On the Security of Multi-Use Identity Based Proxy Re-Encryption.

A thesis submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

Thesis Supervisor: Prof. C. PanduRangan

by
Kartik Devdatta Hambardikar
April, 2012

Indian Institute of Science Education and Research Pune
Sai Trinity Building, Pashan, Pune India 411021

This is to certify that this thesis entitled "On the Security of Multi-Use Identity Based Proxy Re-Encryption." submitted towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Pune, represents work carried out by Kartik Devdatta Hambardikar under the supervision of Prof. C. PanduRangan.

Kartik Devdatta Hambardikar

Thesis committee:
Prof. C. PanduRangan
Dr.Ayan Mahalanobis

A. Raghuram
Coordinator of Mathematics

To my Parents and my Sister

Acknowledgments

I would like to take this opportunity to thank everyone who have helped me directly or indirectly.

I would like to express my sincere gratitude to my guide Prof. C. Pandu Rangan who simulated my interest in theoretical computer science with his passionate style of teaching and explaining. Prof C. Pandu Rangan perfectly fits as a deity in TCS Lab, IIT Madras; A temple of knowledge and ideas.

I would like to thank, my guide Dr. Ayan Mahalanobis, without whom I wouldn't had entered cryptography. Dr. Ayan's readiness to clear your doubts, and help you in way possible, both academic and otherwise, has been of great help to me. He is a great mentor.

I would like to specially thank Sharmila and Vivek, fellow researchers in the lab, who perfectly fit as the oracles. Their in depth knowledge and helpful nature has been a great help to me. The amount of time and efforts they have selflessly devoted to elucidate concepts is something i will be always thankful for. a special vote of thanks for my lab-mate Guhan. He has been very helpful always and without him, this work would had been impossible.

I would like to thank Chaya, Prateek, Sachin, Dhinakaran, Salini, Bala(Vision Lab), Paresh(M-tech) Akash, Sangeeta, Preeta ma'am, Aneesh and Vinay my lab members and my family at IIT, Madras. They made my stay extremely enjoyable. I had entertaining as well as educative interactions with them.

I would also like to thank my friends, Pallavi, Jay , Neha, Prashant, Mark, Amitosh, Shreyans, Onkar, Vaibhav and Karan for supporting me mentally and helping me whenever i needed them.

But most importantly, I am indebted to my parents and my relatives, for their unselfish love and affection showered upon me and always believing in me. A special vote of thanks for my parents for believing in me and making me what I am today.

Abstract

On the Security of Multi-Use Identity Based Proxy Re-Encryption.

by Kartik Devdatta Hambardikar

In a proxy re-encryption (PRE) scheme, a proxy, authorized by Alice, transforms messages encrypted under Alices public key into encryptions under Bobs public key without learning anything about the underlying message(plaintext). In an Identity-Based Encryption the public key of a user is some unique information about the identity of the user, usually the user's email-ID. When Alice sends mail to Bob at "bob@company.com" she simply encrypts her message using the public key string "bob@company.com". There is no need for Alice to obtain Bobs public key certificate. When Bob receives the encrypted mail he contacts a third party, which we call the Private Key Generator (PKG) authenticates himself and then obtains the private key for himself.

Finding a unidirectional, multi-use, and CCA2-secure identity-based proxy re-encryption scheme was presented as an open problem by Green et al. In 2010 Wang et.al. proposed a Multi-Use Identity Based Proxy Re-encryption Scheme[25] as the solution to the open problem. In this thesis, we have identified a security attack on [25] and also show that the attack is within the scope of the established security model[25].

Keywords: Identity-Based Encryptions, Proxy Re-Encryptions, Provable security.

Contents

Chapter 1

Introduction

Cryptography is the study of information hiding and verification. It includes protocols, algorithms and strategies to securely and consistently prevent or delay unauthorized access to sensitive information and enable verifiability of every component in a communication. Cryptography is an interdisciplinary subject, drawing from several fields. Before the time of computers, it was related to linguistics. Nowadays the emphasis has shifted, and cryptography makes extensive use of technical areas of mathematics and theoretical computer sciences. This includes topics from number theory, information theory, computational complexity, statistics and combinatorics. The proliferation of computers and communications systems brought with it a demand for means to protect information in digital form and to provide security services. The main security requirements needed, which modern cryptography is essentially concerned with are:

- authentication: An entity should be able to prove its identity, or any other validation claims it makes. Also, information exchanged between two parties must be authenticated with respect to its origin and content.

- message confidentiality (or privacy): Only an authorized recipient should be able to extract the contents of the message from its encrypted form. This results in measures to hide, stop or delay unauthorized access to the encrypted information.

- message integrity: The recipient should be able to determine if the message has been altered in any way from the original, by an unauthorized entity.

- non-repudiation: An entity must be able to prevent from denying its previous commitments and actions, viz., sending or signing a message.

1.1 Basic Cryptography

This section would give detailed description about the basic elements and various aspects of the modern field of cryptography.

- Symmetric Key Cryptography
- Asymmmetric Key Cryptography - Public Key Cryptography
- Cryptanalysis

1.1.1 Symmetric Key Cryptography

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key or keys that are easily computationally related. The message is made secure using the secret key and decrypted by the receiver using the same secret key. Assuming that the secret key is not known to anybody besides the sender and receiver, this transaction is supposed to be secure. Block ciphers and Stream ciphers are the prominent examples of symmetric key cryptosystems.
Symmetric key cryptosystems are easy to implement as they have relatively less communicational and computational costs compared to public key cryptosystems. This computational advantage is offset by the complex key management techniques required to maintain security. The limitation that every pair of entities involved will start sharing a secret key, makes the key management more tedious and unsafe when the transfer of the secret key is not over the secure channel.

1.1.2 Public Key Cryptography

In a ground-breaking 1976 paper by Whitfield Diffie and Martin Hellman, they proposed the notion of public-key(asymmetric key) cryptography in which two different mathematically related keys were used. Two different keys, (usually a part of pair) are used by the sender and the receiver to encrypt and decrypt information. The

user generates a pair of keys: the public and private key, which are mathematically related, and the task of obtaining the private key from the knowledge of public key is computationally infeasible under the assumptions of hard problems.
The public key is used for encrypting information corresponding for a particular user. The public key published by the user is usually a random string and is not related to the identity of a person. Hence a third party is required for authenticating the public key associated with a user. Certification Authority (CA) does this job of verifying the ownership of the public key associated with the particular user and hence issuing a certificate guaranteeing this fact. This certificate is called digital certificate, which basically assures that information is secure and not tampered. The private key is kept secret and is used for decrypting ciphertext.

Encryption

The aim of an encryption/decryption process is to preserve the secrecy and the integrity of the message which is being transferred during the transaction. The user encrypts the message using the public key of the receiver, and the message is decrypted by the receiver using his/her secret key. Hence the 2 main algorithms involved in a encryption scheme are:

1. Encryption$(m, pk) \rightarrow (c)$, where input is a message m and the public key(pk) of the user and the output is an encrypted message called the cipher text(c).

2. Decryption $(c, sk) \rightarrow (m)$, where input is the a ciphertext c corresponding to the secret key sk, which outputs message m.

Common to both these cryptographic primitives is one algorithm which is used prior to their actual operation. This algorithm is termed as KeyGen (Key Generation algorithm) which performs the generation of the public and private key pair for every user in the system. The public key is known by all the users in the system while private key is kept secret with each user. This algorithm generates key pairs with a strong mathematical relationship such that utilizing a property such that, with a public key the private counterpart can never be calculated in polynomial time algorithm.

Signatures

The main aim is to preserve authentication and non-repudiation of the sender to the receiver of the message. The security of the signature is that it cannot be forged or separated from the message. The message is signed by the sender using his private key and is verified by the receiver using the sender's public key. Hence there are 2 main algorithms involved in a signature scheme:

1. Sign $(m, ssk) \rightarrow (\sigma)$, where input is a secret signing key ssk and the message m and the output is a signed message σ.

2. Verify $(\sigma, vpk) \rightarrow (accept$ or $reject)$, where input is a signed message σ and verifying public key vpk and the output is a reject or accept.

1.2 Other Flavours of Cryptography

In this section we will describe briefly about identity-based cryptography and certificate-less cryptography.

1.2.1 Identity-Based Cryptography

In order to simplify the certificate management problem associated with Public Key Infrastructure, the notion of Identity Based Cryptography was introduced by Shamir in 1984. The distinguishing feature of this class is that the public key of a user is not restricted to some particular value satisfying strict mathematical properties; instead, any unambiguous, publicly known binary string confirming the identity of an entity, such as email address or the IP can be used as a public key. A trusted third party, called the private key generator (PKG), generates the corresponding private keys, with the help of a master private/secret key(msk), for which the corresponding master public key (mpk) is also published. Since the public keys are easily obtainable in this system, it greatly reduces the complexity of establishing and maintaining the public key authentication frame work, as the need for certificate issue and storage is obviated. Only the PKGs system parameters need to be known by a user for starting the process of information exchange. One caveat associated with this category is that it has an inherent problem of key escrow - the PKG is required to be highly trusted, since it is capable of generating any users private key, and consequently, may decrypt

or sign messages in an unauthorized manner. Any compromise on the PKGs part would lead to a total breakdown of the system.

