# Secure and Efficient Identity-based Proxy Multi-signature Using Cubic Residues

Feng Wang[1,2], Chin-Chen Chang[2,3], Changlu Lin[4], and Shih-Chang Chang[5]
*(Corresponding author: Chin-Chen Chang)*

College of Mathematics and Physics, Fujian University of Technology[1]
Fuzhou, Fujian,350108, China
Department of Information Engineering and Computer Science, Feng Chia University[2]
100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
(Email: alan3c@gmail.com)
Department of Computer Science and Information Engineering, Asia University[3]
Taichung 41354, Taiwan
School of Mathematics and Computer Science, Fujian Normal University[4]
Fuzhou, Fujian, 350117, China
Department of Computer Science and Information Engineering, Notional Chung Cheng University[5]
160 San-Hsing, Ming-Hsiung, Chiayi 621,Taiwan
*(Received July 14, 2014; revised and accepted Jan. 16 & June 4, 2015)*

## Abstract

The term "proxy multi-signature" refers to the situation in which a proxy signer is authorized to sign a message on behalf of a group of original signers. Combined with identity-based cryptography, we proposed an efficient identity-based proxy multi-signature scheme using cubic residues without bilinear pairing. Our scheme is secure against existential forgery on adaptive chosen-message and identity attacks under the hardness of integer factorization assumption. Compared with elliptic curve or bilinear pairing, the integer factorization assumption is more reliable and easier to use because it has been developed 2500 years ago. Furthermore, our scheme is more efficient than previous schemes based on bilinear pairing.
*Keywords: Cubic residues, identity-based signature, integer factorization, proxy multi-signature, random oracle model*

## 1 Introduction

Shamir [15] introduced identity-based cryptography in 1984 in order to simplify the key-management procedure of traditional, certificate-based, public-key infrastructures. Shamir's approach allowed an entity's public key to be derived directly from her or his identity, such as an email address, and the entity's private key can be generated by a trusted third party which is called the private key generator (PKG).

The notion of proxy signatures was proposed by Mambo et al. [10] in 1996. They identified the signers into two entities, i.e., the original signer and the proxy signer. The latter can sign a message on behalf of the former with a warrant the former delegated. Proxy signatures have many practical applications, such as distributed systems, grid computing, mobile agent applications, distributed shared object systems, global distribution networks, and mobile communications [2]. Since 1996, the proxy signature has been paid significant attention [7] and various extensions of the proxy signature have been proposed [1, 9, 11, 19, 22], one of which is the proxy multi-signature [9, 19, 22].

In 2000, Yi et al. proposed the proxy multi-signature [22] in which a designated proxy signer can generate a valid signature on behalf of a group of original signers. Proxy multi-signature can be used in the following scenario, i.e., a university wants to release a document that several departments may be involved, for example, the Deans Office, the Student Affairs Office, and the Human Resources Department, etc.. The document must be signed by all of the above entities or by a proxy signer delegated by those entities. Combined with identity-based cryptography, Li and Chen [9] proposed the notion of identity-based proxy multi-signature (IBPMS) and constructed a scheme using bilinear pairings in 2005. However, most existing IBPMS schemes were based on bilinear pairing [4, 9, 14, 20], which required more computational cost than normal operations, such as modular exponentiations in finite fields. Therefore, there was a strong interest in determining how to construct a secure scheme without pairing. In 2011, Tiwari and Padhye [18] proposed a secure IBPMS scheme based on the elliptic curve discrete logarithm problem. Although they claimed that their scheme was more efficient and had a smaller key size than pairing-based schemes, the security on which their method was based on the elliptic curve discrete logarithm problem assumption which was only a few decades old [6].

In this paper, we propose a new identity-based proxy multi-signature (IBPMS) scheme using cubic residues without bilinear pairing. The security of our method is based on the integer factorization assumption which is 2500 years old. We briefly introduce our contributions. First, our scheme is the first identity-based proxy multi-signature scheme using the cubic residues problem. Second, our scheme has been proven to be secure in the random oracle model under the hardness of integer factorization problem assumption. Third, our scheme is made more efficient than Cao and Cao's IBPMS scheme [4] based on bilinear pairing.

The rest of the paper is organized as follows. In Section 2, we introduce the cubic residues problem and integer factorization problem assumption. In Section 3, we give the formal definition and security model of identity-based proxy multi-signature. In Section 4, we propose a new identity-based proxy multi-signature scheme using cubic residues. In Section 5, we give the formal security proof for the proposed scheme under the random oracle model. In Section 6, we compare the efficiency and performance of our scheme with Cao and Cao's IBPMS scheme. Finally, we present our conclusions in Section 7.

# 2 Preliminaries

In this section, we review cubic residues and the method of their construction mentioned in [21] and integer factorization problem assumption

## 2.1 Cubic Residues

**Definition 1.** *For a positive integer $n$, if there is some $x$ that satisfies the expression $x^3 \equiv C \pmod n$, we say that $C$ is a cubic residue modulo $n$, and $x$ is called the cubic root of $C$ modulo $n$.*

From [21], we have Lemma 1, Theorem 1, and Theorem 2.

**Lemma 1.** *Let $p$ be a prime number, $3_p = gcd(3, p-1)$, and $C \in Z_p^*$. We say that $C$ is a cubic residue modulo $p$ if and only if $C^{\frac{(p-1)}{3_p}} \pmod p \equiv 1$.*

Obviously, if $p$ is prime number and $p \equiv 2 \pmod 3$, then every $C \in Z_p^*$ is a cubic residue modulo $p$.

If $q$ is prime number, and $q \equiv 4$ or $7 \pmod 9$, for every $h \in Z_p^*$, we can construct a cubic residue modulo $q$ as follows.

Let $a$ be a non-cubic modulo $q$, we compute $\eta = [(q-1) \pmod 9]/3$, $\lambda = \eta \pmod 2 + 1$, $\beta = (q-1)/3$, $\xi = a^{\eta \cdot \beta} \pmod q$, $\tau \equiv h^{\lambda \cdot \beta} \pmod q$, and

$$b = \begin{cases} 0, & if\ \tau = 1 \\ 1, & if\ \tau = \xi \\ 2, & if\ \tau = \xi^2, \end{cases}$$

then $C = a^b \cdot h$ is a cubic residue modulo $q$.

