Ph.D. Dissertation

博士論文

Provable Security of Identity Based Encryption and Application to Design of Efficient Schemes

IDベース暗号の証明可能安全性と効率的方式設計への応用

指導教員　　松浦幹太　准教授

東京大学大学院 情報理工学系研究科 電子情報学専攻

48-067412　　楊 鵬 (YANG Peng)

平成21年6月15日提出

序文

本論文は筆者が東京大学情報理工学系研究科電子情報学に入学した平成 16 年 10 月から，修士課程二年間と博士課程三年間をわたり，現在平成 21 年 7 月に至るまでに行った研究をまとめたものである．

本論文の主要な部分は次の四つの章からなっている．

- 第二章　ID ベース暗号の安全性定義のフレームワーク
- 第三章　マスター鍵の安全性を考慮した ID ベース暗号
- 第四章　ID ベース暗号の安全性強化およびその評価
- 第五章　ID ベース暗号の高速化に関する研究

本論文の主要な部分は第二章〜第五章の四つの章よりなる．論文を通じて，全ての章では ID ベース暗号の証明可能安全性が扱われている．

各章で得られた結果は独立であるが，第二〜五章の証明可能安全性に対する基本的な概念は共通しており，第二章で構造された安全性定義のフレームワークが基礎となっている．第三章で ID ベース暗号独自の安全性を考慮し、モデルと具体的な方式を提案する．第四章で提案された強化手法により，第三章の提案方式の安全性を強化することができる．第五章で討論された手法を用いれば，第三章の提案方式を含め，殆ど全ての既存 ID ベース暗号方式の効率向上効果が期待できる．

これらは独立して研究されたものであるが，その根底となる基本的な考え方は規を一にしている．即ち，本論文は ID ベース暗号の「証明可能安全性」を中心とした信頼性の向上を目指し，それにより新たな ID ベース暗号の構成法，強化法，あるいは諸性質について論じたものである．各章によって証明可能安全性の取り扱い方に多少の差異はあるが，いずれの場合にも安全性帰着は極めて重要な役割を演じる．また，本論文の目的とするところは，より信頼性の高い ID ベース暗号を実現することであるが，第五章に示される二つの理論は ID ベース暗号以外の従来の公開鍵暗号にも応用分野があると論じている．

本論文の第二章は既に情報処理学会論文誌に掲載された (発表文献 $< 1 >$)．第三章は査読つき国際会議に発表した (発表文献 $< 7 >$)．また，第四章の前半は査読つき国際会議 (発表文献 $< 2, 4, 5 >$) に発表されており、まとめたバージョンが電子情報通信学会英文誌に投稿済みであり，第五節は未発表である．第五章の第二節 (発表文献 $< 8 >$) と第三，五節 (発表文献 $< 6 >$) は査読つき国際会議に，第四節 (発表文献 $< 21 >$) は国内研究会に発表したのみで，全章をまとめたバージョンを電子情報通信学会英文誌に投稿予定がある．

Abstract

Identity based encryption (IBE) schemes have been flourishing since the very beginning of this century. In IBE, it is widely believed that proving the security of a scheme in the sense of IND-ID-CCA2 is sufficient to claim the scheme is also secure in the senses of both SS-ID-CCA2 and NM-ID-CCA2. The justification for this belief is the relations among indistinguishability (IND), semantic security (SS) and non-malleability (NM). But these relations are proved *only* for conventional public key encryption (PKE) schemes in previous works. The fact is that between IBE and PKE, there exists a difference of special importance, i.e. only in IBE the adversaries can perform a particular attack, namely the *chosen identity attack*.

We show that security proved in the sense of IND-ID-CCA2 is validly sufficient for implying security in any other sense in IBE. This is to say the security notion, IND-ID-CCA2, captures the essence of security for all IBE schemes. To achieve this intention, we first describe formal definitions of the security notions for IBE, and then present the relations among IND, SS and NM in IBE, along with rigorous proofs. All of these results are proposed with the consideration of the chosen identity attack.

Regarding concrete IBE schemes, there are (at least) two levels of secret information, i.e., the top-level secret, which is called the master key, and the end-level secrets, which are the users' secret keys. In order to minimize damage in case of an adversary successfully expose users' secret keys, forward security has been introduced into IBE. In a forward secure identity based encryption (FSIBE) scheme, the adversary can obtain no information about the compromised user's secret encrypted before the breaking-in time point.

In this paper, we also construct such a scheme with master key update (FSIBEm) that the top-level secret evolves as same as users' secret keys do, so that even if at some time point the adversary compromises the master key, he can no longer generate users's secret keys corresponding to passed time points. The provable security of our proposal is CPA, strictly weaker than CCA2. This means in order to implement this scheme in real world, the security needs to be enhanced.

To achieve CCA2 security, Fujisaki-Okamoto conversions (FOPKC, FOCRYPTO) and REACT conversion are used *specifically* to enhance a weak IBE scheme's security. However whether they can be *generically* used for such purpose was unknown before this work. In this paper, we discuss applications of Fujisaki-Okamoto conversions and REACT conversion in IBE environment. Our results show that all the conversions are *effective*: plain REACT already achieves a good security reduction while plain FOPKC and plain FOCRYPTO result in bad additional running time of the simulator.

To solve this problem, we further propose a modification to plain Fujisaki-Okamoto conversions. Interestingly, our results may also show a separation between two different attack models. Finally, we choose some concrete parameters to visually explain the effect of how

our modifications substantially improve security reduction comparing with the plain applications.