1.2.2 Certificate-less Cryptography

With a view to mitigate the key escrow problem inherently associated with simple ID-based Cryptosystems, a variant of ID-based cryptography known as Certificate-less Cryptography was introduced by Al-Riyami and Paterson in 2003. Intuitively, this division is a combination of useful features of both the public key as well as the ID-based forms of cryptography, while managing to avoid the flaws and limitations associated with them. In this form, the key generation process is split between the individual user, and the PKG.The trusted PKG first produces a partial private key for the user, and the other part of the key is generated by the user randomly, in a manner similar to PKI. The latter part is kept secret from all other parties, including the PKG. For operations to be carried out, the full private key consisting of both components is required. This automatically creates a binding between the identity and public key of a user, thus certification takes place implicitly. Even though the identity no longer forms the entire public key, due to implicit certification and solution of the key escrow, additional security is achieved without additional computational complexity, thus making this category an attractive option.

1.3 Provable Security

Until the late 1980s, the security of cryptographic schemes was not rigorously defined, instead heuristics, empirical techniques and ad hoc reasoning was used to show their security properties. The notion of provable security brought about a significant change in modern cryptography, by seeking to develop more formal and rigorous techniques, and a mathematical framework under which the security claims of a protocol could be mathematically reasoned out, and hence more reliable. An initial step in this direction was taken by Goldwasser and Micali [1982], where the authors described the first public-key encryption scheme which is provably secure under standard cryptographic assumptions. The general idea is to prove that no reasonable attacker can break a scheme in practice. In this paradigm, along with a definition of the protocol, we precisely describe an adversarial model, consisting of the assumptions regarding an adversarys capabilities, which include the computational resources available at

its disposal, as well as the method and limits of its interaction with the legitimate parties engaged in a protocol. Next, we capture the security claims of a scheme, by defining the goals of the adversary trying to break it. The subsequent approach is to build a system based on certain atomic primitives: usually well-known computational problems with practical assumptions regarding their intractability. The final step is a reductionist proof of security, mathematically derived and making use of complexity theory concepts, which relates the hardness of breaking the cryptosystem to the hardness of breaking the underlying atomic primitives. A valid proof constructed in this manner assures us that the only way for the adversary to achieve its goals, is to break the atomic primitives. The implication is that as long as the primitive computational problem remains computationally infeasible, the prescribed cryptosystem is secure under the chosen definition of security and adversarial model.

1.4 Random Oracle and Standard Model

In modern Cryptography, schemes are being proved secure in an unconventional fashion which was inspired from the field of Computational Complexity theory i.e, The Random Oracle. This model of proof exploits the complexity theory of the protocol and introduces theoretical black boxes called Oracles. These Oracles are formally defined as a mathematical function which maps each query to a random but true response from the output domain. These responses are uniformly distributed in the output domain and it responds to each query the same way every time when the previously asked query is asked again. We can assume is that it is maintaining an update list for the queries and its output. As stated by [Bellare and Rogway] the random oracles are assumed to have the properties that they should possess for practical purposes. The idea is that: Provide all parities -good and bad alike- with the (public) random oracle. Prove a scheme secure using this protocol. Then we can replace these random oracles with objects like cryptographic hash functions. Replacing the random oracle is a heuristic step. In the random oracle model, oracles are imaginary functions and are an abstraction, used in the proof of security in the assumption, as opposed to Standard Model proofs where no such assumptions are made. This strong randomness assumption is suitable for modelling these oracles around cryptographic hash functions for certain schemes. Such a proof generally shows that a system or a protocol is secure by showing that an attacker must require impossible behaviour

from the oracle, or solve some mathematical problem believed hard, in order to break the protocol. No real function can implement a true random oracle. In fact, certain artificial signature and encryption schemes are known which are proven secure in the random oracle model, but which are trivially insecure when any real function is substituted for the random oracle. Nonetheless, for any more natural protocol a proof of security in the random oracle model gives very strong evidence that an attack which does not break the other assumptions of the proof, if any (such as the hardness of integer factorization) must discover some unknown and undesirable property of the hash function used in the protocol to work. On the other hand the conventional notion of proving security of cryptographic schemes was in the Standard model (Bare or Plain Model). Those sole assumptions of the proof of security in this model is based on computational assumption that certain hard problems like the discrete log, factorization, etc. cannot be computed in polynomial time. That is the adversary is given only limited amount of time and computational power at his discretion to attack the scheme. Schemes which can be proven secure using only complexity assumptions are said to be secure in the standard model. Security proofs are difficult to achieve in the standard model, but they provide a very strong sense of security as they are no assumption being based on the randomness of functions.

Chapter 2

Preliminaries

2.1 Definitions

In this section, we state some essential definitions needed for the scheme and attack.

2.1.1 Bilinear Pairing

Let $\mathbb{G}$ be an additive group and let $\mathbb{G}_T$ be a multiplicative group, both of prime order p and let g be generator of $\mathbb{G}$. The bilinear map $\hat{e}$ is admissible only if it satisfies the following conditions:

- ***Bilinearity.*** For all $g_1, g_2, g_3 \in \mathbb{G}$,
  - $\hat{e}(g_1 + g_2, g_3) = \hat{e}(g_1, g_3)\hat{e}(g_2, g_3)$
  - $\hat{e}(g_1, g_2 + g_3) = \hat{e}(g_1, g_2)\hat{e}(g_1, g_3)$
  - $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
- ***Non-Degeneracy.*** For all $g_1, g_2 \in \mathbb{G}$, $\hat{e}(g_1, g_2) \neq I_{\mathbb{G}_T}$, where $I_{\mathbb{G}_T}$ is the identity element of $\mathbb{G}_T$.
- ***Computability.*** There exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$.

Bilinear Diffie-Hellman assumption: In this part, we will study the Biliner Diffie Hellman(BDH) assumption. Given $(g, g^a, g^b, g^c) \in \mathbb{G}^3$ for unknown $a, b, c \in \mathbb{Z}_p$, the BDH problem in $\mathbb{G}$ is to compute $e(g, g)^{abc}$.

2.1.2 Hardness Assumptions

These are the hardness assumptions on which the schemes are usually based on. Here we describe the various hardness assumptions and state the problems.

Discrete Logarithm Problem

If p is a prime, $g, h \in \mathbb{G}$, and $x \in Z_p^*$, and $h = g^x$. Then finding x given g and h is called a Discrete Logarithm Problem(DLP)

Computational Diffie-Hellman Problem

Given $(g, g^a, g^b) \in \mathbb{G}^3$ for unknown $a, b \in \mathbb{Z}_p$, the Computational Diffie-Hellman (CDH) problem in $\mathbb{G}$ is to compute g^{ab}.

Definition: The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}$ is defined as:

$$Adv_{\mathcal{A}}^{CDH} = Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab} \mid a, b \in \mathbb{Z}_p\right]$$

The *CDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{CDH}$ is negligibly small.

Decisional Diffie-Hellman Problem

Given $(g, g^a, g^b, \alpha) \in \mathbb{G}^3 \times \mathbb{G}_{\mathbb{T}}$ for unknown $a, b \in \mathbb{Z}_p$, the Decisional Diffie-Hellman (DDH) problem in $\mathbb{G}$ is to decide if $\alpha = g^{ab}$.

Definition: The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the Decisional Diffie-Hellman (DDH) problem in $\mathbb{G}$ is defined as:

$$Adv_{\mathcal{A}}^{DDH} = Pr\left[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1\right] - Pr\left[\mathcal{A}(g, g^a, g^b, \alpha) = 1\right] \mid a, b \in \mathbb{Z}_p\right]$$

The *DDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{DDH}$ is negligibly small.

Bilinear Diffie-Hellman problem

Given $(g, g^a, g^b, g^c) \in \mathbb{G}^3$ for unknown $a, b, c \in \mathbb{Z}_p$, the Bilinear Diffie-Hellman (BDH) problem in $\mathbb{G}$ is to compute $e(g, g)^{abc}$.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the BDH problem in $\mathbb{G}$ is defined as:

$$Adv_{\mathcal{A}}^{BDH} = Pr\left[\mathcal{A}(g, g^a, g^b, g^c) = e(g,g)^{abc} \mid a, b, c \in \mathbb{Z}_p\right]$$

The *BDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{BDH}$ is negligibly small.

Decisional Bilinear Diffie-Hellman Problem

Given $(g, g^a, g^b, g^c, \alpha) \in \mathbb{G}^4 \times \mathbb{G}_T$ for unknown $a, b, c \in \mathbb{Z}_p$, the Decisional Bilinear Diffie-Hellman (DBDH) problem in $\mathbb{G}$ is to decide if $\alpha = \hat{e}(g,g)^{abc}$.

