbb{G}_1$, where $P' = \alpha P$ for some $\alpha \in \mathbb{Z}_q$ and the computing α given P and P' is infeasible.

3. Choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^\kappa$, and $H_3 : \{0,1\}^\kappa \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$, for some κ. The security analysis will view H_1, H_2, and H_3 as random oracles.

4. The system parameters are $\mathsf{params} = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1, H_2, H_3, P, P'\}$. And then publish them.

3.3.2 Enrollment Protocol $\mathcal{TPS}$

The enrollment protocol makes use Pedersen-VSS protocol [44], but we use elliptic curve notions for discrete logarithm problem.

Denote n servers involving in the protocol as $\{L_1, \ldots, L_n\}$ and the client playing the role of a dealer as C. Let ID and SK be the user's identity and the private key, respectively.

The user picks a password π. We assume π can be mapped into $\mathbb{Z}_q$, and thus we use π as if they were elements of $\mathbb{Z}_q$. The user then performs the protocol upon the client as follows:

1. The client C, as a dealer, distributes user's credentials.

   (a) Select randomly y and $z \in \mathbb{Z}_q^*$ which are uniformly distributed as in [44].

   (b) Choose two random polynomials $f(x)$ and $g(x)$ over $\mathbb{Z}_q$ of degree $k-1$ such that $f(0) = a_0 = y$ and $g(0) = b_0 = z$. Let

   $$\begin{aligned} f(x) &= a_0 + a_1 x + \cdots + a_{k-1} x^{k-1} \quad \text{and} \\ g(x) &= b_0 + b_1 x + \cdots + b_{k-1} x^{k-1}. \end{aligned}$$

   (c) Compute and broadcast $E_i = E(a_i, b_i) = a_i P + b_i P'$ for $i = 0, \ldots, k-1$.

   (d) Compute $K = \hat{e}(yR_{ID}, Q_{ID})$, then create the master symmetric key $K_m = H_2(K)$, where $R_{ID} = H_1(\pi)$ and $Q_{ID} = H_1(ID)$. Then

create the encrypted private key $U_K = E_{K_m}(SK)$ and key verifier $V = H_3(K_m, P)$, and let $Y_i = f(i) \cdot Q_{ID}$ and $Z_i = g(i) \cdot Q_{ID}$ for $i = 1, \ldots, n$.

(e) Send $\{ID, Y_i, Z_i, U_K, V\}$ *secretly* to each player L_i for all $i \in [1, n]$.

2. L_i has received information from C.

(a) Verifies that

$$\hat{e}(Y_i, P) \cdot \hat{e}(Z_i, P') \stackrel{?}{=} \hat{e}\left(Q_{ID}, \sum_{j=0}^{k-1} i^j \cdot E_j\right). \tag{3.1}$$

(b) If Eq.(3.1) is verified to be false, response a *complaint* against C. Otherwise accept and store $\{ID, Y_i, U_K, V\}$ in a storage maintained on each L_i.

3. C discards all information, and completes the enrollment protocol.

For the sake of convenience, we assume that the client has received no complaint in Step 2.

3.3.3 Authenticated Retrieval Protocol $\mathcal{TPR}$

For authenticated retrieval, the client and k servers perform the actions as the following. Denote k servers by $B = \{L_i \mid 1 \leq i \leq k\}$.

1. C sends k servers a request message.

(a) Select a random number uniformly distributed $x \in \mathbb{Z}_q^*$, compute $X = xR_{ID}$.

(b) Send X and ID to each server L_i for $i \in B$.

2. On receiving the request, each server L_i responds as follows:

(a) Retrieve $\{ID, Y_i, U_K, V\}$ from the storage maintained securely on each L_i.

(b) Compute $R_i = \hat{e}(X, Y_i)$, and then reply $\{R_i, U_K, V\}$ to the client.

3. Finally, the client reconstructs user's private key by performing the following:

(a) Compute $l_i = \prod_{j \in B, j \neq i} \frac{j}{j-i}$ for each i^{th} server.

(b) Compute $R'_i = (R_i)^{l_i x^{-1}}$ and $K' = \prod_{i \in B} R'_i$.

(c) Generate $K'_m = H_2(K')$.

(d) If $V \neq H_3(K'_m, P)$, abort.

(e) To obtain the private key, decrypt U_K with the master key K'_m.

Completing the protocol successfully, the client reconstructs the user's private key SK. Figure 3.2 depicts the retrieval protocol.