**Theorem 1.** *Let $p$, $q$ be as mentioned above and $n = p \cdot q$. Then $C = a^b \cdot h$ is a cubic residue modulo $n$, and $s \equiv C^{[2^{\eta-1}(p-1)(q-1)-3]/9} \pmod n$ is a cubic root of $C^{-1}$.*

**Theorem 2.** *Let $n = p \cdot q$. If there is $s_1^3 \equiv s_2^3 \equiv C \pmod n$, and $s_1 \not\equiv s_2 \pmod n$, then $gcd(s_1 - s_2, n)$ is a non-trivial divisor of $n$.*

## 2.2 Integer Factorization Problem Assumption

The integer factorization problem assumption is one of the fundamental hardness problems, which has been studied extensively and used to construct cryptographic schemes. We will analyze the security of our proposed scheme based on this assumption. From [23], we have Definition 2 and Definition 3.

**Definition 2.** *Given $n = p \cdot q$, where $p$ and $q$ are prime numbers and they are unknown publicly, the integer factorization problem is defined to output a prime number $p (1 < p < n)$ such that $p$ can divide $n$.*

**Definition 3 (Integer factorization problem assumption).** *The integer factorization problem (IFP) is a $(t', \epsilon')$-hard assumption, if there is no polynomial time algorithm in time at most $t'$, can solve the integer factorization problem with probability at least $\epsilon'$.*

# 3 Formal Definition and Security Model

We give a formal definition and security model of the identity-based proxy multi-signature scheme based on the works of Cao and Cao [4], Singh and Verma [16], and Sun et al. [17].

## 3.1 Formal Definition of the Identity-based Proxy Multi-signature Scheme

In an identity-based proxy multi-signature scheme, there are two entities named as a group of the original signers and the proxy signer. We use $ID_i$, for $i = 1, 2, \cdots, n$, to denote the identity of original signer $i$, and $ID_{ps}$ to denote the identity of the proxy signer. From [4], we have Definition 4.

**Definition 4.** *An identity-based proxy multi-signature scheme (IBPMS) is a tuple of seven algorithms as IBPMS=(**Setup**, **Extract**, **DelGen**, **DelVeri**, **PMKGen**, **PMSign**, **PMVeri**).*

**Setup.** PKG takes a security parameter as input, and outputs public parameter $PP$ and its master key $MK$.

**Extract.** PKG takes its master key $MK$ and a user's identity $ID_i$ as inputs, and outputs the user's public key and secret key pair $(H_{ID_i}, s_{ID_i})$.

**DelGen.** For $i = 1, 2, \cdots, n$, the original signer $i$ takes her or his secret key $s_{ID_i}$ and a warrant $w$ as inputs, and outputs her or his delegation $D_{i \to ps}$ to the proxy signer.

**DelVeri.** For $i = 1, 2, \cdots, n$, the proxy signer takes delegation $D_{i \to ps}$ from the original signer $i$ and her or his identity $ID_i$ as inputs, and verifies whether or not the delegation is valid.

**PMKGen.** The proxy signer takes her or his secret key $s_{ID_{ps}}$ and delegations $D_{i \to ps}$, $i = 1, 2, \cdots, n$, as inputs, and generates her or his private signing key $sk_{ps}$.

**PMSign.** The proxy signer takes her or his signing key $sk_{ps}$, message $m$, and delegations $D_{i \to ps}$, $i = 1, 2, \cdots, n$, as inputs, and generates the proxy multi-signature $\sigma$ of the message $m$.

**PMVeri.** The verifier takes the proxy multi-signature $\sigma$ and the original signers' identities, $ID_i$, $i = 1, 2, \cdots, n$, and the proxy signer's identity $ID_{ps}$ as inputs, and verifies whether or not the proxy multi-signature is valid.

## 3.2 Security Model

Compared with Cao and Cao's method [4], and Sun et al.'s method [17], we use the security model of the proxy multi-signature which is described in [17]. And, we extend Sun et al.'s model into an identity-based proxy multi-signature to prove the security of our scheme. The adversaries in their model can be classified into three types as follows:

**Type 1.** The adversary, $A_1$, knows nothing except the identities of the original signers and the proxy signer.

**Type 2.** The adversary, $A_2$, knows the secret keys of $n-1$ original signers and proxy signer in addition to what $A_1$ knows in Type 1.

**Type 3.** The adversary, $A_3$, knows the secret keys of all of the original signers in addition to what $A_1$ knows in Type 1, but does not know the secret key of the proxy signer.

Obviously, if an adversary in Type 1 can forge a valid signature of the scheme, the adversary in Type 2 or Type 3 also can forge a valid signature. So, we only consider the Type 2 and Type 3 adversaries in this paper.

With regard to the Type 2 adversary $A_2$, we can assume that she or he has all of the secret keys of the $n-1$ original signers, except for signer $n$. If she or he has a valid delegation, $D_{n \to ps}$, she or he can output a valid proxy multi-signature herself or himself with the secret keys of the other original signers and proxy signer. So, the objective of the Type 2 adversary is to output a valid delegation, $D_{n \to ps}$.

With regard to the Type 3 adversary $A_3$, since she or he has all of the secret keys of the original signers, she or he can output a valid delegation $D_{i \to ps}$, $i = 1, 2, \cdots, n$, herself or himself. So, the objective of the Type 3 adversary is to output a valid proxy multi-signature under delegations $D_{i \to ps}$, $i = 1, 2, \cdots, n$.

Let an adversary $A_t (t = 2 \text{ or } 3)$ be a probabilistic Turing machine, $A_t$ takes public parameter $PP$ and a random tape as inputs and performs an experiment with the algorithm $B$. Inspired from [17], we define the following two definitions.