Definition: The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the DBDH problem in $\mathbb{G}$ is defined as:

$$Adv_{\mathcal{A}}^{DBDH} = Pr\left[\mathcal{A}(g, g^a, g^b, g^c, \hat{e}(g,g)^{abc}) = 1\right] - Pr\left[\mathcal{A}(g, g^a, g^b, g^c, \alpha) = 1\right] \mid a, b, c \in \mathbb{Z}_p\right]$$

The *DBDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{DBDH}$ is negligibly small.
The Discrete Log Problem assumption is a weak assumption compared to the Computational Diffie-Hellman assumption which is inturn a weaker assumption compared to the Decisional Diffie-Hellman assumption. Similarly, Bilinear Diffie-Hellman assumption is a weak assumption compared to the Decisional Bilinear Diffie-Hellman assumption. Weaker the assumption used in the scheme, more secure is the scheme.

Chapter 3

Literature Survey

3.1 Encryption

The aim of an encryption/decryption process is to preserve the secrecy and the integrity of the message which is being transferred during a transaction. The sender encrypts the message using the public key of the receiver, and then ciphertext is decrypted by the receiver using his secret key.

3.1.1 Formal model of Encryption

A generic encryption scheme consists of the following three algorithms:

- ***Setup:***
  In this algorithm, a security parameter is given as an input and the Setup algorithm outputs $params$ as the public parameters, the public keys (pk) and private keys (sk) of the users. Public parameters and pk are publically known while private key sk is kept secret with the receiver.

- ***Encryption:***
  This algorithm is run by the sender to create an encryption on a message m to be sent to the receiver. This algorithm takes as input, the public parameters $params$, the public key pk of the receiver and the message m to be encrypted. The ciphertext c is produced as output which is sent to the receiver.

- ***Decryption:*** On receiving the ciphertext c from the sender, the receiver runs this algorithm. The public parameters $params$, secret key of the receiver sk

and the ciphertext c are given as input to this algorithm. The message intended for the receiver m is given as output of this algorithm.

3.1.2 Security of Encryption Schemes

The main security needed for the encryption schemes are

1. Indistinguishability
2. Non-malleability

Indistinguishability
This is the definition for indistinguishability given by Bellare et al. The goal of the encryption is basically to preserve the privacy of the message. The adversary should not be able to obtain any information about the plaintext from the information of ciphertext, besides maybe the length of the plaintext.

Non-malleability
In indistinguishability, the goal of the adversary is to just guess the plaintext corresponding to the given ciphertext. But, there will be some adversaries who indeed cannot decrypt only with the above information, but can modify the ciphertext into anther valid ciphertext by doing some simple operations so that the modified ciphertext becomes a valid ciphertext for some other random plaintext. This kind of forgery by the adversary is not captured by the notion of indistinguishability.

We will explain the following security models for our thesis:

1. IND-CPA
2. IND-CCA
3. IND-CCA2

These are weaker notions of security called *Indistinguishability under Chosen Plaintext Attack*(IND-CPA), *Indistinguishability under Chosen Ciphertext Attack*(IND-CCA) compared to the *Indistinguishability under Chosen Ciphertext Attack 2*(IND-CCA2). The proof of these security model is given in a game format. The description of these security models is as follows.

IND-CPA Game

An encryption scheme is secure against indistinguishable chosen plaintext attack (IND-CPA) attack if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm and sends the public parameters *params* to the adversary $\mathcal{A}$. The challenger retains the secret key.

2. The Adversary may perform encryptions and other operations.

3. Adversary $\mathcal{A}$ randomly chooses two plaintext messages m_0 and $m_1 \in \{0,1\}^{l_m}$ and receiver Public Key pk^* on which he wishes to be challenged and sends them to the challenger $\mathcal{C}$.

4. $\mathcal{C}$ takes a bit b randomly from $\{0,1\}$ and runs **Encrypt**(m_b^*, sk^*), where sk^* is the secret key corresponding to public key pk^*. $\mathcal{C}$ sends the output c^* to $\mathcal{A}$ as the challenge ciphertext.

5. The adversary $\mathcal{A}$ outputs b'. $\mathcal{A}$ wins this game if $b' = b$.

The advantage of adversary $\mathcal{A}$ in the above game is defined by

$$Adv(\mathcal{A}) = (2 \times Pr(b' = b) - 1)$$

In the above game, the adversary is not provided with the decryption oracle of any randomly generated ciphertext. But, in the real world scenario, an attacker can have access to see a ciphertext and its decrypted message. So, there is less advantage for $\mathcal{A}$ and hence less possibility for the adversary to break the scheme.

IND-CCA Game

An encryption scheme is secure against indistinguishable chosen ciphertext attack (IND-CCA) attack if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game:

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm and sends the public parameters to the adversary $\mathcal{A}$. The challenger retains the secret key.

2. Now the adversary $\mathcal{A}$, in its Training Phase 1, can ask a polynomially bound number of queries to the Decryption oracle.
   Decryption Oracle: $\mathcal{A}$ queries for the decryption of the ciphertext c for producing the underlying plaintext m. During this phase $\mathcal{A}$ can produce its queries adaptively i.e every query is dependant on the previous queries.

3. At the end of Phase 1 $\mathcal{A}$ chooses two plaintext messages m_0 and $m_1 \in \{0,1\}^{l_m}$ and receiver Public Key pk^* on which it wishes to be challenged and sends them to the challenger $\mathcal{C}$.

4. $\mathcal{C}$ takes a bit b randomly from $\{0,1\}$ and runs **Encrypt**(m_b^*, sk^*), where sk^* is the secret key corresponding to public key pk^*. $\mathcal{C}$ sends the output c^* to $\mathcal{A}$ as the challenge ciphertext.

5. The adversary $\mathcal{A}$ outputs b'. $\mathcal{A}$ wins this game if $b' = b$.

The advantage of adversary $\mathcal{A}$ in the above game is defined by

$$Adv(\mathcal{A}) = (2 \times Pr(b' = b) - 1)$$

IND-CCA2 Game

Once a message is encrypted with public key pk of the receiver no one can change the message which is encrypted or do an educated guess of the message encrypted from the ciphertext given. This is the strongest notion available for proving the security of the encryption schemes. This notion is stated formally as follows.
An encryption scheme is semantically secure against chosen ciphertext attack (IND-CCA2) attack if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm and sends the public parameters to the adversary $\mathcal{A}$ and keeps the secret key sk to itself.

2. Now the adversary $\mathcal{A}$, in its Training Phase 1, can ask a polynomially bound number of queries to the Decryption oracle.

Decrypt Oracle: $\mathcal{A}$ queries for the decryption of the ciphertext c also producing the producing the underlying plaintext m. During this phase $\mathcal{A}$ can produce its queries adaptively i.e every query is dependant on the previous queries.

3. At the end of Phase 1 $\mathcal{A}$ randomly chooses two plaintext messages m_0 and $m_1 \in \{0,1\}^{l_m}$ and receiver Public Key pk^* on which it wishes to be challenged and sends them to the challenger $\mathcal{C}$.

4. $\mathcal{C}$ takes a bit b randomly from $\{0,1\}$ and runs **Encrypt**(m_b^*, sk^*), where sk^* is the secret key corresponding to public key pk^*. $\mathcal{C}$ sends the output c^* to $\mathcal{A}$ as the challenge ciphertext.

5. This is Phase 2 of the Training Phase. Now $\mathcal{A}$, after receiving c^* can ask again for polynomially bound number of queries on the Decrypt oracle adaptively in the same way as in Phase 1 except that $\mathcal{A}$ cannot ask for the Decrypt query involving $\langle c^*, pk^* \rangle$.

6. Once this Phase 2 of Training is over, the adversary $\mathcal{A}$ outputs b'. $\mathcal{A}$ wins this game if $b' = b$.

The advantage of adversary $\mathcal{A}$ in the above game is defined by

$$Adv(\mathcal{A}) = (2 \times Pr(b' = b) - 1)$$

Any Scheme is usually decided secure on the notions defined above. IND-CCA2 is the highest level of security compared to IND-CCA and IND-CPA. IND-CPA is the lowest level of security.

3.2 Identity-Based Encryption

In 1984, Shamir proposed a concept of identity-based cryptography. In this system of cryptography, the user's identifier information, such as the email ID of the users can be used as public key for encryption or signature verification, instead of the digital certificates. Hence, Identity based cryptography drastically reduces the cost of maintaining a Public Key Infrastructure.
The basic concept behind identity-based encryption, is that sender Alice uses the

receiver's identifier information which can be any string such as the email ID of that user, to encrypt the message. At the receiver's end, Bob identifies itself to a trusted third party and it generates the private key for Bob, from which Bob can decrypt the ciphertext sent by Alice. The trusted third party is called Private Key Generator (PKG), which generates the private keys for users.
The Identity Based Encryption scheme usually has the following working:

1. ***Setup****:*
   The Private Key Generator creates a master secret key (msk) and the corresponding master public key (mpk). The master public key is publically available.

2. ***Private Key Extraction****:*
   The receiver authenticates himself to the private key generator and obtains his secret key corresponding to his public key. The private key is computed using the master public key and master private key.

3. ***Encryption****:*
   The sender uses the public key of the receiver and the master public key of the IBE to encrypt the plaintext message M.