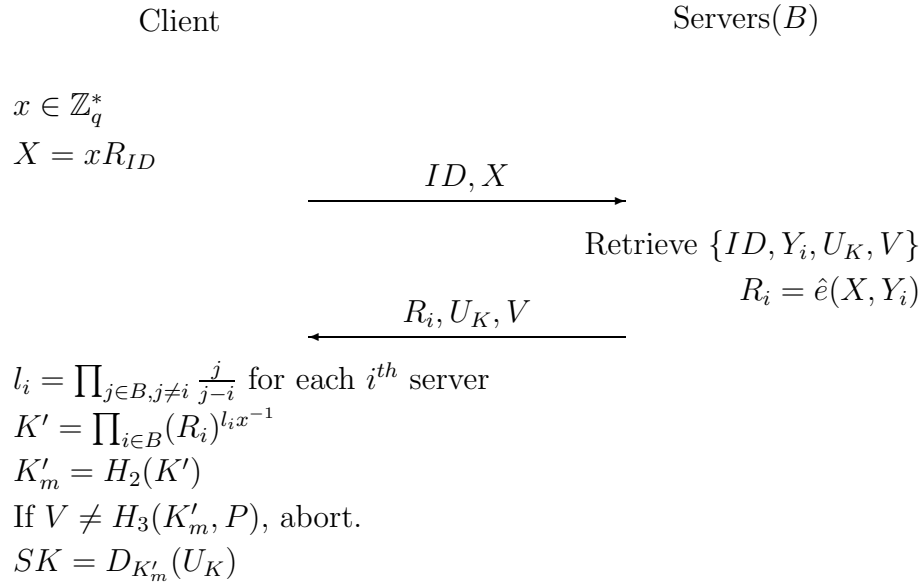

Figure 3.2: Our Threshold Key Retrieval Protocol

Chapter 4
Analysis

4.1 Security Proof

In this section, we discuss the security aspects of our proposed scheme TPKR, and give complete security proof of that using the definitions introduced in the previous chapter.

As a result, the security for our protocol arrives at the security for secret value K, assuming that the adapted symmetric cryptosystem is secure and thus nobody can obtain the private key SK without knowing K. Remember that $UK = E_{K_m}(SK)$, where $K_m = H_2(K)$.

Lemma 4.1 (Correctness) *The protocol* TPKR=$(\mathcal{TPS}, \mathcal{TPR})$ *from Section 3.3 satisfies the correctness of Definition 3.2 with threshold k, for any* $k \leq (n+1)/2$.

Proof: First note that all the honest players indeed hold the verification equation Eq.(3.1) as follows:

$$\hat{e}(Y_i, P) \cdot \hat{e}(Z_i, P') = \hat{e}\left(Q_{ID}, \sum_{j=0}^{k-1} i^j \cdot E_j\right).$$

Since

$$\begin{aligned} \hat{e}(Y_i, P) \cdot \hat{e}(Z_i, P') &= \hat{e}(f(i)Q_{ID}, P) \cdot \hat{e}(g(i)Q_{ID}, P') \\ &= \hat{e}(Q_{ID}, f(i)P + g(i)P'), \end{aligned}$$

where

$$\begin{aligned} f(i)P + g(i)P' &= \sum_{j=0}^{k-1} i^j (a_j P + b_j P') \\ &= \sum_{j=0}^{k-1} i^j E_j. \end{aligned}$$

1. From part 1 of Lemma 2.4, we know that all honest players hold shares (Y_i) which contribute to reconstruct unique secret by combining with client's request message as in step 2 of $\mathcal{TPR}$ protocol. For any set $\mathcal{B}$ of k shares and extra value (X) from client's request message, the unique secret is computed as follows:

$$\begin{aligned} K' &= \prod_{i \in \mathcal{B}} \hat{e}(X, Y_i)^{l_i x^{-1}} \\ &= \prod_{i \in \mathcal{B}} \hat{e}(xR_{ID}, f(i)Q_{ID})^{l_i x^{-1}} \\ &= \prod_{i \in \mathcal{B}} \hat{e}(f(i) \prod_{j \in \mathcal{B}, j \neq i} \frac{j}{j-i} R_{ID}, Q_{ID}) \\ &= \hat{e}(\sum_{i \in \mathcal{B}} f(i) \prod_{j \in \mathcal{B}, j \neq i} \frac{j}{j-i} R_{ID}, Q_{ID}) \\ &= \hat{e}(yR_{ID}, Q_{ID}), \end{aligned}$$

where l_i are appropriate Lagrange interpolation coefficients for the set $\mathcal{B}$. Since the above holds for any set of k correct shares then K' is uniquely defined, where the same extra value (X) which as said is derived from the user's password has to be given to the protocol $\mathcal{TPS}$ and $\mathcal{TPR}$.