The last contribution of this paper addresses efficient IBE scheme design. In history, the concept of stateful PKE (SPKE), where the senders are asked to maintain some state information, was introduced to reduce computation cost of PKE. Alternatively, the classical PKE schemes are called stateless PKE. Informally speaking, SPKE is a technique of randomness reusing. In such schemes, the sender maintains a state to encrypt single or multiple messages. Thanks to this technique, compared with a PKE scheme, SPKE can surprisingly achieve much better encryption performance, e.g., regarding the ElGamal based SPKE scheme, compared with the stateless counterpart DHIES, the exponentiation computations are reduced from two times to one time for the encryption algorithm. A stateful IBE (SIBE) scheme has been discussed, but the security relies on a loose security reduction and a strong complexity assumption.

This paper presents a new SIBE scheme whose security reduction is tighter and the underlying assumption is weaker. The impact can be considered significant because our scheme allows much shorter parameters and more flexibility of choosing group. Furthermore, we study the essence of SIBE scheme by pointing out that what we need to achieve high efficiency is actually only a cryptographic primitive, and we name it stateful identity based key encapsulation mechanism (SIBKEM). We formalize this primitive, and show a composition theorem of SIBKEM and symmetric key encryption. Also, we propose a generic method of constructing such SIBKEM schemes from a well-studied primitive.

Interestingly, our methodology does not stop only in SIBE field: it also affects SPKE research. By employing our technique, one can achieve SPKE scheme based on weak assumption, and also can formalize stateful (public key) key encapsulation mechanism.

謝辞

本論文の完成にあたり，計六十四名の方々に深甚な謝意を表す．これらの方々は学術面や生活面などから筆者の五年間の研究生活を支えて下さってきた．ここで，図 (敬称略) の示すように，大きく七つの意味で分類する．それぞれは，(象限 I) 指導・教育を頂いた方；(象限 II) 応援・支持を頂いた方 ；(象限 III) 相談して頂いた方 ；(象限 IV) 助言・討論して頂いた方；(区域 V) 研究室関係者； (区域 VI) 発表論文共著者； (区域 VII) 相談・助言・討論して頂いた方となる．特に本研究を進めるにあたり，筆者の指導教員である本学准教授**松浦幹太**先生[1] は中核となり，各側面から本論文の完成に最も重要な役割を果たしていると思われる．以降，各段落内の芳名順は基本的に筆者との出会い順に従う．

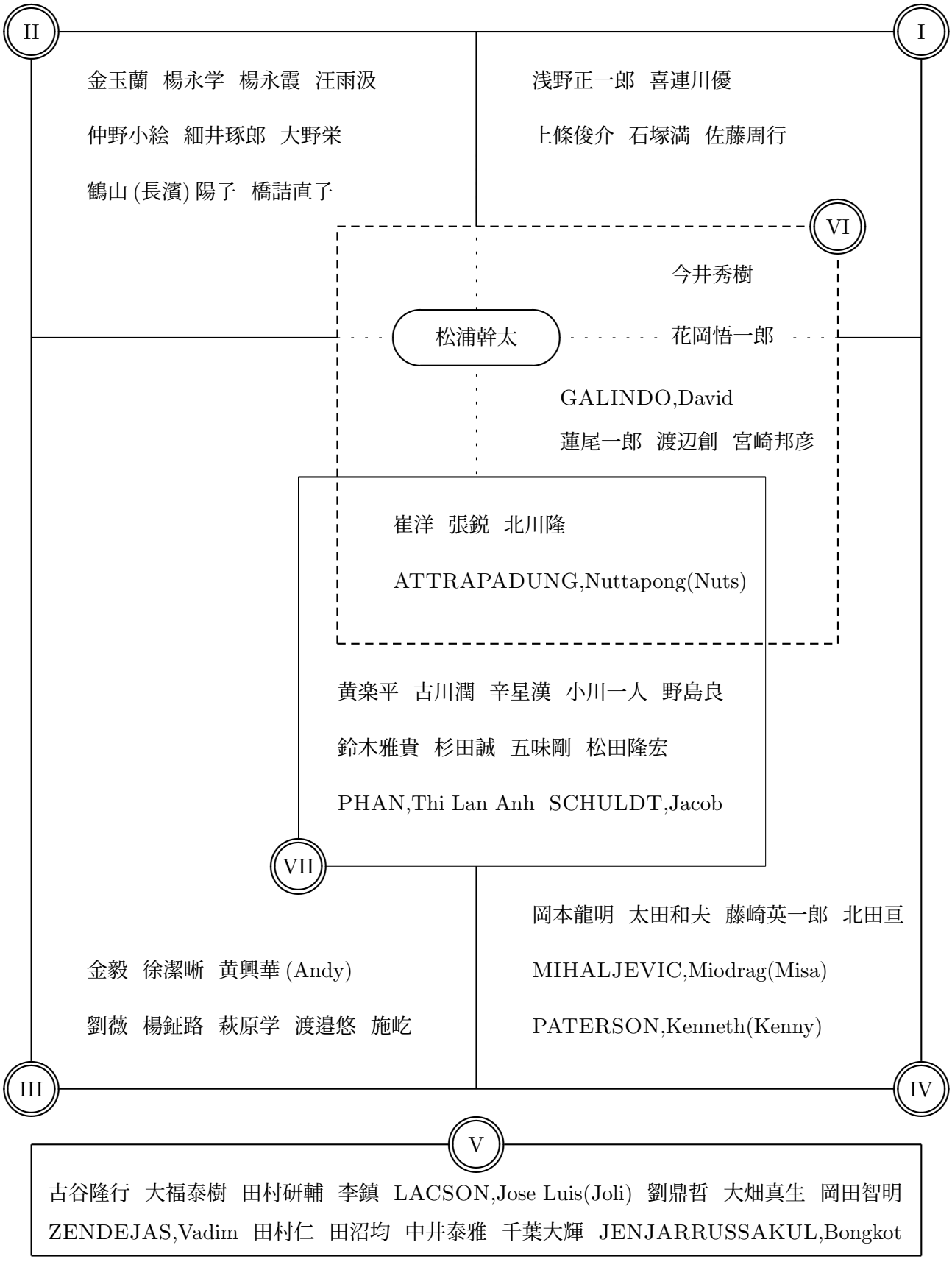
改めて，五年間終始直接御指導を頂いた尊師・松浦先生に心より至高な謝意を申し上げる．松浦先生には，本論文に含まれる四つの研究テーマそれぞれに関する詳細な進み方のみならず，自らの言動から迸る気迫を通じ，学術研究に対して取るべき態度や，科学者であるべき姿勢など高度の指導をはじめ，研究室での倫理や日常の立ち振る舞いなど細部までの教授を頂き，留学生の筆者のメンタルヘルスの面も心遣いして頂き，また八ヶ国国際学会参加および十二回国内学会参加の旅費参加費の調達して頂き，更に修了後の博士後ポジションを確保して頂いたことで，筆者が修士課程・博士課程をわたり学問のみならず，人間性も大きく成長ができ，ただ数十年の人生の中の最も有意義な五年間を過ごすことができたと言える．