4. ***Decryption****:*
   The receiver after receiving the ciphertext, uses his private key provided by the IBE to decrypt the ciphertext to recover the underlying message.

Boneh and Franklin[6] introduced the mathematical primitive 'bilinear pairing' in their Identity Based Encryption Scheme in 2001. At the same time Cock's[10] used the variant of 'integer factorization' to construct his Identity Based Encryption Scheme. But due to the ciphertext size in Cock's scheme, Cock's scheme is inefficient and hence Boneh Franklin's scheme is used widely.

Many schemes are based on the Bilinear Diffie Hellman assumption in identity based encryption schemes. Boneh Franklin Identity Based Encryption Scheme is also based on the Bilinear Diffie Hellman assumption.

3.2.1 Boneh Franklin IBE:

Boneh and Franklin[6] proposed the first pairing based Identity Based Encryption in 2001. Boneh and Franklin also came up with the security definition of IBE and also the reductionist proof that their IBE scheme is secure in their proposed security model under the hardness assumption of Bilinear Diffie Hellman.
Boneh Franklin (BF) IBE scheme is proposed ion a two step process. In the first step, a BasicIdent scheme is proposed and shown to be secure under IND-ID-CPA security model. Basically, in this step , they try to simulate the private key extraction queries made by the adversary. Then using the BasicIdent, they propose the FullIdent scheme which is IND-ID-CCA secure in the random oracle model.
The BasicIdent scheme is as intuitively can be explained as follows

- ***Setup:***
  Let ê: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear pairing and p be a generator of $\mathbb{G}$. Pick a random $x \in \mathbb{Z}_p^*$ and set $P_{pub} = x\mathrm{P}$. x is the master private key and P_{pub} is the master public key which is generated by the Private Key Generator. Choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n$ So basically the master secret key is x which will be known only to the Private Key Generator (PKG) and the public parameters are params = {P, P_{pub}, ê, H_1, H_2, $\mathbb{G}$, $\mathbb{G}_T$}

- ***Key-Gen:***
  Given an identity ID (which is a random length string) $\in \{0,1\}^*$, the Private Key Generator (PKG) computes the private key for the identity ID , $d_{ID} = H_1(ID)^x$. d_{ID} is the private key corresponding to the identity ID.

- ***Encryption:***
  To encrypt a message M $\in \{0,1\}^n$ to the identity ID, compute $Q_{ID} = H_1(ID)$. The encrypter chooses a random $r \in \mathbb{Z}_p^*$ and computes the ciphertext as follows:

  $$C = \langle rP, M \oplus H_2(\hat{e}(Q_{ID}, P_{pub})^r) \rangle$$

  the r used by the encrypter is used like it is the private key of the encrypter and the ciphertext can be opened by only by the person have the secret key corresponding to the public key used in the ciphertext.

- ***Decryption:***
  To decrypt the ciphertext which is of the form $C = \langle U, V \rangle$, the decrypter uses the private key associated with the public key ID, and computes the underlying message as:
  $$V \oplus H_2(\hat{e}(d_{ID}, U)) = M$$
  So if the ciphertext is valid , then the decryption will give a valid message as output.

The security analysis of the Boneh Franklin IBE scheme is dependent on the BDH instance. Given a BDH instance the challenger sets up parameters for his IBE scheme and hands over the public parameters (params) to the adversary. If the adversary wins the game then the challenger gets the solution to the BDH problem. The master secret key is associated with the solution of that BDH instance.
FullIdent is the result of applying the Fujisaki-Okamoto (FO) transformation[13] to BasicIdent. FO Tansformation converts a IND-ID-CPA secure scheme to a IND-ID-CCA secure secure.

Let $\mathcal{E}_{pk}(M; r)$ be the encryption of message M under the public key pk and the randomness r in some public key encryption scheme and the let the scheme be a CPA secure scheme. Fujisaki-Okamoto converts this scheme to a CCA secure scheme by addition of a two hash functions and a randomness σ. The new scheme after transformation is called hybrid encryption, so:

$$\mathcal{E}_{pk}^{hybrid}(M) = \langle E_{pk}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle$$

The security of this Boneh-Franklin scheme is proved by a two step game.

To reduce the workload from the Private Key Generator (PKG), Horwitz and Lynn [16] suggested that a hierarchy of Private Key Generators in which the PKGs have to compute secret/private keys only to the users immediately below them in the hierarchy should be incorporated to a normal IBE scheme. In Hierarchy Identity Based Encryption, the users is attached not only to his identity but he is identified by a tuple of identities which contains identities of each of his ancestor. However Horwitz and Lynn were not able to design a fully function-able HIBE.Gentry and Silverberg [14], realized a fully-function-able HIBE scheme that allows a general n-level hierarchy using Boneh and Franklins IBE scheme.

3.2.2 Other ID-Based Schemes:

During the time of Boneh-Franklin scheme, Cocks[10] had also proposed an Identity Based Encryption Scheme based on Quadratic Residues. Cocks scheme is not based on bilinear pairing. However, the drawback of the Cock's Scheme is that it encrypts each bit of the plaintext and hence the length of the cipher text is very large and hence in-efficient. Boneh and Boyen proposed a scheme[7] that was adaptively secure without the use of random oracles. Boneh and Boyen used the Selective Identity Security model. However, Boyen and Boneh scheme was inefficient and more for theoretical than practical use. Waters[24] modified the same scheme and proposed a scheme which was practically implementable.
Table 1 shows the various Identity Based Schemes and their properties. Their underlying hard problem, security model and random oracle(RO) dependency. It also gives the relation to pairings.

Table 1: Identity Based Encryption schemes

IBE Scheme	**RO / RO free**	**Hard Problem**	**Security Model**	**Pairing based**
Boneh Franklin IBE[6]	R-O	BDH	IND-ID-CCA	Pairing
Boneh-Katz-Wang[17]	R-O	BDH	CPA	Pairing
Attrapadung et. al.	R-O	BDH	CCA	Pairing
Boneh-Boyen[7]	R-O free	DBDH	Selective Identity	Pairing
Waters[24]	R-O free	DBDH	Selective Identity	Pairing
Cocks[10]	R-O	RSA variant	-	Pairing free

3.3 Proxy Re-Encryption

Proxy Re-Encryption allows the proxy to transform a ciphertext under Alice's secret to a ciphertext under Bob's secret.

Alice, who is the president of the company wants to temporarily forward his emails to Bob, who is the vice-president of the company without giving Bob her secret key. Alice can decrypt the mails using her private key and encrypt again using Bob's public key, but what if Alice is off-line? Alice can assign a third party (proxy) with her private key and hence the proxy can decrypt the mails to Alice, encrypt using Bob's public key and send it to Bob. But this requires an unrealistic trust in proxy.

The first Proxy Re-Encryption scheme was proposed by Blaze, Bleumer and Strauss[5] in 1998, where the proxy was not a fully trusted proxy but a semi trusted proxy. Blaze, Bleumer and Strauss (BBS) proposed a notion of 'atomic proxy cryptography' where the semi-trusted proxy converts the ciphertext made for Alice into ciphertexts made for Bob without understanding the underlying plaintext. After that many other schemes were proposed with various properties.

The person whose ciphertext is going to get is called the delegator and the person receiving the re-encrypted ciphertext is called the delegatee. The semi-trusted third party whose re-encrypts the delegator's ciphertext for the delegatee is called the proxy.

Proxy Re-Encryption have varied applications. Proxy Re-Encryption can be used for email forwarding, law enforcement, and performing cryptographic operations on storage-limited devices.

Proxy re-encryption schemes have applications in digital rights management (DRM), distributed file storage systems, law enforcement, encrypted email forwarding, and outsourced filtering of encrypted spam. The motive of Re-Encryption is decrypting under one key for encryption under another key, should not allow the re-encryptor module to compromise the secrecy of encrypted messages. This was related to the compromise of Apples iTunes DRM. With the increasing importance of cloud computing, proxy re-encryption is going to be tremendously important.

The first Proxy Re-Encryption scheme was proposed by Blaze and Bleumer in 1998. It is basically based on simple modification of the Elgamal encryption scheme. Let $(\mathbb{G}, *)$ be a group of prime order p and g be the generator of the group $mathbbG$. Let the public key of Alice be $X = g^x$ and of Bob be $Y = g^y$, where x, y are the secret keys of Alice and Bob respectively. A sender wants to send a message $m \in \mathbb{G}$

to Alice, so the sender randomly selects $r \in Z_p^*$ and then converts it to ciphertext as follows. $C_{Alice} = (C_1, C_2)$, where $C = X^r$ and $C_2 = m * g^r$. The proxy P is given the re-encryption key rekey = y/x. So the proxy can convert the ciphertext C_{Alice} to C_{Bob} by computing $C_1' = (C_1)^{rekey}$ and $C_2' = C_2$. and hence now Bob can decrypt the mails using his secret key.

The definitions of some of the properties of Re-Encryption schemes are as follows:

- *Unidirectional:*
  Delegation from $A \rightarrow B$ does not allow re-encryption from $B \rightarrow A$. Unidirectional schemes are usually preferred but in some applications bi-directional schemes are also required.