2. Let's consider the secret value K. We can let K be $g^{\lambda y}$ for some λ where g is a generator of group $\mathbb{G}_2$. Since y is chosen randomly from $\mathbb{Z}_q^*$ as in [44], therefore $K = g^{\lambda y}$ is also a random element in the group $\mathbb{G}_2$. On the other hand, by virtue of part 3 of Lemma 2.4, the view and thus actions of the adversary are independent of the secret y. ■

As a result, we can state that TPKR can be resistant to corruption of even $k-1$ of $n \geq 2k-1$ servers. A user chooses randomly a secret value y uniformly distributed in $\mathbb{Z}_q^*$ during the execution of $\mathcal{TPS}$. Even there exists an adversary who can corrupt at most $k-1$ servers among $n \geq 2k-1$, any subset of k servers constructs the secret value K uniformly distributed in $\mathbb{G}_2$.

Lemma 4.2 (Secrecy) *Protocol* TPKR *from Section 3.3 satisfies the secrecy of Definition 3.2.*

Proof: It can be proved in a similar way that is used to prove Lemma 3 in [28]. We can state that, for any input secret y and y', if the dealer $\mathcal{C}$ is uncorrupted then there is no difference between the adversarial view of an execution of TPKR in which $\mathcal{C}$ shares y', from a view of TPKR in which $\mathcal{C}$ shares y.

There exists a Sim such that for every $n/2$-threshold static secure-channels adversary $\mathcal{A}$ with history $h_{\mathcal{A}}$, for any given system parameter `params`. Let E stand for an instance (P, P') of Pedersen commitment [44]. Let $f(x), g(x)$ be any $k-1$ degree polynomials such that $f(0) = y$. Consider a run of TPKR in which $\mathcal{C}$ uses polynomials $f(x), g(x)$ and the random input of $\mathcal{A}$ is $r_{\mathcal{A}}$. Note that once we fix $f(x), g(x)$, $r_{\mathcal{A}}$ then everything else in the run of this protocol is determined. Denote the outputs of such run as $\mathcal{TPKR}_{\mathcal{A}}^{Data}$ $((h_{\mathcal{A}}, r_{\mathcal{A}}), E; f(x), g(x))$. We denote the set of corrupted players as $P_{\mathcal{B}}$ and the set of uncorrupted players as $P_{\mathcal{G}}$.

Note that any $k-1$ degree polynomials $f'(x), g'(x)$ such that $f'(i) = f(i)$ for $L_i \in P_{\mathcal{B}}$ and $f(x)+\alpha g(x) = f'(x)+\alpha g'(x)$ where $P' = \alpha P$, the adversary's output in $\mathcal{TPKR}_{\mathcal{A}}^{Data}$ $((h_{\mathcal{A}}, r_{\mathcal{A}}), E; f'(x), g'(x))$ is the same as in $\mathcal{TPKR}_{\mathcal{A}}^{Data}$ $((h_{\mathcal{A}}, r_{\mathcal{A}}), E; f(x), g(x))$.

If we fix $y, y', r_{\mathcal{A}}$, and range the polynomials $f(x), g(x)$ among all $k-1$ degree polynomials such that $f(0) = y$, then we see that the distribution of the adversary view in the following two cases are equal, for every y, y', and $r_{\mathcal{A}}$:

1. TPKR on $f'(x), g'(x)$, and $r_{\mathcal{A}}$ followed by protocol on the resulting $\mathcal{TPKR}_{\mathcal{A}}^{Data}$,
   where $f'(x), g'(x)$ are random $k-1$ degree polynomials such that (0) $f'(0) = y'$; (1) $f'(i) = f(i)$ on $L_i \in P_{\mathcal{B}}$; (2) $f'(x)+\alpha g'(x) = f(x)+\alpha g(x)$ where $f(x), g(x)$ are random $k-1$ degree polynomials such that $f(0) = y$.

2. TPKR on $f(x), g(x)$, and $r_{\mathcal{A}}$ with outputs denoted $\mathcal{TPKR}_{\mathcal{A}}^{Data}[y]$, followed by protocol on inputs $\mathcal{TPKR}_{\mathcal{A}}^{Data}[y']$ output by replacement procedure $\mathcal{T}_{TPKR}$ $(\mathcal{TPKR}_{\mathcal{A}}^{Data}[y], y', \alpha)$ where $f(x), g(x)$ are random $k-1$ degree polynomials such that $f(0) = y$.