**Definition 5.** *For an identity-based proxy multi-signature scheme, we define an experiment of the adversary $A_t (t = 2$ or $3)$ with the security parameter $\lambda$ as follows:*

**Step 1.** *Algorithm $B$ runs the Setup algorithm and returns public parameter $PP$ to the adversary $A_t$.*

**Step 2.** *$B$ maintains several lists, e.g., $E_{list}$, $D_{list}$, $S_{list}$, and initializes them as null.*

**Step 3.** *When the adversary $A_t$ makes adaptive queries from the algorithm $B$, $B$ maintains several oracles and answers as follows:*

- ***Extract oracle:*** *The oracle takes a user's identity $ID_i$ as input, returns her or his private key $s_{ID_i}$, and puts the tuple $(ID_i, s_{ID_i})$ into $E_{list}$.*
- ***DelGen oracle:*** *The oracle takes the original signer's identity $ID_i$ and the warrant $w$ as inputs, returns the delegation $D_{i \to ps}$, and puts the tuple $(ID_i, w, D_{i \to ps})$ into $D_{list}$.*
- ***PMSign oracle:*** *The oracle takes the message $m$ and the delegations $D_{i \to ps}$, $i = 1, 2, \cdots, n$ as inputs, returns a proxy multi-signature $\sigma$ signed by the proxy signer and puts the tuple $(m, w, \sigma)$ into $S_{list}$.*

**Step 4.** *Eventually, $A_t$ outputs a forgery.*

- *If $t = 2$, then it is the Type 2 adversary $A_2$. The forgery is of the tuple $(ID_n, w, D_{n \to ps})$, and $(ID_n, w, D_{n \to ps})$ is valid delegation of $ID_n$ with warrant $w$, and $ID_n \notin E_{list}$, $(ID_n, w) \notin D_{list}$.*
- *If $t = 3$, then it is the Type 3 adversary $A_3$. The forgery is of the tuple $(m, w, \sigma)$, and $(m, w, \sigma)$ is a valid proxy multi-signature, and $ID_p \notin E_{list}$, $(w, m) \notin S_{list}$.*

*If the output satisfies one of the above two items, $A_t$'s attack was successful.*

**Definition 6.** *For any polynomial adversary $A_t$ ($t = 2$ or $3$), if the probability of $A_t$'s success in the above experiment is negligible, then, the identity-based proxy multi-signature scheme is said to be secure against existential forgery on adaptive chosen-message and identity attacks.*

# 4 Our Proposed IBPMS Scheme

In this section, we describe a new identity-based proxy multi-signature scheme. We designed our scheme, which extends the identity-based signature [21], based on the cubic residues. The proposed scheme includes the following seven algorithms:

**Setup.** Given the security parameters $k$ and $l$, PKG carries out the algorithm and returns public parameters $PP$ and master key $MK$ as follows:

1) Randomly generates two $k$-bits prime numbers $p$ and $q$, satisfying $p \equiv 2 \pmod 3$ and $q \equiv 4$ or $7 \pmod 9$, respectively; then computes $n = p \cdot q$.
2) Computes $d = [2^{\eta-1}(p-1)(q-1)-3]/9$, $\eta = [(q-1) \pmod 9]/3$, $\lambda = \eta \pmod 2 + 1$, $\beta = (q-1)/3$.
3) Randomly selects a non-cubic residue $a$ modulo $q$ and computes $\xi \equiv a^{\eta \cdot \beta} \pmod q$.
4) Selects four hash functions $H_1 : \{0,1\}^* \to Z_n^*$,$H_2$, $H_3$ ,$H_4 : \{0,1\}^* \to \{0,1\}^l$.

PKG publishes $(n, a, \eta, \lambda, H_1, H_2, H_3, H_4)$ as the public parameter $PP$ and keeps $(p, q, d, \beta)$ secret as the master key $MK$.

**Extract.** Given public parameter $PP$, the master key $MK$, and identity $ID_i$ of user $i$, for $i = 1, 2, \cdots, n$, PKG computes the corresponding secret key as follows:

1) Computes $\tau_i \equiv H_1(ID_i)^{\lambda \cdot \beta} \pmod q$.
2) Computes $b_i = \begin{cases} 0, & if\ \tau_i = 1 \\ 1, & if\ \tau_i = \xi \\ 2, & if\ \tau_i = \xi^2 \end{cases}$, and $C_i = a^{b_i} \cdot H_1(ID_i) \pmod n$, $s_{ID_i} \equiv (C_i)^d \pmod n$.

PKG transmits secret key $(s_{ID_i}, b_i)$, for $i = 1, 2, \cdots, n$ to user $i$ via a secure channel.

**DelGen.** Let $ID_i$, for $i = 1, 2, \cdots, n$, be the identity of the original signer $i$, and $ID_{ps}$ be the identity of the proxy signer. The original signer $i$, for $i = 1, 2, \cdots, n$, wants to delegate the proxy signer to get a warrant $w$ of message $m$, so she or he takes her or his secret key $(s_{ID_i}, b_i)$, and warrant $w$ as inputs and outputs the delegation $D_{i \to ps}$. Then, the original signer $i$, for $i = 1, 2, \cdots, n$, continues as follows:

1) Randomly selects $r_i \in Z_n^*$, computes $R_i \equiv r_i^3 \pmod n$, and broadcasts $R_i$ to the other original signers.
2) Computes $R \equiv \prod_{i=1}^n R_i \pmod n$, $h_w = H_2(w,\ R)$, $V_i \equiv r_i \cdot s_{ID_i}^{h_w} \pmod n$.

Each original signer $i$ sends her or his delegation $D_{i\to ps} = (ID_i, b_i, w, R_i, V_i)$ to the proxy signer.

**DelVeri.** To verify each delegation $D_{i\to ps}$ with warrant $w$, the proxy signer computes $R \equiv \prod_{i=1}^n R_i \pmod n$, $h_w = H_2(w,\ R)$, $C_i \equiv a^{b_i} \cdot H_1(ID_i) \pmod n$, and checks $V_i^3 \cdot {C_i}^{h_w} \equiv R_i \pmod n$ for $i = 1, 2, \cdots, n$. If the equation holds, she or he accepts $D_{i\to ps}$ as a valid delegation; otherwise, it is rejected.

**PMKGen.** If the proxy signer accepts all delegations $D_{i\to ps}$, for $i = 1, 2, \cdots, n$, she or he computes $h_{ps} = H_3(ID_{ps}, w, R)$, $V \equiv \prod_{i=1}^n V_i \pmod n$, $sk_{ps} \equiv s_{ID_{ps}}^{h_{ps}} \cdot V \pmod n$ and takes $sk_{ps}$ as her or his private signing key.