全般にわたり，種々御検討を頂いた中央大学教授 (本学名誉教授) **今井秀樹**先生ならびに産業技術総合研究所研究員**花岡悟一郎**さんに至誠な謝意を表す．博士三年間のアドバイザ教員・本学位論文審査会委員を担当し，貴重な示唆を頂いた本学教授**浅野正一郎**先生，**喜連川優**先生に深謝する．また，審査会主査・本学教授**石塚満**先生，審査会委員・本学准教授**上條俊介**先生，**佐藤周行**先生に深謝する．特に，今井先生には，二〇〇四年入門希望の筆者を定員オーバー状態だった松浦研究室に推薦して頂き，その後国際国内学会発表へ複数回同席して頂き，何よりも心強い後援となる．花岡さんには，四年間以上主催される「新明るい暗号勉強会」(旧称 ID ベース暗号勉強会）に参加させて頂き，情報セキュリティ分野の国内海外ベテラン研究者たちを極力に紹介して頂き，暗号理論の入門知識や学術論文の書き方など研究者の基礎技術を教育して頂き，更に筆者の博士課程に不可欠な経済支援となる文部科学省国費留学生奨学金の詳細申請案を企画・執行して頂いた全てのことが，本論文の完成に深く繋がると言い過ぎない．

また，学問に関する独特な視点を共有し，安全性定義基礎概念を詳説して頂いた産総研ポスドク**崔洋**さん；各国見聞を共有し，研究への熱意を感染し，研究討論して共に夜明かしして頂いた産総研研究員**張鋭**さん；共に暗号理論分野で探索し始め，関西文化を紹介して頂いた中央大学研究員**北川隆**さん；最先端デジタル技術に興味を持つ同人であり，異郷でお互いの異文化を共有・尊敬し合って頂いた産総研研究員**アッタラパドゥン・ナッタポン**さん (Nuts)；ならびに **David Galindo** さん[2] ，**蓮尾一郎**さん，**渡辺創**さん，**宮崎邦彦**さん，以上共に論文執筆し

[1] All the research results in this dissertation are under supervision of Associate Professor Kanta Matsuura, from Oct. 2004 to Jul. 2009.

[2] I thank David as a co-author of LATIN'06.


II
I
金玉蘭 楊永学 楊永霞 汪雨汲
仲野小絵 細井琢郎 大野栄
鶴山 (長濱) 陽子 橋詰直子
浅野正一郎 喜連川優
上條俊介 石塚満 佐藤周行
VI
今井秀樹
松浦幹太
花岡悟一郎
GALINDO,David
蓮尾一郎 渡辺創 宮崎邦彦
崔洋 張鋭 北川隆
ATTRAPADUNG,Nuttapong(Nuts)
黄楽平 古川潤 辛星漢 小川一人 野島良
鈴木雅貴 杉田誠 五味剛 松田隆宏
PHAN,Thi Lan Anh SCHULDT,Jacob
VII
岡本龍明 太田和夫 藤崎英一郎 北田亘
金毅 徐潔晰 黄興華 (Andy)
MIHALJEVIC,Miodrag(Misa)
劉薇 楊鉦路 萩原学 渡邉悠 施屹
PATERSON,Kenneth(Kenny)
III
IV
V
古谷隆行 大福泰樹 田村研輔 李鎮 LACSON,Jose Luis(Joli) 劉鼎哲 大畑真生 岡田智明
ZENDEJAS,Vadim 田村仁 田沼均 中井泰雅 千葉大輝 JENJARRUSSAKUL,Bongkot

た八名の方々に謝意を表す.