   $\mathcal{T}_{TPKR}(\mathcal{TPKR}_{\mathcal{A}}^{Data}[y], y', \alpha)$ mentioned above takes as inputs a given secret-sharing $\mathcal{TPKR}_{\mathcal{A}}^{Data}[y]$, target value y' and α s.t. $P' = \alpha P$, and outputs its replacement $\mathcal{TPKR}_{\mathcal{A}}^{Data}[y']$ as in [28]. Note that the data which is visible to $\mathcal{A}$, *i.e.* the public data and the private data of the players controlled by the $\mathcal{A}$, must remain the same in $\mathcal{TPKR}_{\mathcal{A}}^{Data}[y]$ and $\mathcal{TPKR}_{\mathcal{A}}^{Data}[y']$, so this *replacement* is always only a modification of the private data of the players controlled by the simulator.

From above description, (1) since $f(x)$ is a random polynomial such that $f(0) = y$, then $f'(x)$ is a random polynomial such that $f(0) = y'$; and (2) there is a one-to-one mapping between a choice of $g(x)$ and a choice of $g'(x)$, and thus since $g(x)$ is a random polynomial then so $g'(x)$. ■

Given Lemmas 4.1 and 4.2, we can state the following theorem:

Theorem 4.3 *Assume that $n \geq 2k-1$. Then the protocol* TPKR *is k-secure threshold-authenticated roaming protocol according to Definition 3.2 with fault-tolerance k.*

The following lemma shows the security of the protocol $\mathcal{TPR}$ against a strong adversary who corrupts k or more servers if he desires.

Lemma 4.4 *Under GDH assumption, the protocol $\mathcal{TPR}$ is secure against a strong adversary defined by Definition 3.3.*

Proof: Let a *strong adversary* $\mathcal{A}$ be able to corrupt k or more servers and thus to obtain at least k shares. In this case, we need to show that $\mathcal{A}$ can not reconstruct the secret value K' without knowing the user's password.

In order to break the protocol, $\mathcal{A}$ tries to compute K'' such that

$$K'' = \prod_{i \in S} \hat{e}(R', Y_i)^{l_i}, \tag{4.1}$$

with knowledge of system parameter params, any set $\mathcal{S}$ of t secret shares Y_i for $i = 1, 2, \ldots, t$ $(t \geq k)$ and client's request messages X, where l_i is appropriate Lagrange interpolation coefficients for the set $\mathcal{S}$.

We assume $\mathcal{A}$ has three options to compute K'': (1) Solve DLP, *i.e.* find an integer n such that $Q = nP$, (2) Solve CDHP in such a way that given $(P, \alpha P, \beta P) \in \mathbb{G}_1$ compute $\alpha\beta P$, and (3) Guess correct password, *e.g.*, by mounting password guessing attack.

1. In order to compute Eq.(4.1), $\mathcal{A}$ first unblinds X to obtain R', *i.e.* find an integer x (or $x' = x\beta$) such that $R' = xR_{ID}$ (or $R' = x'P$). Thus no adversary can compute R', under the assumption DLP is hard.

2. Let $Q_{ID} = \alpha P$, $R_{ID} = \beta P$. Even given a triple $(P, \alpha P, y\beta P)$, $\mathcal{A}$ can not compute $\alpha\beta yP$, such that $K'' = \hat{e}(yR_{ID}, Q_{ID}) = \hat{e}(P, P)^{\alpha\beta y} = g^{\alpha\beta y}$ where g is a generator of $\mathbb{G}_2$, assuming that $\mathbb{G}_1$ is a GDH group.

On the other hand, given $\{Y_i = y_i Q_{ID} \mid i = 1, 2, \ldots, t\}$, $\mathcal{A}$ can compute the following at the best:

$$\begin{aligned} Y'' &= \prod_{i \in S} \hat{e}(y_i Q_{ID}, P)^{l_i} \\ &= \hat{e}(yQ_{ID}, P) \\ &= \hat{e}(P, P)^{\alpha y} \\ &= g^{\alpha y}. \end{aligned}$$

3. Let π' be a password guess by $\mathcal{A}$. Thus $R' = H_1(\pi')$. Now, $\mathcal{A}$ computes the following:

$$\begin{aligned} K'' &= \prod_{i \in S} \hat{e}(R', Y_i)^{l_i} \\ &= \prod_{i \in S} \hat{e}(R', f(i)Q_{ID})^{l_i} \\ &= \hat{e}(yR', Q_{ID}), \end{aligned}$$

However, $\mathcal{A}$ can not verify whether his guess is correct or not. More over, by Lemma 4.2 it is impossible to distinguish K'' from K'. ■

Theorem 4.5 *Assume $n \geq 2k - 1$. Then the protocol* TPKR *is strongly secure according to Definition 3.4, under GDH assumption.*

Proof: The proof of the theorem comes immediately from Theorem 4.3 and Lemma 4.4. ■

4.2 Comparison

With mainly compared to [16] and [27], our scheme is capable of resisting that fewer than threshold servers are compromised. When only k honest servers are involved in the protocol, the user can retrieve his private key with knowledge of his own password. Besides, even attacker has succeeded in compromising k or more servers but without knowing the user's password, she still cannot obtain any information about the user's credential.