- *Proxy Invisibility:*
  It does not require the sender who sends message to Alice to be aware of the existence of the proxy. The same should hold for the delegatee.

- *Collusion Resistance:*
  It is hard for the delegatee and the proxy to combine and compute the delegator's private key.

- *Non-interactive:*
  The generation of re-encryption key should be accomplished by the delegator using delegatee's public key. No interaction with the delegatee is required for creating the Re-encryption key.

- *Non-transitive:*
  The proxy, alone cannot re-delegate decryption rights.It should be hard for the proxy to use the re-keys from $rk_{1\rightarrow 2}$ and $rk_{2\rightarrow 3}$ and compute $rk_{1\rightarrow 3}$ without involving any parties.

Dodis and Ivan[12] proposed a unidirectional scheme but it was not collusion resistant. Ateniese et al. proposed a a scheme[4] in 2006 but it had some short comings. It was CPA secure and also the proxy alone could create delegation rights even if the two parties never agreed for it. Many schemes were proposed after that. Various schemes with their properties have been stated in the table 2.

Table 2 : Proxy Re-Encryption Schemes

PRE Scheme	**Uni-Bi-directional**	**Security Model**	**RO-free**	**Collusion Resistant**	**Pairing free**
BBS Scheme[5]	Bi	CPA	No	No	Yes
Dodis-Ivan[12]	Uni	-	No	No	Yes
Ateniese et.al.[4][2]	Uni	CPA	No	Yes	No
Hohenberger et.al[15]	Uni	CPA	Yes	Yes	No
Canetti-Hohenberger[8]	Bi	CCA	Yes	No	No
Libert-Vergnaud[18]	Uni	RCCA	Yes	Yes	No
Libert-Vergnaud-Trace[19]	Uni	CPA	Yes	Yes	No
Deng et.al	Bi	CCA	No	No	Yes
Shao-Cao	Uni	CPA	No	No	Yes
Ateniese et.al.[1]	Uni	CPA	Yes	Yes	No
Sherman Chow et.al.	Uni	CCA	No	Yes	Yes
Weng et.al[11]	Uni	CCA	-	No	Yes

A formal definition of Proxy Re-Encryption schemes is given as follows:

Definition:
A proxy Re-Encryption Scheme is a tuple of probabilistic polynomial time (ppt) algorithms (Setup, KeyGen, ReKeyGen, Enc, Re-Enc, Dec). The Description of these algorithms are as follows:

1. ***Setup:***
   The setup algorithm takes k as input security parameter and outputs parameters $params$, which include a description of the message space M. The parameters $params$ are publically available.

2. ***KeyGen:***
   This Key-Gen (Key Generation) algorithm outputs public key (pk_i) and secret key (sk_i) of the user i.

3. ***Re-Key-Gen:***
   On Input of the delegator's private key (sk_i), delegatee's public key (pk_j), the Re-Key-Gen algorithm generates the Re-Encryption Keys ($rk_{i \to j}$) from the user i to j.

4. ***Enc:***
   This algorithm takes pk_i as the public key of the user i and the message m and outputs a ciphertext c_i for the user i using the public key pk_i with the underlying message m.

5. ***Re-Enc:***
   On input of the re-encryption keys ($rk_{i \to j}$) and the ciphertext (c_i) of the user pk_i, the Re-Encryption algorithm outputs the ciphertext (c_j) for the user j.

6. ***Dec:***
   On input of the private of the private key sk_j of the user under j and the corresponding ciphertext (c_j), the Decryption algorithm outputs the message m corresponding to the ciphertext c_j.

Chapter 4

Proposed attack on MU-IB-PRE

This chapter deals with the existing Multi-Use Identity Based Proxy Re-Encryption (MU-IB-PRE) scheme and our proposed attack.

4.1 Existing MU-IB-PRE

Hongbing Wang, Zhenfu Cao and Licheng Wang[25] proposed a Multi-Use and Unidirectional identity-based proxy re-encryption scheme in the journal Information Sciences in 2010. The proposed identity based proxy re-encryption scheme by Wang and Wang is IND-CCA2 secure had has multiple properties like multi-use and unidirectional etc.The scheme is based on Decisional Bilinear Diffie-Hellman assumption in the random oracle model. Under the security model proposed by the authors of the paper, the scheme is IND-CCA2 secure.

Description of the scheme: The scheme proposed by Wang et. al. is an identity based encryption scheme which generates re-encryptable ciphertexts and is proven to be IND-PrID-CCA2 secure in the random oracle model. The scheme is multi-use and unidirectional. Construction of a unidirectional and multi-use IB-PRE scheme was presented as an open problem by Green and Ateniese.

Multi-Use is an important property for a proxy re-encryption schemes. A multi-use PRE scheme permits the proxy to convert the ciphertext from A to B, then the result ciphertext can be converted from B to C by the same or different proxy. An ID based proxy re-encryption allows the choosing of identifier string related to a user as the public key and hence can be re-encrypted from one user to another and then further ahead. Everytime, a string attached to the user is used as the public key of the user.

The sender, encrypts the message using ID_1's public key and sends the ciphertext to ID_1. For re-encryption, the proxy P_1 is given the re-encryption key to convert the ciphertext for ID_1 to ID_2 without proxy understanding the underlying plaintext. As the scheme is multi-use, for further re-ecnryption, ID_2 gives the proxy P_2 , the re-encryption keys for converting the ciphertext from ID_2 to ID_3. The same continues further from ID_3 to ID_4 and so on...

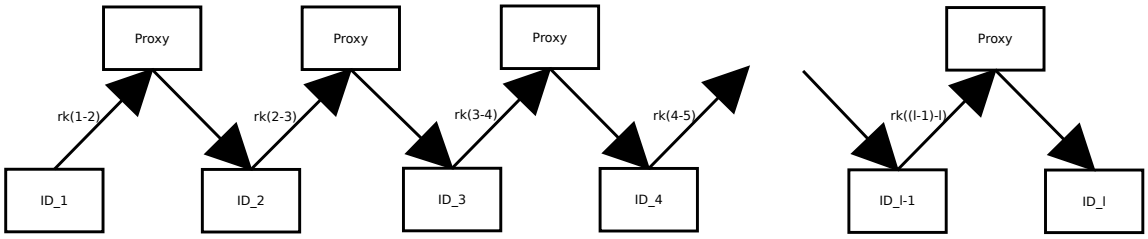

Figure 4.1: Identity-Based Proxy Re-Encryption

The H. Wang et. al. proposed scheme is as follows: Let 1^k be the security parameter and (q, g, $\mathbb{G}$, $\mathbb{G}_T$, $\hat{e}$) $\leftarrow$ Bsetup(1^k), and *Sig = (G, S, V)* be a strongly unforgeable signature scheme, where l=l(k) denotes the length of the verification keys output by $G(1^k)$. The IB-PRE scheme proposed by Wang et.al. consists of six algorithms

$$(Setup, Extract, RKExtract, Encrypt, Reencrypt, Decrypt)$$

Setup: (1^k) Let the message space be M = $0, 1^{k_0}$ such that $k_0 < k$ and both 2^{-k_0} and 2^{k-k_0} are negligible. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative groups of the same prime order q, such that the discrete log problems in both $\mathbb{G}$ and $\mathbb{G}_T$ are intractable. Suppose that g is the generator of $\mathbb{G}$, and ê: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear pairing map. Let $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}^*$ be two hash funtions. To generate the scheme parameters, select master secret key to be msk = $s \leftarrow_R Z_q^*$ and output the

system parameters as

$$params = (M, k, k_0, \mathbb{G}, \mathbb{G}_T, q, \hat{e}, H_1, H_2, g, g^s)$$

Extract: $(params, msk, ID)$
To generate a decryption key for the identity $\in \{0,1\}^*$, compute $sk_{ID} = H_1(ID)^s$ and send it to the user with the identity ID via a secure channel.
Encrypt: (params, ID, m)
To encrypt the message $m \in M$ under the identity ID, do the following:

1. Select $s \leftarrow_R Z_q^*$ and $\sigma \in \{0,1\}^{k-k_0}$ at random.
2. Compute C = $\{c_{1,1}, c_{1,2}, c_{1,3}\}$, where $c_{1,1} = g^r$, $c_{1,2} = (m||\sigma)\hat{e}(g^s, H_1(ID)^r)$, and $c_{1,3} = H_2(m||\sigma||c_{1,1})g^{r\sigma}$
3. Compute U = $H_4(c_{1,1}||c_{1,2}||c_{1,3})^r$
4. Output the ciphertext $c_{id}^1 = \langle C, U \rangle$

RKExtract: (params, sk_{id_i}, id_j)
. To generate a re-encryption key for $id_i's$ proxy P_i, id_i selects $r_i \leftarrow_R Z_q^*$, $X_i \leftarrow_R \mathbb{G}_T$ at first, then computes

$$R_1^i = g^r, R_2^i = X_i\hat{e}(g^s, H_1(ID_j)^{r_i}), R_3^i = svk_{p_i}$$

,

$$R_4^i = H_2(svk_{p_i})^{r_i}, R_5^i = H_2(R_1^i||R_2^i||R_3^i)^{r_i}, R_6^i = H_3(X_i)sk_{id_i}^{-1}$$

where svk_{P_i} is a publically available verification key of P_i. Finally, id_i outputs $rk_{id_i \to id_j}$ = $(R_1^i, R_2^i, R_3^i, R_4^i, R_5^i, R_6^i)$