無線通信分野の知識を教えて頂いた本学博卒**黄楽平**さん；常に中国の歴史に興味を示して頂いたNEC研究員**古川潤**さん；鍵交換分野の知識を共有して頂いた産総研研究員**辛星漢**さん；社会人研究者の立ち振る舞いを見せ，本論文予備審査の準備に貴重な御助言を頂いたNHK研究員**小川一人**さん；古典暗号芸術の面白さを伝わって頂いたNICT研究員**野島良**さん；研究にとって健康の重要性を示して頂いた日本銀行研究員**鈴木雅貴**さん；世界中暗号技術の研究現状を討論して頂いた**杉田誠**さん；IDベース暗号勉強会の初期メンバーであり，共に成長して頂いた**五味剛**さん；如何なる事項でも相談ができ，暗号研究への熱意を感染し，最も頼りになる後輩であるJSPS特別研究員・本研究室博士学生**松田隆宏**君；強い精神力を示して頂いた本研究室修卒**Phan Thi Lan Anh**君；幅広い分野において知識を共有し，健康な生き方から電子署名の先端技術まで複数分野の専門家とも言える本研究室博士学生**Jacob Schuldt**君[3]，以上本論文に適切な御助言を頂き，生活面も豊富にして頂いた十一名の方々に謝意を表す

次に，毎年情報セキュリティ全国大会にて外部研究者と自由自在に交流できる場(岡本部屋)を作って頂いたNTT情報流通プラットフォーム研究所室長**岡本龍明**さん；一生勉強し続ける模範を立てて頂いた電気通信大学教授**太田和夫**先生；理論研究の厳密さを徹底的に追及する姿勢を見せて頂いたNTT研究員**藤崎英一郎**さん；活発な議論を頂いた本研室修卒**北田亘**君；定期的に最新海外の研究動向を報告し，修士論文に貴重な助言を頂いたSANU教授**Miodrag Mihaljevic**先生(Misa)[4]；本論文の第五章第二節に的確な御助言を頂いたRoyal Holloway教授**Kenneth Paterson**先生(Kenny)[5]，以上本研究に関して直接に御討論を頂いた六名の方々に感謝する.

そして，大学同窓でもあり，修士課程・博士課程をわたり長年励まし合いながら，IT業界全般的なテクノロジとエンジニアリングの共通点・分岐点について議論し，将来の夢を語り合って頂いた親友・IBM(中国)研究員**金毅**君[6]；年下でありながら早い段階から研究に投身し，共に科学に頑張ってきた理化学研究所修士学生**徐潔晰**君；ベテラン社会人として社会生存法則を語って頂いた**黄興華** (Andy)さん；唯一の同期である本研究室修卒**劉薇**君；高度な科学視野を持ち，色々なアドバイスを頂いた本学喜連川研究室研究員**楊鉦路**さん；留学生活を多彩にして頂いた産総研研究員**萩原学**さん；高い行動力を有し，後輩でありながら就職活動の心得を共有して頂いた本研究室修卒**渡邉悠**君；旺盛なチャレンジング精神を持ち，元気を共有して頂いた本研究室修士学生**施屹**君，以上八名の友人に感謝する.

入門・入学・進学・出張など手続きを円滑にして頂いた本研元秘書**仲野小絵**さん；研究設備の整備や日常研究生活の面倒を見て頂いた本研技術職員**細井琢郎**さん；在学中五年間裏で学内教務を支えて頂いた本学電気系事務室職員**大野栄**さん；共に楽しい研究室生活を構築して頂いた本研元秘書**鶴山 (長濱) 陽子**さん；研究室に新しい血液として注入し，元気を運んで頂いた本研秘書**橋詰直子**さん，以上御支援を頂いた五名の方々に深謝する.

本研究室定期打ち合わせミーティングに参加の共同研究者・先輩・後輩である**古谷隆行**さん，**大福泰樹**さん，**田村研輔**さん，**李鎮**君，**Jose Luis Lacson (Joli)** さん，**劉鼎哲**さん，**大畑真生**君，**岡田智明**さん，**Vadim Zendejas**君，**田村仁**さん，**田沼均**さん，**中井泰雅**君，**千葉**

[3]I thank Jacob for being a good comrade with vast knowledge.
[4]I thank Misa for discussing his new research results.
[5]I thank Kenny for his suggestion on Chapter 5, Section 2.
[6]I thank Yi for being a great friend.

大輝君，**Bongkot Jenjarrussakul** 君，以上研究室関係者十四名の方々に感謝する．

更に，未曾有のグローバル金融危機に伴う不況の中，本論文の予備審査準備早期から論文まとめの半ばにわたる五ヶ月就職活動をわたり，入社内々定を頂いた株式会社アイ・ティ・フロンティア (**ITF**)，株式会社日立システムアンドサービス (**HitachiSystems**)，みずほ情報総研株式会社 (**Mizuho-IR**)，日本アイ・ビー・エム株式会社 (**IBM**)，ならびに最終選考まで進ませて頂いた日本エリクソン株式会社 (**Ericsson**)，株式会社野村総合研究所 (**NRI**)，アクセンチュア株式会社 (**Accenture**) に感謝する．早い段階で進路が決まったお陰で，更に博士論文に集中でき，より質の高い論文を完成することに間接的な繋がりがあると考えられる．

最後，筆者の全ての業績の根本となるこの命を頂き，人生の初めての教師である慈母**金玉蘭**；重要な世界観・人生観・価値観を慎重に育てて頂いた厳父**楊永学**；筆者が日本へ渡航することに不可欠な役割を果たし，家族に全力を尽くし，二〇〇四年九月に子宮頸癌で逝去した至親の叔母 (父の妹) **楊永霞**；本研究を進めるにあたり，五年間終始お世話になり，学問に専念できる環境構築に協力し，心の平穏を与えて頂いた愛妻**汪雨汲**，以上四名の家族に至心の謝意を申し上げる．

Table of Contents

序文 i

Abstract ii

謝辞 iv

1 INTRODUCTION 1
- 1.1 Research Motivation 1
- 1.2 Overview of Contributions 6
- 1.3 Preliminary 8
  - 1.3.1 Conventions 8
  - 1.3.2 Security Models 9
  - 1.3.3 Complexity Assumptions 13