**PMSign.** The proxy signer takes $sk_{ps}$ as input and randomly selects $r_{ps} \in Z_n^*$, computes $R_{ps} \equiv r_{ps}^3 \pmod n$, $h_m = H_4(ID_{ps}, w, m, R_{ps})$, $V_{ps} \equiv r_{ps} \cdot sk_{ps}^{h_m} \pmod n$. The tuple $(ID_1, ID_2, \cdots, ID_n, ID_{ps}, b_1, b_2, \cdots, b_n, b_{ps}, m, w, R, R_{ps}, V_{ps})$ is the proxy signature of message $m$ on behalf of all original signers $i$, for $i = 1, 2, \cdots, n$.

**PMVeri.** In order to verify the proxy multi-signature $(ID_1, ID_2, \cdots, ID_n, ID_{ps}, b_1, b_2, \cdots, b_n, b_{ps}, m, w, R, R_{ps}, V_{ps})$ of message $m$ under warrant $w$, the verifier conducts the following: computes $h_{ps} = H_3(ID_{ps}, w, R)$, $h_w = H_2(w, R)$ , $h_m = H_4(ID_{ps}, m, w, R_{ps})$, $C \equiv \prod_{i=1}^n (a^{b_i} \cdot H_1(ID_i)) \pmod n$ , $C_{ps} \equiv a^{b_{ps}} \cdot H_1(ID_{ps}) \pmod n$, then checks $V_{ps}^3 \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w \cdot h_m} \equiv R_{Ps} \cdot R^{h_m} \pmod n$; if the equation holds, then she or he accepts it; otherwise, it is rejected.

Our scheme is correct because the following equation holds:

$$\begin{aligned}
& V_{ps}^3 \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & (r_{ps} \cdot sk_{ps}^{h_m})^3 \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & (r_{ps} \cdot (d_{ID_{ps}}^{h_{ps}} \cdot V)^{h_m})^3 \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & (r_{ps} \cdot (d_{ID_{ps}}^{h_{ps}} \cdot \textstyle\prod_{i=1}^n r_i \cdot s_{ID_i}^{h_w})^{h_m})^3 \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & r_{ps}^3 \cdot ((d_{ID_{ps}}^3)^{h_{ps}} \cdot \textstyle\prod_{i=1}^n r_i^3 \cdot \prod_{i=1}^n (s_{ID_i}^3)^{h_w})^{h_m} \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & r_{ps}^3 \cdot ((d_{ID_{ps}}^3)^{h_{ps}} \cdot \textstyle\prod_{i=1}^n r_i^3 \cdot \prod_{i=1}^n (s_{ID_i}^3)^{h_w})^{h_m} \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & R_{ps} \cdot ({C_{ps}}^{-h_{ps}} \cdot R \cdot \textstyle\prod_{i=1}^n {C_i}^{-h_w})^{h_m} \cdot {C_{ps}}^{h_{ps}\cdot h_m} \cdot C^{h_w\cdot h_m} \\
\equiv\ & R_{ps} \cdot R^{h_m} \pmod n.
\end{aligned}$$

# 5 Security Proof of Our Proposed Scheme

In this section, we give the security proof of our proposed scheme. We show that our scheme is secure against existential forgery under adaptive chosen-message and identity attacks in the random oracle model. We prove our scheme against Type 2 adversaries and Type 3 adversaries, respectively.

If a Type 2 adversary $A_2$ has the ability to break our scheme, we can construct a polynomial time algorithm $B$, by interacting with $A_2$, to solve the integer factorization problem.

**Theorem 3.** *Given a pair of security parameters $(k,l)$, if the integer factorization problem is $(t^{'},\epsilon^{'})$-hard, then our identity-based proxy multi-signature scheme is $(t,q_{H_2},q_D,\epsilon_2)$-secure against existential forgery under adaptive chosen-message and identity attacks for the Type 2 adversary $A_2$, which satisfies:*

$$\epsilon^{'} \geq \frac{4}{9}\cdot\left(\frac{(\epsilon_2-\delta_2)^2}{q_{H_2}+1}-\frac{\epsilon_2-\delta_2}{2^l}\right),$$

$$t^{'} = 2t + O\left(k^2\cdot l + k^3\right),$$

*where $q_{H_2}$ and $q_D$ denote the number of queries that $A_2$ can ask to the random oracle $H_2$ and DelGen oracle, respectively, and $\delta_2 = \frac{q_D\cdot(q_{H_2}+q_D)}{3\cdot 2^k}$.*

*Proof.* Assuming that adversary $A_2$ breaks the proposed scheme, we can construct an algorithm $B$ to resolve the integer factorization problem.

Given an integer $n = p\cdot q$ (for some unknown $p$ and $q$), and a non-cubic residue $a \pmod n$, we will design an algorithm $B$ to output $p$ and $q$ with non-negligible probability.

**Step 1.** Algorithm $B$ sends $(n,a)$ to adversary $A_2$ as public parameters.

**Step 2.** $B$ maintains several lists, i.e., $H_{1,list}$, $H_{2,list}$,$E_{list}$, and $D_{list}$ and initializes them as null.