Table 4.1 depicts computation load of TPKR compared with that of [27]. We denote **E** and **M** by computation load for exponentiation and multiplication, respectively. Let n be the number of servers and k be threshold.

From Table 4.1, we see that our scheme TPKR is more efficient one than *Jab01* with respect to the computation during the retrieval, since the inequality $n\mathbf{E} \geq k\mathbf{E}+k\mathbf{M}$ holds, where $n \geq 2k - 1$, *i.e.* $(k - 1)\mathbf{E} \geq k\mathbf{M}$ if $k \geq 2$.

Table 4.1: Computation in the Retrieval on the Client Side

	Jab01 [27]	TPKR
Main computation parts	$S_i = R_i^{x^{-1}}$ $K' = h(S_1 \parallel \cdots \parallel S_n)$	$R'_i = R_i^{l_i x^{-1}}$ $K' = h(\prod_{i \in B} R'_i)$
Computation load	$n\mathbf{E}$	k(**E**+**M**)

However with respect to the server side, the computation load of our protocol may be less efficient due to pairing operation of which, as known, costs several times expensive than that of an exponentiation [12].

Chapter 5
Conclusions

In this thesis, we have studied the design and analysis of secure (n, k)-threshold password-authenticated roaming protocol. We have reviewed previous works related to the password-based protocol as well as other cryptographic tools which are used in construction of our protocol.

We have presented a new threshold password-authenticated roaming protocol based on GDH group using bilinear pairings. It allows a roaming user to download a private key from remote servers, without revealing the password to off-line guessing. We note that, as a goal of a multi-server roaming system, a protocol has to allow a user to get his private key from the servers, even if some of the servers are compromised. With this point of view, we give a description of a threshold password-only roaming protocol using pairings.

In this paper, we use (k, n)-*threshold scheme* in which only k honest servers or more are engaged to reconstruct a secret value. Furthermore, in our protocol, even attacker has succeeded in compromising more than k servers but without knowing the user's password, he still cannot obtain any information about the user's credentials. Our scheme is based on bilinear pairings that could be built from Weil pairing or Tate pairing.

As further works, we leave ourselves more rigorous security analysis and more accurate complexity analysis for our protocol. In addition, we have to discuss implementation issues for the protocol [2, 4, 3, 21, 12].

겹선형 쌍을 이용한 Threshold 패스워드-인증 키 Retrieval 프로토콜

이송원

인터넷의 급속한 발전으로 사용자는 네트워크에 용이하게 접근할 수 있게 되었다. 이를 통하여, 서비스 공급자로부터 특정의 서비스를 받거나 비밀 정보를 사전에 서버에 저장한 후 이를 나중에 다운로드 할 수 있다. 이때, 사용자는 자신이 정당한 사용자임을 서버에게 확인시켜 주어야만 한다. 사용자의 신원(ID)을 검증하기 위하여, 많은 실세계 시스템들은 패스워드-기반의 인증을 사용한다. 그러나 대부분의 사용자 패스워드는 상대적으로 작은 집합 공간에서 선택되고 쉽게 기억될 수 있다는 근본적인 문제점을 가지고 있는데, 이것은 곧 공격자가 패스워드를 쉽게 추측해 낼 수 있음을 의미한다.

로밍 사용자는 자신의 전용 단말기 이외의 단말기를 이용하여 네트워크에 접근하여 자신의 ID와 패스워드만을 가지고 원격의 서버들로부터 비밀키를 다운로드하고자 하는 사용자를 말하며, 이때 자신의 비밀정보를 저장하고 있는 스마트 카드 등을 휴대하지 않는다. 스마트 카드는 기밀 정보를 저장하는 중요한 수단으로 사용되지만, 많은 실 환경에서 적용하기 어려운 점이 있는데, 주로 불편함에 기인한다. 예를 들면, 사용자는 스마트 카드와의 통신을 위한 별도의 인터페이스를 필요로 한다. 이러한 관점에서, 강력한 패스워드 기반의 인증 프로토콜들이 제안되고 있으며, Perlman 등[45], Ford 등[16], 그리고 Jablon[27] 등이 제안한 프로토콜이 대표적이다.