2 FRAMEWORK OF SECURITY NOTIONS 15
- 2.1 Introduction 15
  - 2.1.1 Security Notions for PKE 15
  - 2.1.2 Towards Defining Security Notions for IBE 16
  - 2.1.3 Contributions 17
- 2.2 Definitions of Security Notions 18
  - 2.2.1 One-wayness 19
  - 2.2.2 Indistinguishability 20
  - 2.2.3 Semantic Security 20
  - 2.2.4 Non-malleability 22
- 2.3 Relations among Security Notions for IBE 22
  - 2.3.1 Equivalence between IND and SS 23
  - 2.3.2 Relation between IND and NM 27
  - 2.3.3 Separation between IND and OW 30
- 2.4 Conclusions 32

3 A FORWARD SECURE SCHEME WITH MASTER KEY UPDATE 33
- 3.1 Introduction 33
  - 3.1.1 Related Works 33
  - 3.1.2 Contributions 34
- 3.2 Security Model of FSIBEm 34
  - 3.2.1 Algorithms of FSIBEm 34
  - 3.2.2 Security Notion of FSIBEm 35

3.3 A FSIBEm Scheme Based on DBDH Assumption in Standard Model 36
3.3.1 Construction 36
3.3.2 Security Proof 37
3.4 Conclusions 37

4 MEANS OF SECURITY ENHANCEMENT 38
4.1 Introduction 38
4.1.1 On Achieving IND-ID-CCA2 Security 39
4.1.2 Contributions 39
4.2 Investigation and Security Proof of Plain Enhancements 40
4.2.1 IND-ID-CPA Enhancement 40
4.2.2 OW-ID-CPA Enhancement 43
4.2.3 OW-ID-PCA Enhancement 47
4.2.4 Discussion 51
4.3 Towards More Efficient Enhancements 52
4.3.1 More Efficient IND-ID-CPA Enhancement 52
4.3.2 More Efficient OW-ID-CPA Enhancement 54
4.3.3 Discussion 56
4.4 Numerical Explanation by Encryption Time 57
4.4.1 Parameter Setting 57
4.4.2 T_P of Plain FOPKC 58
4.4.3 T_M of Modified FOPKC 58
4.4.4 Discussion 59
4.5 Numerical Explanation by Group Size 60
4.5.1 Parameter Setting 60
4.5.2 Main Result 61
4.6 Conclusions 61

5 ON DESIGN OF EFFICIENT SCHEMES 63
5.1 Introduction 63
5.1.1 Related Works 64
5.1.2 Contributions 64
5.1.3 Security Notions of Stateful Encryption 65
5.2 Stateful Identity Based Key Encapsulation Mechanism 67
5.2.1 Algorithms of SIBKEM 67
5.2.2 Security Notion of SIBKEM 68
5.2.3 Composition Theorem 68
5.2.4 Generic Construction of SIBE 70
5.2.5 Instantiations and Comparisons 72
5.3 How to Remove Gap Assumptions and Maintaining Tight Reductions 75
5.3.1 Construction 75

5.3.2 Security Proof 75
5.4 Extension♯1: Stateful Key Encapsulation Mechanism 76
5.4.1 Algorithms of SKEM 76
5.4.2 Security Notion of SKEM 76
5.4.3 Composition Theorem 77
5.5 Extension♯2: SPKE Based on Weak Assumption 79
5.5.1 Construction 79
5.5.2 Security Proof 80
5.6 Conclusions 80

6 CONCLUSION **81**

Bibliography **82**

Publicatoins **86**

Chapter 1 INTRODUCTION

This dissertation argues nothing concrete about how better identity based encryption is than the other cryptographic primitives. This dissertation strictly focuses on techniques how better we can improve identity based encryption.

Identity based encryption (IBE) is a public key encryption mechanism where an arbitrary string, such as the recipient's identity, can serve as a public key. This convenience eliminates the need to distribute public key certificates. On the other hand, in conventional public key encryption (PKE) schemes, it is unavoidable to access the online public key directory in order to obtain the public keys. IBE schemes are largely motivated by many applications such as encrypting emails with the recipient's e-mail address.

Although the basic concept of IBE was proposed by Shamir [44] more than two decades ago, it is extremely challenging to find a fully functional scheme. It took cryptographers in the world more than 15 years to accomplish such a mission. Only very recently was the first scheme proposed [16]. In 2001, Boneh and Franklin defined a security model and gave the first fully functional solution provably secure in the random oracle model [6]. After hierarchical IBE was introduced [28], subsequent researches further extended to the standard model [19, 20, 14, 13, 48, 29, 18, 21, 12].

Every year, there are talks about IBE research in top-level information security conferences. For many years, IBE has been one of the hottest research fields. People discusses all aspects of IBE's security where might exits a potential attack. People find new ways of employing IBE to provide solutions in various scenarios in real world. People propose novel cryptographic primitives based on basic IBE component. But, this dissertation argues nothing about how better identity based encryption is than the other cryptographic primitives. This dissertation strictly focuses on techniques how better we can improve identity based encryption.

1.1 Research Motivation

First of all, from a high level, Figure 1.1 illustrates an overview of all topics that will be discussed in this dissertation. Frame R, G and E represent improvement of reliability, generality and efficiency, respectively. And the framework of security notions should be considered as the very foundation that supports not only all the three aspect above, but also every IBE research with provable security.

Before discussing how to establish such an important framework, we first review the case of PKE.