Re-encrypt: (params, $rk_{id_i \to id_j}$, $c_{id_i}^l$)

1. To re-encrypt a first level ciphertext $c_{id_i}^l$, denoted by $c_{id_i}^1$, do the following:
   (a) Pass $c_{id_i}^1$ as $(c_{1,1}, c_{1,2}, c_{1,3}, c_{1,4})$ and $rk_{id_i \to id_j}$ as $(R_1^i, R_2^i, R_3^i, R_4^i, R_5^i, R_6^i)$.
   (b) Check if $\hat{e}(g, c_{1,4}) = \hat{e}(c_{1,1}, h_4 c_{1,1}||c_{1,2}||c_{1,3})$. If not , return $\perp$

(c) Otherwise, compute $\mathrm{C} = (c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{1,4}, c'_{2,1}, c'_{2,2}, c'_{2,3}, c'_{2,4}, c'_{2,5})$, where

$$c'_{1,1} = c_{1,1}, c'_{1,2} = c_{1,2}\hat{e}(c_{1,1}, R_6^1), c'_{1,3} = c_{1,3}, c'_{1,4} = c_{1,4}$$

$$c'_{2,1} = R_1^1, c'_{2,2} = R_2^1, c'_{2,3} = R_3^1, c'_{1,1} = R_4^1, c'_{2,5} = R_5^1$$

(d) Suppose ssk_{P_1} is the signature key of id'_1s proxy P_1 corresponding to R_3^1

(e) Run the signing algorithm $S(ssk_{p_1}, (c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{1,4}, c'_{2,1}, c'_{2,2}, c'_{2,3}, c'_{2,4}, c'_{2,5}))$ to generate a signature on the ciphertext tuple

$$(c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{1,4}, c'_{2,1}, c'_{2,2}, c'_{2,3}, c'_{2,4}, c'_{2,5})$$

, and denote the signature as S_1

(f) Output the ciphertext $c_{id_j}^2 = \langle C, S_1 \rangle$.

2. To re-encrypt at *lth-level(l¿1)* ciphertext $c_{id_j}^l$, do:

(a) Pass $c_{id_i}^l$ as $(c_{1,1}, ..., c_{l,1}, c_{l,2}, c_{l,3}, c_{l,4}, c_{l,5}, S_{l-1})$ and $rk_{id_i \to id_j}$ as $(R_l^i, R_l^i, R_l^i, R_l^i, R_l^i, R_l^i)$.

(b) Check if $\hat{e}(g, c_{l,4}) = \hat{e}(c_{l,1}, h_4 c_{l,1} || c_{l,2} || c_{l,3})$. If not , return $\perp$

(c) For $\forall k \in [2, l]$, checking

i. whether $e(g, c_{k,4}) = e(c_{k-1}, H_2(c_{k,3}))$, and

ii. $V(svk_{P_{k-1}}, S_{k-1}, (c_{1,1}, ..., c_{k,1}, c_{k,2}, c_{k,3}, c_{k,4}, c_{k,5})) = 1$.
Whenever one of them fails, return $\perp$. Otherwise, do the next:

(d) Compute $\mathrm{C} = (c'_{1,1}....., c'_{l,1}, c'_{l,2}, c'_{l,3}, c'_{l,4}, c'_{l,5}, c'_{l,6}, c'_{l+1,1}, c'_{l+1,2}, c'_{l+1,3}, c'_{l+1,4}, c'_{l+1,5})$, where $c'_{l,2} = c'_{l,2}\hat{e}(c_{1,1}, R_6^l)$, $c'_{l+1,1} = R_1^l, c'_{l+1,2} = R_2^l, c'_{l+1,3} = R_3^l, c'_{l+1,4} = R_4^l, c'_{l+1,5} = R_5^l$, and all other elements remain unchanged.

(e) Suppose ssk_{P_i} is the signature key of id_i's proxy P_i corresponding to R_3^l.

(f) Run the signing algorithm $S(ssk_{p_i}, (c'_{1,1}, ..., c'_{1+1,1}, c'_{1+1,3}, c'_{1+1,4}, c'_{l+1,5}))$ to generate a signature on the ciphertext tuple $(c'_{1,1}, ..., c'_{1+1,1}, c'_{1+1,3}, c'_{1+1,4}, c'_{l+1,5})$, and denote the signature as S_l

(g) Output the ciphertext $c_{id_j}^{l+1} = \langle C, S_l \rangle$

Decrypt: (params, $sk_{id_i}, c_{id_j}^l$).
If $c_{id_j}^l$ can not be passed as $c_{1,1}, c_{1,2}, c_{1,3}, c_{1,4}$ for the first level ciphertext, or

$(c'_{1,1}, ..., c'_{1,1}, ..., c'_{l,6})$ for an *lth*-level ciphertext ($l > 1$), then return $\perp$; Otherwise continue the following process:

(a) If l=1. then perform

i. Verify that $\hat{e}(g, c_{1,4}) = \hat{e}(c_{1,1}, H_2(c_{1,1}||c_{1,2}||c_{1,3}))$. If not, return $\perp$.

ii. Otherwise, compute $m' \leftarrow c_{1,2}/\hat{e}(c_{1,1}, sk_{id})$.

iii. Pass m' as (m, σ)

iv. Verify that $c_{1,3} = H_2(m||\sigma||c_{1,1})c_{1,1}^{\sigma}$. If not, return $\perp$; Otherwise, output m.

(b) Otherwise, if $l > 1$, perform

i. Verify that $\hat{e}(g, c_{l,5}) = \hat{e}(c_{l,1}, H_2(c_{l,1}||c_{l,2}||c_{l,3}))$. If not, return $\perp$.

ii. For $\forall$k $\in [2, l]$, check

A. Whether $\hat{e}(g, c_{k,4}) = \hat{e}(c_{k,1}, H_2(c_{k,3}))$

B. $V(svk_{P_{k-1}}, S_{k-1}, (c_{1,1}, ..., c_{k,1}, c_{k,2}, c_{k,3}, c_{k,4}, c_{k,5})) = 1$.
Whenever one of them fails, output $\perp$. Otherwise, do the following:

iii. Compute $X_{l-1} \leftarrow c_{l,2}/\hat{e}(c_{l,1}, sk_{id})$.

iv. For i from l-2 down to 1, compute

$$X_i \leftarrow c_{i+1,2}/\hat{e}(c_{i+1,1}, H_3 X_{i+1})$$

v. Compute $m' \leftarrow c_{l,2}/\hat{e}(c_{l,1}, H_3(X_1))$

vi. Pass m' as (m, σ)

vii. If $c_{1,3} \neq H_2(m||\sigma||c_{1,1})c_{1,1}^{\sigma}$, return $\perp$; Otherwise return m.

4.2 Proposed attack on MU-IB-PRE:

Description of the attack

The scheme proposed by Wang et. al [25] is not secure against CPA attack. Any receiver of the Multi-hop Proxy Re-encrypted cipher text can decrypt the ciphertext (not meant for him) in the future after receiving a multi-hop proxy re-encrypted ciphertext.

Illustration with an example:

Let σ_1 be a ciphertext of the message m encrypted by the sender for the receiver with identity id_1. Let σ_2, σ_3, σ_4, σ_5 be the ciphertext obtained by re-encrypting σ_1 for receiver's id_2, id_3, id_4, id_5 respectively.

$$Sender \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3 \rightarrow \sigma_4 \rightarrow \sigma_5$$

Now consider id_5 decrypts the ciphertext σ_5 using his secret key corresponding to id_5. As per the decryption algorithm proposed in the scheme [25], id_5 obtains the intermediate value $X_{4,5}$, $X_{3,4}$, $X_{2,3}$, $X_{1,2}$ as in the step (b)(3) of the decryption in [25].

These values correspond to

$$\sigma_1 \xrightarrow{X_{1,2}} \sigma_2 \xrightarrow{X_{2,3}} \sigma_3 \xrightarrow{X_{3,4}} \sigma_4 \xrightarrow{X_{4,5}} \sigma_5$$

Now, id_5 can decrypt any re-encrypted ciphertext using $X_{1,2}$, if the re-encryption structure has

$$sender \longrightarrow id_1 \longrightarrow id_2 \longrightarrow ...$$

We will now show how the adversary in the CPA game can decrypt the challenge ciphertext without having the secret key of the challenge receiver and without violating the constraints given in the confidentiality game of the MU-IB-PRE:

1. Let id_A be a user for which the adversary knows the private key

$$(sk_A) = H_1(ID_A)^s$$

2. Let σ^* be the challenge ciphertext for the challenge receiver id_T

3. Now the adversary A generates the encryption of message m which is not equal to m_0 or m_1[challenge messages] for receiver id_T. Let the ciphertext be σ_T

4. The adversary queries the re-encrypt oracle with σ_T for re-encryption from id_T to id_2 where σ_2 is the resulting ciphertext.