**Step 3.** $B$ responds to $A_2$'s queries as follows:

- $H_1$**-oracle:** $A_2$ requests $H_1$ on $ID_i$, and $B$ checks if $ID_i$ existed in $H_{1,list}$. If not, $B$ picks a random $s_i \in Z_n^*$ and $b_i \in \{0,1,2\}$, computes $h_{1,i} = H_1(ID_i) \equiv \frac{s_i^3}{a^{b_i}} \pmod n$, and adds the tuple $(ID_i,h_{1,i},s_i,b_i)$ into $H_{1,list}$; then, $B$ returns $h_{1,i}$ to $A_2$.
- $H_2$**-oracle:** $A_2$ requests $H_2$ on $(w,R)$, and $B$ checks if $(w,R)$ existed in $H_{2,list}$. If not, $B$ picks a random $e \in \{0,1\}^l$, adds the tuple $(w,R,e)$ into $H_{2,list}$, then, $B$ returns $e$ to $A_2$.
- **Extract oracle:** $A_2$ requests Extract algorithm on $ID_i$ , and $B$ checks if $ID_i$ existed in $E_{list}$. If not, $B$ returns to $H_1$-oracle and gets $(ID_i,h_{1,i},s_i,b_i)$ of $H_{1,list}$; then, $B$ returns $(s_i,b_i)$ to $A_2$ and adds the tuple $(ID_i,s_i,b_i)$ into $E_{list}$.
- **DelGen oracle:** $A_2$ requests delegation on $(ID_n,w)$. According to the assumption, $A_2$ has the secret keys of the original signers $i$, $i = 1,2,\cdots,n-1$, by requesting Extract oracle. For $i = 1,2,\cdots,n-1$, $A_2$ randomly selects $r_i \in Z_n^*$, computes $R_i \equiv r_i^3 \pmod n$, and sends $R_i$, where $i = 1,2,\cdots,n-1$, to $B$. $B$ randomly selects $V_n,\tau \in \{0,1\}^l$, computes $R_n \equiv V_n^3\cdot (a^{b_n}\cdot H_1(ID_n))^{\tau} \pmod n$, and $R \equiv \prod_{i=1}^{n} R_i \pmod n$; if $R$ already exists in $H_{2,list}$, failure is returned; else $(ID_n,b_n,w,R_n,V_n)$ is returned as the original signer $n$'s delegation to $A_2$; also, $\tau$ is returned for the sake of helping $A_2$ completing the delegation on $(ID_i,w)$ for $i = 1,2,\cdots,n-1$. $B$ adds the tuple $(ID_n,b_n,w,R_n,V_n)$ into $D_{list}$ and adds $(w,R,\tau)$ into $H_{2,list}$.

**Step 4.** $A_2$ outputs a delegation forgery of warrant $w^*$ and $ID_n^*$ with $D_{n\to ps}^* = (ID_n^*,b_n^*,w^*,R_n^*,V_n^*)$, which $(ID_n^*,w^*)$ is not requested on the DelGen oracle, and $ID_n^*$ is not requested on the Extract oracle.

**Step 5.** Finally, we will show how $B$ resolves the integer factorization problem with $A_2$'s delegation forgery.

We apply the oracle replay technique describes in Forking Lemma [12, 13] to factor $n$, i.e., $B$ resets $A_2$ two times. For the first time, $B$ records all the transcripts that interacted with $A_2$. For the second time, $B$ starts with the first time random tape and returns the same answers to $A_2$, except $H_2$-oracle. Each time, when $A_2$ asks $H_2$-oracle, $B$ chooses different random numbers, $e^*$,$e^{**}$, as the answer, respectively.

After two rounds of interacting with $B$, $A_2$ forges two delegations $(ID_n^*,b_n^*,w^*,R_n^*,V_n^*)$, $(ID_n^*,b_n^*,w^*,R_n^*,V_n^{**})$, together with delegations of original signers $1,2,\cdots,n-1$, sends them to $B$. Then, $B$ executes as follows:

- $B$ computes $R^* \equiv \prod_{i=1}^{n} R_i^* \pmod n$, returns to the previous three records of $H_{2,list}$ lists for $(w^*,R^*)$, obtains, $e^*$, $e^{**}$, and checks whether or not they satisfy $(e^*-e^{**}) \equiv 0 \pmod 3$; if so, then $B$ aborts it.
- Else $B$ can obtain$(V_n^*)^3\cdot(C_n^*)^{e^*} = R_n^*$, $(V_n^{**})^3\cdot(C_n^*)^{e^{**}} \equiv R_n^* \pmod n$, where $C_n^* \equiv a^{b_n^*}\cdot H_1(ID_n^*) \pmod n$.
- $B$ obtains $(V_n^*/V_n^{**})^3 \equiv (C_n^*)^{e^{**}-e^*} \pmod n$.
- If $(e^{**}-e^*) \equiv 1 \pmod 3$, there is some $x \in Z_p^*$ satisfies the equation $(e^{**}-e^*) = 3x+1$. So we obtain $(V_n^*/V_n^{**})^3 \equiv (C_n^*)^{3x+1} \pmod n$, and therefore $C_n^* \equiv \left(\frac{V_n^*}{V_n^{**}\cdot(C_n^*)^x}\right)^3 \pmod n$.

- If $(e^{**} - e^{*}) \equiv 2 \pmod 3$, there is some $x \in Z_p^*$ satisfies the equation $(e^{**} - e^{*}) = 3x - 1$. So we obtain $(V_n^*/V_n^{**})^3 \equiv (C_n^*)^{3x-1} \pmod n$, and therefore $C_n^* \equiv \left(\frac{V_n^{**} \cdot (C_n^*)^x}{V_n^*}\right)^3 \pmod n$.

Then, if $(e^{**} - e^{*}) \not\equiv 0 \pmod 3$, $B$ obtains the cubic root of $C_n^*$. And $B$ can look up the list $H_{1,list}$ and obtain another cubic root of $C_n^*$. Then, $B$ obtains two cubic roots of $C_n^*$. If the two cubic roots are not equal, $B$ can factor $n$ according to Theorem 2.

Since $e^*, e^{**}$ are picked randomly, the probability of $(e^{**} - e^{*}) \not\equiv 0 \pmod 3$ is $\frac{2}{3}$, and the probability that the two cubic roots of $C_n^*$ are inequal is $\frac{2}{3}$.

Next, we will analyze the probability of $A_2$ successfully forging two valid delegations similar to [3].

Let $\epsilon_2^*$ denote the probability of $A_2$ forging a delegation in a single run, and $\epsilon_2$ denote the probability of $A_2$ forging a delegation in the real attack.