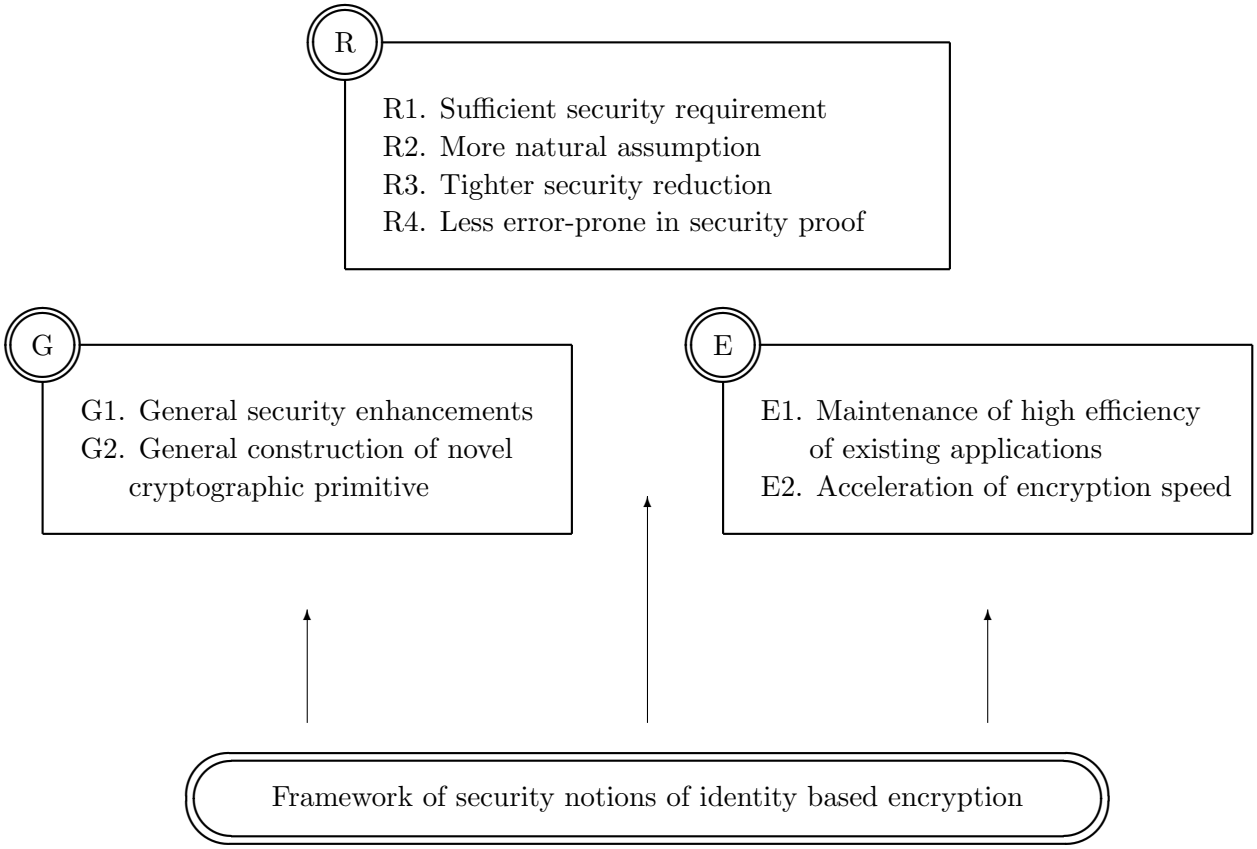


Figure 1.1: Overview of Research Motivation

Security Notions for PKE

A convenient way to formalize notions of security for cryptographic schemes is to consider combinations of various *security goals* and possible *attack models*. Three essential security goals being considered in the case of PKE are *one-wayness* (OW),*indistinguishability* (IND), *semantic security* (SS) [32], and *non-malleability* (NM) [23], i.e. $\mathsf{G}_i \in \{\mathsf{OW},\mathsf{IND},\mathsf{SS},\mathsf{NM}\}$. The attack models are the *chosen plaintext attack* (CPA) [32], the *non-adaptive chosen ciphertext attack* (CCA1) [23] and the *adaptive chosen ciphertext attack* (CCA2) [42], i.e. $\mathsf{A}_j \in \{\mathsf{CPA},\mathsf{CCA1},\mathsf{CCA2}\}$. Their combinations give nine security notions for PKE, e.g. IND-CCA2.

SS is widely accepted as the natural goal of encryption scheme because it formalizes an adversary's inability to obtain any information about the plaintext from a given ciphertext. The equivalence of SS-CPA and IND-CPA has been proved [32]; and the equivalences between SS-CCA1,2 and IND-CCA1,2 have been proven only recently [47, 31]. On the other hand, NM formalizes an adversary's inability, given a challenge ciphertext y^*, to output a different ciphertext y' in such a way that the plaintexts x and x' underlying these two ciphertexts are meaningfully related, e.g., $x' = x + 1$. The implications from IND-CCA2 to NM under any attack have been proved [5]. For these reasons, along with the convenience of proving security in the sense of IND, in almost all concrete schemes, IND-CCA2 is considered to be the "right" standard security notion for PKE.

Towards Defining Security Notions for IBE

In Boneh and Franklin's milestone paper [16], the security notions are natural extensions to the standard ones for PKE, namely indistinguishability-based ones. Actually, so far in the literature, the security notion IND-ID-CCA2 is widely considered to be the adequate one that captures the essence of security for IBE.

Due to a particular mechanism, the adversaries are granted more power in IBE than in PKE. Essentially, the adversaries can access the *key extraction oracle*, which answers the private key of any queried public key (identity). Including this particular *adaptive chosen identity attack*, we formalize the security notions for IBE, e.g., IND-ID-CCA2, in this way: G_i-ID-A_j, where $\mathsf{G}_i \in \{\mathsf{IND},\mathsf{SS},\mathsf{NM}\}$, ID denotes the particular attack mentioned above, and $\mathsf{A}_j \in \{\mathsf{CPA},\mathsf{CCA1},\mathsf{CCA2}\}$. Boneh and Franklin were the first to define the security notion for IBE, by naturally extending IND-CCA2 to IND-ID-CCA2.