5. The adversary again queries the re-encrypt oracle with σ_2 as the input for re-encryption from id_2 to id_A. Here adversary knows the private key of id_A. Thus adversary decrypts σ_A and obtains all the intermediate values and the message m.

6. Step (b)(3) of the decrypt algorithm, the adversary obtains the value X_{T2} and X_{2A}. Note that X_{T2} is same for all the re-encryption from id_T to id_2 irrespective of the ciphertext being re-encrypted.

7. Now the Adversary can re-encrypt the σ^* (challenge ciphertext) from id_T to id_2 [Note: The private key of id_2 is not known to A], where $\hat{\sigma_2}$ is the resulting ciphertext for which id_2 as the receiver. Thus this is a legal query.

8. Again the adversary queries the re-encrypt oracle. The adversary queries the re-encrypted ciphertext $\hat{\sigma_2}$ for re-encryption from id_2 to id_3. In the scheme, X_{T2} is the same for all ciphertexts, re-encrypted from id_T to id_2 and the adversary knows X_{T2} for the previous steps. Thus the adversary can decrypt the challenge ciphertext using X_{T2} and obtian the message m^*.

Therefore the scheme is not CPA-secure.

This attack is demonstrated below:

1. The challenge ciphertext $\sigma^* = \text{Encrypt}(m^*, id_T) =$
   $\langle c^*_{1,1} = g^r,$
   $c^*_{1,2} = (m^*||R) \cdot \hat{e}(g^s, H_1(ID_T)^r),$
   $c^*_{1,3} = H_2(m^*||R||c_{1,1})^r,$
   $c^*_{1,4} = H_4(c_{1,1}||c_{1,2}||c_{1,3})^r\rangle$

2. Compute $\sigma_T = \text{Encrypt}(m, id_T) =$
   $\langle c_{(1,1)_T} = g^{r_T},$
   $c_{(1,2)_T} = (m||\sigma) \cdot \hat{e}(g^s, H_1(ID_T)^{r_T}),$

$c_{(1,3)_T} = H_2(m||\sigma||c_{1,1})^{r_T}$,
$c_{(1,4)_T} = H_4(c_{1,1}||c_{1,2}||c_{1,3})^{r_T}\rangle$

3. $\sigma_2 = \text{ReEncrypt}(\sigma_T, id_T, id_2) =$
$\langle c_{(1,1)_2} = g^{r_T}$,
$c_{(1,2)_2} = c_{1,2_T} \cdot \hat{e}(c_{1,1}, R_6^{id_T \to id_2}) = (m||\sigma) \cdot \hat{e}(g, H_3(X_{T2}))^{r_T}$,
$c_{(1,3)_2} = H_2(m||\sigma||c_{1,1})^{r_T}$,
$c_{(1,4)_2} = H_4(c_{1,1}||c_{1,2}||c_{1,3})^{r_T}$,
$c_{(2,1)_2} = R_1^{id_T \to id_2} = g^{r_{id_T \to id_2}}$,
$c_{(2,2)_2} = R_2^{id_T \to id_2} = X_{T2} \cdot \hat{e}(g^s, H_1(ID_2)^{r_{id_T \to id_2}})$,
$c_{(2,3)_2} = R_3^{id_T \to id_2} = svp_{P_{id_T \to id_2}}$,
$c_{(2,4)_2} = R_4^{id_T \to id_2} = H_2(svp_{P_{id_T \to id_2}})^{id_T \to id_2}$,
$c_{(2,5)_2} = R_5^{id_T \to id_2} = H_2(R_1^{id_T \to id_2}||R_2^{id_T \to id_2}||R_3^{id_T \to id_2})^{r_{id_T \to id_2}}$,
$S_2 = S(ssk_{P_{id_T \to id_2}}(c_{(1,1)_2}, c_{(1,2)_2}, c_{(1,3)_2}, c_{(1,4)_2}, c_{(2,1)_2}, c_{(2,2)_2}, c_{(2,3)_2}, c_{(2,4)_2}, c_{(2,5)_2}))\rangle$

4. $\sigma_A = \text{ReEncrypt}(\sigma_2, id_2, id_A) =$
$\langle c_{(1,1)_A} = g^{r_T}$,
$c_{(1,2)_A} = (m||\sigma) \cdot \hat{e}(g, H_3(X_{T2}))^{r_T}$,
$c_{(1,3)_A} = H_2(m||\sigma||c_{1,1})^{r_T}$,
$c_{(1,4)_A} = H_4(c_{1,1}||c_{1,2}||c_{1,3})^{r_T}$,
$c_{(2,1)_A} = g^{r_{id_T \to id_2}}$,
$c_{(2,2)_A} = c_{2,2_2} \cdot \hat{e}(c_{2,1}, R_6^{id_2 \to id_A}) = X_{T2} \cdot \hat{e}(g^s, H_3(X_{2A}))$,
$c_{(2,3)_A} = svp_{P_{id_T \to id_2}}$,
$c_{(2,4)_A} = H_2(svp_{P_{id_T \to id_2}})^{id_T \to id_2}$,
$c_{(2,5)_A} = H_2(R_1^{id_T \to id_2}||R_2^{id_T \to id_2}||R_3^{id_T \to id_2})^{r_{id_T \to id_2}}$,
$c_{(2,6)_A} = S(ssk_{P_{id_T \to id_2}}(c_{(1,1)_2}, c_{(1,2)_2}, c_{(1,3)_2}, c_{(1,4)_2}, c_{(2,1)_2}, c_{(2,2)_2}, c_{(2,3)_2}, c_{(2,4)_2}, c_{(2,5)_2}))$,
$c_{(3,1)_A} = R_1^{id_2 \to id_A} = g^{r_{id_2 \to id_A}}$,
$c_{(3,2)_A} = R_2^{id_2 \to id_A} = X_{2A} \cdot \hat{e}(g^s, H_1(ID_A)^{r_{id_2 \to id_A}})$,
$c_{(3,3)_A} = R_3^{id_2 \to id_A} = svp_{P_{id_2 \to id_A}}$,
$c_{(3,4)_A} = R_4^{id_2 \to id_A} = H_2(svp_{P_{id_2 \to id_A}})^{id_2 \to id_A}$,
$c_{(3,5)_A} = R_5^{id_2 \to id_A} = H_2(R_1^{id_2 \to id_A}||R_2^{id_2 \to id_A}||R_3^{id_2 \to id_A})^{r_{id_2 \to id_A}}$,
$S_A = S(ssk_{P_{id_T \to id_2}}(c_{(1,1)_A}, c_{(1,2)_A}, c_{(1,3)_A}, c_{(1,4)_A}, c_{(2,1)_A}, c_{(2,2)_A}, c_{(2,3)_A}$,
$c_{(2,4)_A}, c_{(2,5)_A}, c_{(2,6)_A}, c_{(3,1)_A}, c_{(3,2)_A}, c_{(3,3)_A}, c_{(3,4)_A}, c_{(3,5)_A})\rangle$

5. X_{T2} is obtained by decrypt (σ_A, sk_A) as follows:

- $Compute X_{2A} = (c_{3,2_A}/\hat{e}(c_{3,2_A}, sk_{id_A}))$
- from X_{2A}, we compute,
  $X_{T2} = (c_{2,2_A}/\hat{e}(c_{2,1_A}, H_3(X_{2A})))$

6. $\hat{\sigma_2} = \text{ReEncrypt}(\sigma^*, id_T, id_2) =$
   $\langle c^*_{(1,1)} = g^r,$
   $c^*_{(1,2)} = c^*_{1,2} \cdot \hat{e}(c^*_{1,1}, R_6^{id_T \to id_2}) = (m^*||R) \cdot \hat{e}(g, H_3(X_{T2}))^r,$
   $c^*_{(1,3)} = H_2(m^*||R||c_{1,1})^r,$
   $c^*_{(1,4)} = H_4(c^*_{1,1}||c^*_{1,2}||c^*_{1,3})^r,$
   $c^*_{(2,1)} = R_1^{id_T \to id_2} = g^{r_{id_T \to id_2}},$
   $c^*_{(2,2)} = R_2^{id_T \to id_2} = X_{T2} \cdot \hat{e}(g^s, H_1(ID_2)^{r_{id_T \to id_2}}),$
   $c^*_{(2,3)} = R_3^{id_T \to id_2} = svp_{P_{id_T \to id_2}},$
   $c^*_{(2,4)} = R_4^{id_T \to id_2} = H_2(svp_{P_{id_T \to id_2}})^{id_T \to id_2},$
   $c^*_{(2,5)} = R_5^{id_T \to id_2} = H_2(R_1^{id_T \to id_2}||R_2^{id_T \to id_2}||R_3^{id_T \to id_2})^{r_{id_T \to id_2}},$
   $S^*_2 = S(ssk_{P_{id_T \to id_2}}(c^*_{(1,1)_2}, c^*_{(1,2)_2}, c^*_{(1,3)_2}, c^*_{(1,4)_2}, c^*_{(2,1)_2}, c^*_{(2,2)_2}, c^*_{(2,3)_2}, c^*_{(2,4)_2}, c^*_{(2,5)_2}))\rangle$