In $H_{2,list}$, all the records $(w, R, e)$ are filled by $H_2$-oracle query and DelGen oracle query. So there are, at most $q_{H_2} + q_D$, different $R$'s. For every DelGen oracle, $B$ randomly selects $V_n, \tau \in \{0,1\}^l$, computes $R_n = V_n^3 \cdot (a^{b_n} \cdot H_1(ID_n))^{\tau}$ and $R = \prod_{i=1}^{n} R_i$, therefore, $R$ can be considered as the random cubic residue modulo $n$. Obviously, the number of elements in cubic residues modulo $n$ is $(3 \cdot 2^k)$. So the probability that $R$ is in the $H_{2,list}$ is, at most $\frac{q_{H_2}+q_D}{3 \cdot 2^k}$. So the probability of $A_2$ forging a delegation in a single run is $\epsilon_2^* \geq \epsilon_2 - \frac{q_D \cdot (q_{H_2}+q_D)}{3 \cdot 2^k}$.

Let $p_i$ denote the probability of forgery based on the $i^{th} H_2$-oracle query in a single run; then

$$\epsilon_2^* = \sum_{i=1}^{q_{H_2}+1} p_i.$$

Let $p_{i,s}$ denote the probability of forgery together based on $i^{th} H_2$-oracle query with input $s$, where $s$ is a specific random tape input of length $m$. Then

$$2^m \cdot p_i = \sum_{s \in \{0,1\}^m} p_{i,s}.$$

For a specific random tape $s$, since twice valid forgery need different outputs of $H_2$-oracle query, the probability of twice forgery based on the same $i^{th} H_2$-oracle query is $p_{i,s} \cdot (p_{i,s} - 2^{-l})$. Let $P_i$ denote the probability of twice forgery based on the same $i^{th} H_2$-oracle query in two runs; then

$$P_i = \sum_{s \in \{0,1\}^m} 2^{-m} \cdot p_{i,s} \cdot (p_{i,s} - 2^{-l}) \geq p_i^2 - 2^{-l} \cdot p_i.$$

So, the probability of twice forgery based on the same $H_2$-oracle query in two runs is $\sum_{i=1}^{q_{H_2}+1} P_i$. We have

$$\sum_{i=1}^{q_{H_2}+1} P_i \geq \sum_{i=1}^{q_{H_2}+1} p_i^2 - \sum_{i=1}^{q_{H_2}+1} 2^{-l} \cdot p_i \geq \frac{(\epsilon_2^*)^2}{q_{H_2}+1} - \frac{\epsilon_2^*}{2^l}$$

$$\geq \frac{\left(\epsilon_2 - \frac{q_D \cdot (q_{H_2}+q_D)}{(3 \cdot 2^k)}\right)^2}{q_{H_2}+1} - \frac{\epsilon_2 - \frac{q_D \cdot (q_{H_2}+q_D)}{(3 \cdot 2^k)}}{2^l}.$$

Taking $(e^{**} - e^{*}) \not\equiv 0 \pmod 3$ and the difference of the two cubic roots of $C_n^*$ into account, the probability of factoring $n$ is $\epsilon' \geq \frac{4}{9} \sum_{i=1}^{q_{H_2}+1} P_i \geq \frac{4}{9} \cdot \left(\frac{(\epsilon_2 - \delta_2)^2}{q_{H_2}+1} - \frac{\epsilon_2 - \delta_2}{2^l}\right)$, where $\delta_2 = \frac{q_D \cdot (q_{H_2}+q_D)}{3 \cdot 2^k}$. So, the theorem is proved. □

As to the running time, according to [3], $B$ has to run $A_2$ twice and perform some other operations to factor $n$. So $B$ should spend the time $t' = 2t + O(k^2 \cdot l + k^3)$ to factor $n$.

**Theorem 4.** *Given a security parameter $(k, l)$, if the integer factorization problem is $(t', \epsilon')$-hard, then our identity-based proxy multi-signature scheme is $(t, q_{H_4}, q_S, \epsilon_3)$-secure against existential forgery under adaptive chosen-message and identity attacks for the Type 3 adversary $A_3$, which satisfies:*

$$\epsilon' \geq \frac{4}{9} \cdot \left(\frac{(\epsilon_3 - \delta_3)^2}{q_{H_4}+1} - 2^{-l} \cdot (\epsilon_3 - \delta_3)\right)$$

$$t' = 2t + O\left(k^2 \cdot l + k^3\right),$$

*where $q_{H_4}$ and $q_S$ denote the number of queries that $A_3$ can ask to the random oracle $H_4$ and PMSign, respectively, and $\delta_3 = \frac{q_S \cdot (q_{H_4}+q_S)}{3 \cdot 2^k}$.*

*Proof.* This proof is similar to that of Theorem 3. So, we just describe the main difference with Theorem 3 as follows:

**Step 1.** Algorithm $B$ does the same as Step 1 of Theorem 3.

**Step 2.** $B$ deletes $D_{list}$ list and adds $H_{3,list}$, $H_{4,list}$, $S_{list}$ lists, and initializes them as null.

**Step 3.** $B$ deletes DelGen oracle and adds $H_3$, $H_4$ and PMSign oracle accordingly.

- **$H_3$-oracle:** $A_3$ requests $H_3$ on $(ID_{ps}, w, R)$, $B$ checks if $(ID_{ps}, w, R)$ existed in $H_{3,list}$. If not, $B$ picks a random $\mu \in \{0,1\}^l$ and adds the tuple $(ID_{ps}, w, R, \mu)$ into $H_{3,list}$; then $B$ returns $H_3(ID_{ps}, w, R) = \mu$ to $A_3$.
- **$H_4$-oracle:** $A_3$ requests $H_4$ on $(ID_{ps}, w, m, R_{ps})$, and $B$ checks if $(ID_{ps}, w, m, R_{ps})$ existed in $H_{4,list}$. If not, $B$ picks a random $\eta \in \{0,1\}^l$ and adds the tuple $(ID_{ps}, w, m, R_{ps}, \eta)$ into $H_{4,list}$; then, $B$ returns $H_4(ID_{ps}, w, m, R_{ps}) = \eta$ to $A_3$.
- **PMSign oracle:** $A_3$ requests PMSign algorithm on $(w, m)$. $A_3$ randomly selects $r_i \in Z_n^*$ and computes $R_i = r_i^3 \pmod n$, $R = \prod_{i=1}^{n} R_i \pmod n$, and requests $H_2$-oracle query and obtains $H_2(w, R) = e$. Since $A_3$ knows all the

Table 1: Comparison of security

| Scheme | Security Proof Method | Mathematics Tool | Assumption* |
|---|---|---|---|
| *Cao and Cao [4]* | Random oracle | bilinear pairings | CDH |
| *Our scheme* | Random oracle | Cubic residues | IFP |

*CDH stands for computational Diffie-Hellman assumption, and IFP stands for integer factorization problem.