본 논문에서는 다중-서버를 이용한 threshold 패스워드 로밍 프로토콜을 제안한다. 기존에 제안된 다중-서버 프로토콜들의 경우, 서버들 중의 일부가 손상된 경우 사용자는 프로토콜을 성공적으로 완료할 수 없다. 즉, 그러한 경우 사용자는 자신의 비밀키를 다운로드 할 수 없게 된다. 본 논문에서

제안하는 프로토콜은 (k, n)-threshold scheme을 사용하는데 단지 k개의 정상적인 서버들만을 통하여서 비밀키를 복구할 수 있다. 또한, 공격자가 k개 이상의 서버를 공격하였을지라도 패스워드를 알고 있지 않는 한 사용자의 어떠한 비밀정보도 알아낼 수 없다. 우리의 scheme은 Weil 쌍이나 Tate 쌍에서 구현될 수 있는 겹선형 쌍에 기반 한다.

References

1. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes", *Advances in Cryptology - CRYPTO '98*, LNCS 1462, pp.26-45, Springer-Verlag, 1998.

2. D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", *Advances in Cryptology - CRYPTO 2001*, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
   Full version available at http://crypto.stanford.edu/ibe.

3. P. Barreto, H. Kim, B. Lynn and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems", *Advances in Cryptology - CRYPTO 2002*, LNCS 2442, pp.354-369, Springer-Verlag, 2002.

4. D. Boneh, B. Lynn and H. Schacham, "Short Signatures from the Weil Pairing", *Advances in Cryptology - ASIACRYPT 2001*, LNCS 2248, pp.514-532, Springer-Verlag, 2001.

5. S. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks", *IEEE Symposium on Research in Security and Privacy*, 1992.

6. S. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: A Password-based Protocol Secure Against Dictionary Attacks and Password File Compromise", Technical Report, AT&T Bell Laboratories, 1994.

7. V. Boyko, P. MacKenzie and S. Patel, "Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman", *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807, pp.156-171, Springer-Verlag, 2000.

8. D. Boneh, "The decision Diffie-Hellman problem", *Proc. Third Algorithmic Number Theory Symposium*, LNCS 1423, pp.48-63, Springer-Verlag, 1998.

9. 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.

10. M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks", *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807, pp.139-155, Springer-Verlag, 2000.

11. J. Baek and Y. Zheng, "Identity-Based Threshold Decryption", *Cryptology ePrint Archive 2003/164*.

12. J. Cha and J. Cheon, "An Identity-Based Signature from Gap Diffie-Hellman Groups", *PKC 2003*, LNCS 2567, pp.18-30, Springer-Verlag, 2003.

13. C. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Adaptive Security for Threshold Cryptosystems", *Advances in Cryptology - CRYPYO '99*, LNCS 1666, pp.98-116, Springer-Verlag, 1999.

14. D. Denning and G. Sacco, "Timestamps in Key Distribution Systems", *Communication of the ACM*, Vol.24, No.8, pp.533-536, 1981.

15. W. Diffie and M. Hellman, "New Direction in Cryptography", *IEEE Transactions on Information Theory*, Vol.22, No.6, pp.644-654, 1976.

16. W. Ford and B. Kaliski, "Server-Assisted Generation of a Strong Secret from a Password", 9^{th} *International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, IEEE, 2000.

17. S. Goldwasser and M. Bellare, "Lecture Notes on Cryptography", available at http://www.cs.ucsd.edu/users/mihir/papers/gb.html, Aug.2001.

18. R. Gennaro, "Theory and Practice of Verifiable Secret Sharing", PhD Thesis, MIT, May 1996.

19. R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems", *Advances in Cryptology - EUROCRYPT '99*, LNCS 1592, pp.295-310, Springer-Verlag, 1999.

20. R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Secure Applications of Pedersen's Distributed Key GenerationProtocol", *CT-RSA 2003*, LNCS 2612, pp.373-390, Springer-Verlag, 2003.

21. F. Hess, "Efficient Identity Based Signature Schemes Based on Pairings", *SAC 2002*, LNCS 2595, pp.310-324, Springer-Verlag, 2003.

22. S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols", *ACM Transactions on Information and System Security*, Vol.2, No.3, pp.230-268, 1999.

23. Y. Hwang, D. Yum and P. Lee, "EPA: An Efficient Password-Based Protocal for Authenticated Key Exchange", *ACISP 2003*, LNCS 2727, pp.452-463, Springer-Verlag, 2003.