Let us rigorously investigate whether IND-ID-CCA2 could be considered the "right" notion for IBE, besides the intuitive reason that it is analogous to IND-CCA2. The natural approach to justify the appropriateness for IBE is, analogously to the case of PKE, to (i) first define SS- and NM- based security notions for IBE (ii) and then establish the relations among the above security notions. To be more specific, we establish implications from IND-ID-CCA2 to all the other notions; i.e., IND-ID-CCA2 is the *strongest* security notion for IBE.

Intuition tells us that task (i) can be simply achieved by considering the analogy to the case of shifting IND-CCA to IND-ID-CCA2 as done in [16], and that task (ii) immediately

follows from the relations among the notions as in the case of PKE because we shift all the notions with the same additional attack power (namely, the accessibility to the key extraction oracle). However, we emphasize that the tasks will not follow simply and immediately until rigorous definitions for task (i) and rigorous proofs for task (ii) are presented. We accomplish both tasks in this paper.

Protection of the Top-level Secret

There are (at least[1]) two levels of secret in an IBE scheme. They are the top-level secret, which is called the master key, and the end-level secrets, which are the users' secret keys. In order to minimize damage in case of an adversary successfully expose users' secret keys, forward security [2, 8] has been introduced into IBE [19, 49, 15]. In a forward secure identity based encryption (FSIBE) scheme, the adversary can obtain no information about the compromised user's secret encrypted before the breaking-in time point.

In this paper, we focus on constructing such an *FSIBE scheme with master key update* (FSIBEm) that the top-level secret evolves as same as users' secret keys do, so that even if at some time point the adversary compromise the master key, he can no longer generate users's secret keys corresponding to passed time points. Note this attack can be mounted in all the other previous works.

On Achieving IND-ID-CCA2 Security

On the other hand, rather than building a IND-ID-CCA2 secure IBE directly, many researches of IBE schemes first build a "basic scheme" with only lower-level security, other than (IND-ID-CCA2) security, then *specifically* apply certain security enhancement to upgrade the basic scheme to a new scheme with IND-ID-CCA2 security. However, these security enhancements are proposed in the PKE environment; e.g., FOPKC [24] is known to enhance IND-CPA secure scheme, FOCRYPTO [25] enhances OW-CPA, and REACT [38] enhances OW-PCA. But, it is still unknown whether these enhancements could *generically* upgrade weak security to IND-ID-CCA2 security in IBE environment.

FOCRYPTO is used to achieve IND-ID-CCA2 security in Boneh-Franklin's paper [16] for the first time. Galindo [27] has noticed a small flawed step in the proof of [16], however the security reduction in the corrected proof was even looser. In order to achieve a better security reduction, Galindo [27] employed FOPKC. We also note that, in fact, the proof given in [16, 27] did not take account of applying generic FOPKC or FOCRYPTO transforms, but has mainly considered how to reduce the security of the "full" scheme to that of an IND-CCA2 secure PKE.

Another variant of Boneh-Franklin scheme with tighter security reduction was given by Libert and Quisquater [37], with a REACT-like appearance by adopting the KEM-DEM idea. We note that this sense of "redundancy" is not the original sense of Phan and Pointchval, since optimistically a point on a curve for bilinear pairing has a length of 171 bits, which

[1]For simplicity, here we only consider single layer IBE. And our following discussion affects HIBE case.

is slightly longer than 160 bits of necessary "redundancy". The more important thing is, again, there is no clear discussion on generic transforms for IBE in their paper, since this is not the theme of their work.

Acceleration of Encryption

Public key cryptosystems support the very foundation of the digital communication world. A lot of PKE schemes, where the public keys seem meaningless random bit-strings, are discrete-logarithm based. Since discrete-exponentiation computation is much heavier than some other computation, e.g., hash computation and symmetric key encryption/decryption, we can consider discrete-exponentiation computation consumes most of the energy of the whole system consumption. Thus, a natural idea, which is also an important research aspect, to improve such public key encryption schemes is to reduce these heavy computations.

In 2006, Bellare, Kohno and Shoup [9] introduced the concept of "stateful PKE" (SPKE), where the senders are asked to maintain some state information. Alternatively, the classical PKE schemes are called stateless PKE. Informally speaking, SPKE is a technique of randomness reusing. In such schemes, the sender maintains a state to encrypt single or multiple messages. In order to make SPKE immune from chosen ciphertext attack, the proposed scheme combined an IND-CCA secure symmetric encryption scheme SE, resulting in a hybrid encryption scheme. Thanks to this technique, the efficiency is much improved, e.g., for the ElGamal based SPKE scheme in [9], compared with the stateless counterpart DHIES [1], the exponentiation computations are reduced from two times to one time for the encryption algorithm.

In IBE research field, Phong et al. [41] proposed the first "stateful identity based encryption" (SIBE) scheme, which is based on [16]. The security reduction of their scheme is relatively loose, which means to maintain the same security level, one has to adapt longer keys. In real world, longer keys means larger group, and in consequence slower computation. They left a question of how to improve the security reduction to be tighter.

Also, as in Bellare et al.'s paper [9], symmetric key encryption is considered in both of the security model and the security proof. Can we simplify the model and the proof? We want to achieve this goal because simpler model provides clear vision on the essence of a cryptographic primitive, and shorter proof helps us evade human-error.

Towards Making Assumption Weaker

In the ElGamal based SPKE scheme in [9], the security is reduced to the gap-Diffie-Hellman (gap-DH) assumption, namely it is hard to solve the computational DH problem even with a decision oracle. In order to implement gap-DH in practice, special elliptic curves are required, e.g., supersingular curves or MNT curves, which greatly hinders the practicality of the scheme.