Table 2: Comparison with other schemes

| Scheme | Extract | DelGen | DelVeri | PMKGen | PMSign | PMVeri | Total | Total Time (ms) |
|---|---|---|---|---|---|---|---|---|
| *Cao and Cao [4]* | $1M_p$ $+1H_M$ | $2M_p$ $+1H_M$ | $2H_M$ $+3O_P$ | $1M_p$ | $2M_p$ $+1H_M$ | $1M_p$ $+3H_M$ $+4O_P$ | $7M_p$ $+8H_M$ $+7O_P$ | 209.26 |
| *Our scheme* | $1E_n$ | $1E_n$ | $1E_n$ | $1E_n$ | $1E_n$ | $3E_n$ | $8E_n$ | 42.48 |

Table 3: Cryptographic running time (ms)

| Modular Exponentiation | Pairing | Pairing-based Scalar Multiplication | Map-to-point Hash |
|---|---|---|---|
| 5.31 | 20.04 | 6.38 | 3.04 |

secret keys of original signers, $A_3$ can compute $V_i \equiv r_i \cdot s_{ID_i}^e \pmod n$ and obtain all the delegation $D_{i\to ps} = (ID_i, b_i, w, R_i, V_i)$, $i = 1, 2, \cdots, n$. $A_3$ sends $D_{i\to ps}$, $i = 1, 2, \cdots, n$, to $B$ to request PMSign algorithm on $(w, m)$. $B$ computes $R \equiv \prod_{i=1}^{n} R_i \pmod n$ and obtains $H_3(ID_{ps}, w, R) = \mu$ by looking up the list $H_{3,list}$ - in $H_3$-oracle. $B$ picks random $V_p$, $\varsigma \in \{0,1\}^l$, and computes $C \equiv \prod_{i=1}^{n} (a^{b_i} \cdot h_{1,i}) \pmod n$, $C_{ps} \equiv a^{b_p} \cdot h_{1,ps} \pmod n$, $V = \prod_{i=1}^{n} V_i$, $R_{ps} \equiv V_{ps}^3 \cdot ((C_{ps})^{\mu} \cdot C^{\eta}/R)^{\varsigma} \pmod n$. If $R_{ps}$ already exists in $H_{4,list}$, $B$ returns failure, else returns $(ID_1, ID_2, \cdots, ID_n, ID_p, b_1, b_2, \cdots, b_n, b_p, m, w, R, R_p, V_p)$ as proxy multi-signature of $(w, m)$ to $A_3$. $B$ adds the tuple $(ID_1, ID_2, \cdots, ID_n, ID_p, b_1, b_2, \cdots, b_n, b_p, m, w, R, R_p, V_p)$ into $S_{list}$, and adds $(ID_{ps}, w, m, R_{ps}, \varsigma)$ into $H_{4,list}$.

**Step 4.** $A_3$ outputs a proxy multi-signature forgery of $(w, m)$ with $\sigma^* = (ID_1^*, ID_2^*, \cdots, ID_n^*, ID_{ps}^*, b_1^*, b_2^*, \cdots, b_n^*, b_{ps}^*, m^*, w^*, R^*, R_{ps}^*, V_{ps}^*)$, which $ID_{ps}^*$ has not be requested on the Extract oracle, and $(m^*, w^*)$ has not be requested on the PMSign oracle.

**Step 5.** Similar with Theorem 3, $B$ resets $A_3$ twice with the same random tape, and gives the different random number until $A_3$ asks $H_4$-oracle. And $A_3$ can forge two proxy multi-signatures with the same value $R_{ps}$. $B$ can resolve integer factorization problem with $A_3$'s proxy multi-signature forgery.

As to the probability and running time, both of them are similar with Theorem 3. □

Furthermore, by Theorems 3 and 4, we can conclude Theorem 5 easily.

**Theorem 5.** *Given a security parameter $(k, l)$, if the factoring problem is $(t', \epsilon')$-hard, then our identity-based proxy multi-signature scheme is $(t, q_{H_2}, q_{H_4}, q_D, q_S, \epsilon)$-secure against existential forgery under adaptive chosen-message and identity attacks, which satisfies:*

$$\epsilon' \geq \frac{4}{9} \cdot \left( \frac{(\epsilon - \delta)^2}{2 \cdot \max\{q_{H_2} + 1, q_{H_4} + 1\}} - 2^{-l} \cdot (\epsilon - \delta) \right)$$

$$t' = 2t + O\left(k^2 \cdot l + k^3\right),$$

*where $\epsilon = \epsilon_2 + \epsilon_3$ and $\delta = \delta_2 + \delta_3$.*

We conclude that our scheme is secure against existential forgery under adaptive chosen-message and identity attacks under integer factorization problem assumption.

# 6 Comparison and Performance

In this section, we compare our scheme with Cao and Cao's IBPMS scheme [4]. The two schemes are provable security based on different hardness assumptions in the random oracle model. We describe them in detail in Table 1.

In order to simplify the complexity, we used the method of [5], which considers only a single original signer. Let $M_p, H_M, O_P, E_n$ denote one pairing-based scalar multiplication, map-to-point hash function, pairing operation, and modular exponentiation, respectively. In order to make our analysis clearer, we changed the total computation cost into running time in the last column of Table 2 according to Table 3, which is referred to reference [8].

According to Tables 1 and 2, our schemes total running time decreased drastically compared with Cao and Cao's scheme [4]. The security of our scheme is based on integer factorization problem assumption without bilinear pairing. We note that the integer factorization problem assumption is 2500 years old.