24. IEEE, "IEEE P1363.2: Standard Specifications for Password-Based Public-Key Cryptographic Techniques", Draft D2002-2-12.

25. D. Jablon, "Strong Password-Only Authenticated Key Exchange", *ACM Computer Communications Review*, Vol.26, No.5, pp.5-26, 1996.

26. D. Jablon, "Extended Password Key Exchange Protocols Immune to Dictionary Attacks", *WETICE '97*, pp.248-255, 1997.

27. D. Jablon, "Password Authentication Using Multiple Servers", *CT-RSA 2001*, LNCS 2020, pp.344-360, Springer-Verlag, 2001.

28. S. Jarecki, "Efficient Threshold Cryptosystems", PhD Thesis, MIT, June 2001.

29. A. Joux and K. Nguyen, "Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups", *Cryptology ePrint Archive 2001/003*.

30. J. Katz, "A Forward-Secure Public-Key Encryption Scheme", *Cryptology ePrint Archive 2002/060*.

31. S. Kim, B. Kim and S. Park, "Comments on Password-Based Private Key Download Protocol of NDSS'99", *Electronics Letters*, Vol.35, No.22, IEE Press, pp.1937-1938, 1999.

32. J. Kim, H. Kwon, H. Park, S. Kim and D. Won, "An Improvement of Verisign's Key Roaming Service Protocol", *ICWE 2003*, LNCS 2722, pp.281-288, Springer-Verlag, 2003.

33. J. Katz, R. Ostrovsky and M. Yung, "Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords", *Advances in Cryptology - EUROCRYPT 2001*, LNCS 2045, pp.475-494, Springer-Verlag, 2001.

34. C. Kaufman, R. Perlman and M. Speciner, "Network Security: Private Communication in a Public World", Prentice Hall, 2002.

35. T. Kwon and J. Song, "Authentication and Key Agreement via Memorable Password", *Cryptology ePrint Archive 2000/026*.

36. T. Kwon and J. Song, "A Study on the Generalized Key Agreement and Password Authentication Protocol", *IEICE Transactions on Communications*, Vol.E83-B, No.9, pp.2044-2050, 2000.

37. B. Libert and J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems", *PODC '03*, pp.163-171, 2003.

38. P. MacKenzie, "More Efficient Password-Authenticated Key Exchange", *CT-RSA 2001*, LNCS 2020, pp.361-377, Springer-Verlag, 2001.

39. A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a fnite feld", *IEEE Transactions on Information Theory*, Vol.39, No.5, pp.1639-1646, 1993.

40. A. Menezes, P. Oorschot and S. Vanstone, "Handbook of Applied Cryptography", CRC Press LLC, 1997.

41. P. MacKenzie, T.Shirmpton and M.Jakobsson, "Threshold Password-Authenticated Key Exchange(Extended Abstract)", *Advances in Cryptology - CRYPTO 2002*, LNCS 2442, pp.385-400, Springer-Verlag, 2002.

42. T. Okamoto and D. Pointcheval, "The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes", *PKC 2001*, LNCS 1992, pp.104-118, Springer-Verlag, 2001.

43. S. Patel. "Number theoretic attacks on secure password schemes", *IEEE Symposium on Research in Security and Privacy*, pp.236-247, 1997.

44. T. Pedersen, "Non-interactive and Information theoretic Secure Verifiable Secret Sharing", *Advances in Cryptology - CRYPTO '91*, LNCS 576, pp.129-140, Springer-Verlag, 1992.

45. R. Perlman and C. Kaufman, "Secure Password-Based Protocol for Downing a Private Key", *1999 Network and Distributed System Security Symposium*, Internet Society, 1999.

46. M. Raimondo and R. Gennaro, "Provably Secure Threshold Password-Authenticated Key Exchange", *Advances in Cryptology - EUROCRYPT 2003*, LNCS 2656, pp.507-523, Springer-Verlag, 2003.

47. V. Shoup and R. Gennaro, "Securing Threshold Cryptosystems against Chosen Ciphertext Attack", *Advances in Cryptology - EUROCRYPT '98*, LNCS 1403, pp.1-16, Springer-Verlag, 1998.

48. A. Shamir, "How to Share a Secret", *Communication of the ACM*, Vol.22, No.11, pp.612-613, 1979.

49. A. Shamir, "Identity-Based Cryptosystem and Signature Schemes", *Advances in Cryptology - CRYPTO '84*, LNCS 196, pp.47-53, Springer-Verlag, 1985.