7. $\hat{\sigma_3} = \text{ReEncrypt}(\hat{\sigma_2}, id_2, id_3) =$
   $\langle c^{**}_{(1,1)} = g^r,$
   $c^{**}_{(1,2)} = (m^*||R) \cdot \hat{e}(g, H_3(X_{T2}))^r,$
   $c^{**}_{(1,3)} = H_2(m^*||R||c_{1,1})^r,$
   $c^{**}_{(1,4)} = H_4(c^*_{1,1}||c^*_{1,2}||c^*_{1,3})^r,$
   $c^{**}_{(2,1)} = R_1^{id_T \to id_2} = g^{r_{id_T \to id_2}},$
   $c^{**}_{(2,2)} = c^*_{2,2_2} \cdot \hat{e}(c^*_{2,1}, R_6^{id_2 \to id_3}) = X_{T2} \cdot \hat{e}(g^s, H_3(X_{23})),$
   $c^{**}_{(2,3)} = R_3^{id_T \to id_2} = svp_{P_{id_T \to id_2}},$
   $c^{**}_{(2,4)} = R_4^{id_T \to id_2} = H_2(svp_{P_{id_T \to id_2}})^{id_T \to id_2},$
   $c^{**}_{(2,5)} = R_5^{id_T \to id_2} = H_2(R_1^{id_T \to id_2}||R_2^{id_T \to id_2}||R_3^{id_T \to id_2})^{r_{id_T \to id_2}},$
   $c^{**}_{(2,6)} = S(ssk_{P_{id_T \to id_2}}(c^*_{(1,1)_2}, c^*_{(1,2)_2}, c^*_{(1,3)_2}, c^*_{(1,4)_2}, c^*_{(2,1)_2}, c^*_{(2,2)_2}, c^*_{(2,3)_2}, c^*_{(2,4)_2}, c^*_{(2,5)_2})),$
   $c^{**}_{(3,1)} = R_1^{id_2 \to id_A} = g^{r_{id_2 \to id_A}},$
   $c^{**}_{(3,2)} = R_2^{id_2 \to id_A} = X_{2A} \cdot \hat{e}(g^s, H_1(ID_A)^{r_{id_2 \to id_A}}),$
   $c^{**}_{(3,3)} = R_3^{id_2 \to id_A} = svp_{P_{id_2 \to id_A}},$
   $c^{**}_{(3,4)} = R_4^{id_2 \to id_A} = H_2(svp_{P_{id_2 \to id_A}})^{id_2 \to id_A},$
   $c^{**}_{(3,5)} = R_5^{id_2 \to id_A} = H_2(R_1^{id_2 \to id_A}||R_2^{id_2 \to id_A}||R_3^{id_2 \to id_A})^{r_{id_2 \to id_A}},$
   $S^{**}_A = S(ssk_{P_{id_T \to id_2}}(c_{(1,1)_A}, c_{(1,2)_A}, c_{(1,3)_A}, c_{(1,4)_A}, c_{(2,1)_A}, c_{(2,2)_A}, c_{(2,3)_A},$
   $c_{(2,4)_A}, c_{(2,5)_A}, c_{(2,6)_A}, c_{(3,1)_A}, c_{(3,2)_A}, c_{(3,3)_A}, c_{(3,4)_A}, c_{(3,5)_A})\rangle$

8. Message m^* is obtained as follows:

- X_{2T} is already precomputed in the step
- $M = c^{**}_{1,2}/\hat{e}(c^{**}_{1,1}, H_3(X_{2T}))$
- Pass M as (m^*, R)

Hence the adversary can win the game. Hence the adversary can break the scheme with a non-negligible advantage.
This shows that the scheme is not CPA secure as the adversary does not make any decryption queries and still can break the scheme.
One way of avoiding the attack maybe by introduction of randomness by the proxy. The randomness introduced by the proxy will be new everytime the proxy gets to re-encrypt a new round of ciphertext. The previous randomness used by the identity will be the same but the ciphertext would be also dependent on the randomness introduced by the proxy which will be refreshed every transaction. this might save from this attack. This is a new form of attack, which has to be considered while dealing with multi-use identity based proxy re-encryption.

Chapter 5

Conclusions

Green et. al. had presented, finding a multi-use, undirectional and CCA secure identity based proxy re-encryption scheme as an open problem. Wang et. al. had proposed a scheme [25] which was thought to be CCA-2 secure. Our attack on the scheme [25] without violating the constraints given in the confidentiality game of MU-IB-PRE, shows that the scheme [25] is not CPA secure. We showed that a CPA adversary can break the scheme.

One way of avoiding this attack maybe by the introduction of randomness at the level of proxy too. This will restrict the message to two type of randomness, one of the ID making the re-encryption key and one of the proxy. Hence the adversary won't be able to break the scheme as it has to find both the randomness to decrypt the ciphertext.

Bibliography

[1] G. Ateniese, K. Benson, S. Hohenberger, Key-private proxy re-encryption, in: M. Fischlin (Ed.), CT-RSA, Lecture Notes in Computer Science, vol. 5473, Springer, 2009, pp. 279294.

[2] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, in: NDSS, The Internet Society, 2005.

[3] G. Ateniese and M. Green. Identity-Based Proxy Re-Encryption. In Proc. of ACNS07, LNCS 4521, pp. 288-306, Springer-Verlag, 2007.

[4] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Trans. Inform. Syst. Secur. 9 (1) (2006) 130.

[5] M. Blaze, G. Bleumer, M. Strauss, Divertible protocols and atomic proxy cryptography, in: EUROCRYPT, 1998, pp. 127144.

[6] D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, in: [24], 2001, pp. 213229.

[7] D. Boneh, and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In advances in Cryptology-Eurocrypt04, LNCS 3027, pp. 223-238, Springer-Verlag, 2004

[8] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, in: P. Ning, S.D.C. di Vimercati, P.F. Syverson (Eds.), ACM Conference on Computer and Communications Security, ACM, 2007, pp. 185194.

[9] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In EUROCRYPT 2004, volume 3027 of LNCS, pages 207222, 2004.

[10] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, IMA Int. Conf., volume 2260 of Lecture Notes in Computer Science, pages 360363. Springer, 2001.

[11] R. H. Deng, J. Weng, S. Liu, K. Chen. Chosen-Ciphertext Secure Proxy Re-encryption without Pairings. In proc. of International Conference on Cryptology and Network Security, CANS08, pp. 1-17, 2008.

[12] Y. Dodis, and A.-A. Ivan. Proxy Cryptography Revisited. In Proc. of NDSS03, 2003.

[13] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric En- cryption Schemes, In Advances in Cryptology-Crypto99, LNCS 1666, pp. 537-554, Springer-Verlag, 1999.

[14] C. Gentry and A. Silverberg, Hierarchical ID-Based Cryptography, Proceedings of ASIACRYPT 2002, LNCS 2501, Springer-Verlag 2002, pages 548566.

[15] Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely Obfus- cating Re-encryption. In TCC, volume 4392 of Lecture Notes in Computer Science, pages 233252. Springer, 2007.

[16] J. Horwitz and B. Lynn, Toward Hierarchical Identity-Based Encryption, Proceedings of EUROCRYPT 2002, LNCS 2332, Springer-Verlag 2002, pages 466481.

[17] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, ACM Conference on Computer and Communications Security, pages 155164. ACM, 2003.

[18] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption, in: R. Cramer (Ed.), Public Key Cryptography, Lecture Notes in Computer Science, vol. 4939, Springer, 2008, pp. 360379.

[19] Beno Libert and Damien Vergnaud. Tracing Malicious Proxies in Proxy Re-encryption. In Pairing, volume 5209 of Lecture Notes in Computer Science, pages 332353. Springer, 2008.

[20] J. Shao, Z. Cao, CCA-secure proxy re-encryption without pairings, in: S. Jarecki, G. Tsudik (Eds.), Public Key Cryptography, Lecture Notes in Computer Science, vol. 5443, Springer, 2009, pp. 357376.

[21] J. Shao, Z. Cao, X. Liang, H. Lin, Proxy re-encryption with keyword search, Inf. Sci. 180 (13) (2010) 25762587.

[22] Jun Shao, Peng Liu, Zhenfu Cao, Guiyi Wei: Multi-Use Unidirectional Proxy Re-Encryption. ICC 2011: 1-5

[23] T. Smith, DVD Jon: Buy DRM-less Tracks from Apple Itunes, 2005, http://www.theregister.co.uk/2005/03/18/itunes-pymusique.

[24] Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [73], pages 114127.

[25] Hongbing Wang, Zhenfu Cao, Licheng Wang: Multi-use and unidirectional identity-based proxy re-encryption schemes. Inf. Sci. 180(20): 4042-4059 (2010)

[26] Y. Zhou, B. Fang, Z. Cao, X. chun Yun, X. Cheng, How to construct secure proxy cryptosystem, Inform. Sci. 177 (19) (2007) 40954108.