# 7 Conclusions

Identity-based proxy multi-signature has proposed for years, and several schemes have been proposed. However, most of the existing scheme is based on bilinear pairing or elliptic curve. In this paper, we propose an efficient identity-based proxy multi-signature scheme using cubic residues. The security of our scheme is based on the integer factorization problem assumption, which is more reliable and easier to use because it has been developed 2500 years ago. Our scheme is prove security against existential forgery under adaptive chosen-message and identity attacks. Furthermore, the efficiency of our scheme is higher than the existing scheme based on bilinear pairing such as Cao and Cao's scheme etc.

# Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

# References


[1] M. R. Asaar, M. Salmasizadeh, and W. Susilo, "An identity-based multi-proxy multi-signature scheme without bilinear pairings and its variant," *The Computer Journal*, vol. 58 , no. 4, pp. 1021–1039, 2015.

[2] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *Journal of Cryptology*, vol. 25, no. 1, pp. 57–115, 2012.

[3] Z. C. Cai, X. L. Dong, and Z. F. Cao, "Identity based signature scheme based on quadratic residues," *Science in China Series F: Information Sciences*, vol. 39, no. 2, pp. 199–204, 2009.

[4] F. Cao, and Z. F. Cao, "A secure identity-based proxy multi-signature scheme," *Information Sciences*, vol. 179, no. 3, pp. 292–302, 2009.

[5] X. F. Cao, and W. D. Kou, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[6] Cryptography Stack Exchange, *Why Is Elliptic Curve Cryptography Not Widely Used, Compared to RSA?*, Nov. 15, 2011. (http://crypto.stackexchange.com/questions/1190/why-is-elliptic-curve-cryptography-not-widely-used-compared-to-rsa).

[7] M. L. Das, A. Saxena, and D. B. Phata, "Algorithms and approaches of proxy signature: A survey," *International Journal of Network Security*, vol. 9, no. 3, pp. 264–284, 2009.

[8] D. B. He, J. H. Chen, and R. Zhang, "Efficient and provably-secure certificateless signature scheme without bilinear pairings," *International Journal of Communication Systems*, vol. 25, no. 11, pp. 1432–1442, 2012.

[9] X. X. Li, and K. F. Chen, "ID-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings," *Applied Mathematics and Computation*, vol. 169, no. 1, pp. 437–450, 2005.

[10] M. Mambo, K. Usuda, and E. Oamoto, "Proxy signatures: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronic Communications and Computer Science*, vol. E79-A, no. 9, pp. 1338–1354, 1996.

[11] C. H. Pan, S. P Li, Q. H. Zhu, C. Z. Wang, and M. W. Zhang, "Notes on proxy signcryption and multi-proxy signature schemes," *International Journal of Network Security*, vol. 17, no. 1, pp. 29–33, 2015.

[12] D. Pointcheval, and J. Stern, "Security proofs for signature schemes," in *Advances in Cryptology (Eurocrypt'96)*, LNCS 1070, pp. 387–398, Springer, May 1996.

[13] D. Pointcheval, and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptography*, vol. 13, no. 3, pp. 361–396, 2000.

[14] R. A. Sahu, and S, Padhye, "Provable secure identity-based multi-proxy signature scheme," *International Journal of Communication Systems*, vol. 28, no. 3, pp. 497–512, 2015.

[15] A. Shamir, "Identity based cryptosystems and signature schemes," in *Proceedings of Advances in Cryptology (CRYPTO'84)*, LNCS 196, pp. 47–53, Springer, 1984.

[16] H. Singh, and G. K. Verma, "ID-based proxy signature scheme with message recovery," *Journal of Systems and Software*, vol. 85, no. 1, pp. 209–214, 2012.

[17] Y. Sun, C. X. Xu, Y. Yu, and B. Yang, "Improvement of a proxy multi-signature scheme without random oracles," *Computer Communications*, vol. 34, no. 3, pp. 257–263, 2011.

[18] N. Tiwari, and S. Padhye, "An ID-based proxy multi signature scheme without bilinear pairings," in *Proceedings of First International Conference on Security Aspects in Information Technology*, LNCS 7011, pp. 83–92, Springer, 2011.

[19] N. Tiwari, S. Padhye, and D. He "Provably secure proxy multi-signature scheme based on ECC," *Information Technology And Control*, vol. 43, no. 2. pp. 198–203, 2014.

[20] Q. Wang, and Z. F. Cao, "Identity based proxy multi-signature," *Journal of Systems and Software*, vol. 80, no. 7, pp. 1023–1029, 2007.

[21] Z. W. Wang, L. C. Wag, S. H. Zheng, Y. X. Yang, and Z. M. Hu, "Provably secure and efficient identity-based signature scheme based on cubic residues". *International Journal of Network Security*, vol. 14, no. 1, pp. 104–109, 2012.

[22] L. J. Yi, G. Q. Bai, and G. Z. Xiao, "Proxy multi-signature scheme: A new type of proxy signature scheme," *Electronics Letters*, vol. 36, no. 6, pp. 527–528, 2000.

[23] Y. Yu, Y. Mu, W. Susilo, Y. Sun, and Y. F. Ji, "Provably secure proxy signature scheme from factorization," *Mathematical and Computer Modelling*, vol. 55, no. 3-4, pp. 1160–1168, 2012.

**Feng Wang** was born in Shandong province, China, in 1978. He received his B.S. degree in Mathematics from Yantai Normal University (now named Ludong University), Yantai, in 2000 and the M.S. degree in Applied Mathematics from the Guangzhou University, Guangzhou, in 2006. Currently, he is a Lecturer in the College of Mathematics and Physics at Fujian University of Technology and a visiting scholar in Department of Information Engineering and Computer Science at Feng Chia University. His research interests include computer cryptography and information security.

**Changlu Lin** received the BS degree and MS degree in mathematics from the Fujian Normal University, P.R. China, in 2002 and in 2005, respectively, and received the Ph.D degree in information security from the state key laboratory of information security, Graduate University of Chinese Academy of Sciences, P.R. China, in 2010. He works currently for the School of Mathematics and Computer Science, and the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University. He is interested in cryptography and network security, and has conducted research in diverse areas, including secret sharing, public key cryptography and their applications.

**Shih-Chang Chang** received his B.S. degree in 2005 and his M.S. degree in 2007, both in Department of Information Engineering and Computer Science from Feng Chia University, Taichung, Taiwan. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.

**Chin-Chen Chang** received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.