50. N. Smart, "An Identity based Authenticated Key Agreement Protocol based on the Weil Pairing", Electronic Letters, Vol.38, No.13, pp.630-632, 2002.

51. D. Vo, F. Zhang and K. Kim "A New Threshold Blind Signature Scheme from Pairings", *SCIS 2003*, Vol.1/2, pp.233-238, Jan.2003.

52. T. Wu, "The Secure Remote Password Protocol", *1998 Network and Distributed System Security Symposium*, pp.97-111, Internet Society, 1998.

Acknowledgements

First and foremost I would like to thank my academic advisor Prof. Kwangjo Kim for his constant direction and support. He always has shown his consistent affection and encouragement for me to carry out my research and life in ICU. Special thanks are due to Prof. Jae Choon Cha and Dr. Dong Hoon Lee for their generosity and serving in my thesis committee.

Particularly, I would like to thank staffs in my company, Korea Financial Telecommunications and Clearings Institute (KFTC), for giving me a good chance to do research. I am also so much thankful to my senior managers and colleagues: Soonju Lee, Sangwhan Han, Sunmi Rho, and so on. And, special thanks go to Hyunju Hong for her administrative support.

I would also like to thank all members of cryptology and information security laboratory: Kyusuk Han, Sangwon Lee, Zeen Kim, Byunggon Kim, Hwasun Chang, Chuljoon Choi, Sungjoon Min, Jaehyrk Park, Joongman Kim, Soogil Choi, Jungkyu Yang, Seokkyu Kang, Vo Duc Lim from Vietnam, Yan Xie, Xiaofeng Chen, Ping Wang, Jiqiang Lv, and Ren Kui from China, and Divyan from India, for giving me lots of interests and good advices during the course of my study. I also thank Jeongmi Choi for helpful support as a staff member

In addition, I would like to thank the graduates: Myungsun Kim, Jongseong Kim, Wooseok Ham, Hyunrok Lee, Hyungki Choi, and Jungyeon Lee for their everlasting guidance in life and study of ICU and I want to present my sincere gratitude to Sangbae Park and Yunkyoung Jeong of HCI lab., Jungbae Park of CN lab., and Juhyung Lee of LIT lab.

My biggest gratitude goes to my parents, and to my parents-in-law for their endless concerns and devotional affection. I cannot forget their trust and encouragement on me. God bless my family to be happy forever.

My love and thanks go to my wife Myungsin Kim for her endless encourage and devotion, and to my lovely son Hyungjoon. I dedicate this work to them.

Curriculum Vitae

Name : Songwon Lee

Date of Birth : Nov. 23. 1968

Sex : Male

Nationality : Korean

Education

1986.3–1993.2	Applied Statistics Korea University (B.A.)
2002.3–2004.2	Cryptology and Information Security, Engineering Information and Communications University (M.S.)

Career

2003.7–	Graduate Research Assistant Ubiquitous System Security Technology: Protecting Digital Contents from Illegal Use NITZ Co.
2003 Summer	Undergraduate Teaching Assistant ICE1212 Introduction to Information Security School of Engineering, ICU

2002.9–2002.12 Graduate Research Assistant
Research on Easy Security Technology
Electronics and Telecommunications Research Institute(ETRI)

2002.3– Graduate Research Assistant
Cultivation of Top Level IT Security Manpower
The Ministry of Information and Communications(MIC)

Academic Experience

2002.3– KIISC (Korea Institute of Information Security and Cryptology) student member

Publications

(1) 2003.12 Songwon Lee and Kwangjo Kim, "Threshold Password-Authenticated Key Retrieval Protocol Using Bilinear Pairings", *Proc. of CISC-W03*, pp.463-468, Hanyang Univ., Korea, Dec.6, 2003.

(2) 2003.10 Songwon Lee, Jeongkyu Yang, and Kwangjo Kim, "Threshold Password-Based Authentication Using Bilinear Pairings", *Proc. of CSS2003*, pp.385-390, Kitakyushu, Japan, Oct.29-31, 2003.

(3) 2003.7 Songwon Lee and Kwangjo Kim, "A Study on the Password-based Authentication Protocol for the Roaming User", *Proc. of CISC2003*, pp.289-292, Paichai Univ., Korea, Jul.4, 2003.

(4) 2002.11 Wooseok Ham, Jongseung Kim, Songwon Lee, JaeHyrk Park, Sugil Choi, Kwangjo Kim, Sookyeon Kim and Taekyong Nam, "Research Trend of QoSS and Its Application", *Proc. of CISC 2002*, pp.352-355, Hankuk Aviation Univ., Korea, Nov.16, 2002.