The security is reduced to the gap bilinear DH (gap-BDH) assumption, namely it is hard to solve a bilinear Diffie-Hellman problem with a decision oracle. Without any doubt, it is

Chapter No.	Main proposal	Reliability	Generality	Efficiency	Security model
C3	Forward Security	R1 + R2	-	↓	standard model
C4	FOIBE	R3	G1	E1	random oracle model
C5.2	Stateful IBKEM	R4	G2	E2	standard model
C5.3	Twin SIBE	R2 + R3	-	↓	random oracle model

Table 1.1: Results of Each Chapter

preferable if we can replace this gap assumption with standard assumption.

To remove the gap assumption from previous constructions, we apply the idea of twin public keys, an idea introduced by Crash, Kiltz and Shoup [21]. Furthermore, observe that the security proof of [41] is acquired by first constructing a selectively secure SIBE and then to convert the security to fully secure SIBE, such that Coron's [22] optimization technique cannot not be employed. We take this into account and manage to improve the security reduction cost of the stateful IBE scheme of [41]. We first propose a stateful IBE scheme. In addition to removing the gap assumptions from original schemes, we even get better security reductions. We then propose a stateful PKE scheme. Compared with previously previously proposed stateful PKE, the gap assumption is removed from the security proofs, while the reduction cost remains the same.

1.2 Overview of Contributions

There are several contributions in this paper.

In Chapter 2, first, we formally present the definitions of the security notions for IBE schemes. The overall definitions are built upon previous work [5, 16, 31]. With our framework, we can formalize all the security notions for IBE.

Second, we rigorously prove the relations among these notions and conclude that, IND-ID-CCA2 is the "right" security notion for IBE. Our intuition about those relations turns out to be right: the implication G_1-ID-$\mathsf{A}_1 \Rightarrow \mathsf{G}_2$-ID-$\mathsf{A}_2$ holds in IBE if and only if G_1-$\mathsf{A}_1 \Rightarrow \mathsf{G}_2$-$\mathsf{A}_2$ holds in PKE, where the corresponding security goals G_i and attack models A_j are as mentioned above.

Our results could be considered to have the same flavor as some historical results, to name just one, the equivalence between IND-CCA2 and SS-CCA2 for PKE. There, although IND-CPA and SS-CPA were defined and proved equivalent in 1984 [32], the equivalence between IND-CCA2 and SS-CCA2 was not proved rigorously until 2003 [47]. During this long period of time, people simply believed that shifting the attack power from CPA to CCA2 did not affect the equivalence.

The main results of the remaining chapters can be listed in Table 1.1. Details of reliability, generality and efficiency are given in Figure 1.1.

In Chaptper 3, our third contribution is that we combined Waters' HIBE (Waters) [48] and

Boneh-Boyen's HIBE (BonehBoyen) [13] to a hierarchical FSIBE. We employed Waters as the identity hierarchy and BonehBoyen as the time hierarchy.

In Chaptper 4, our fourth contribution is that we prove these conversions (FOPKC, FOCRYPTO, REACT) can be applied to IBE *generically* with polynomial security reductions. But in IBE, the reductions of FOPKC and FOCRYPTO turn significantly worse than in PKE. Recall that in the conventional public key setting, FOPKC conversion can be proven with a "tight" security reduction to its underlying primitives.

Under this circumstance, we propose a slight modification of FOPKC and FOCRYPTO conversions. Thanks to this modification, we can partially overcome the problem, say, we can obtain better security reductions. The modification is very simple and computationally efficient: just hash the user's identity with other inputs to the random oracle. However, this simple idea actually works! Both the modified FOPKC and the modified FOCRYPTO admit exactly much tighter reductions as their public key counterparts. This is our fifth contribution.

On the other hand, the plain REACT already gives a good reduction cost, without any modification. Interestingly, these results may indicate a separation between the chosen plaintext attack (CPA) and plaintext checking attack (PCA) in the IBE setting.

Our sixth contribution is that in order to intuitively explain how our modification improves the security reduction, we further choose proper concrete parameters, and estimate the average running time of the simulator. For the chosen parameters, using a single PC (or a single dedicated hardware), an IND-ID-CCA2 adversary breaks the IND-ID-CCA2 security of "basic Boneh-Franklin scheme + plain FOPKC conversion" with about 10^{24} years in addition to break the IND-ID-CPA security of the basic Boneh-Franklin scheme. This is to say this additional time in plain FOPKC conversion is unacceptable in the realistic world. On the other hand, it needs only additional 10^8 or 10^9 years in the case of the modified FOPKC conversion. Consider possible paralleled computing, say 1 million personal computers, this value decreases to $10^2 \sim 10^3$ years. Furthermore, after applying Moore's law, in 15 years, this value will decrease to 1.30 years, which is acceptable.

In Chaptper 5, our seventh contribution is that we introduce a simpler primitive called *stateful identity based KEM* (SIBKEM), which eventually enables a modular design approach for SIBE schemes, together with IND-CCA secure symmetric encryption. We formally give a composition theorem for such approach. Next, we give a generic construction for SIBKEM based on so-called *identity based non-interactive key exchange* (IBNIKE). As its name suggests, an IBNIKE scheme is a non-interactive key exchange scheme that two players set up their shared key. Our construction is in a totally black-box manner: given any IBNIKE scheme, we can construct an SIBKEM scheme without essential modifications of the algorithms nor resorting to random oracles.

Our eighth contribution is that, we demonstrate several instantiations of our generic constructions and compare them with known stateful PKE schemes. Since our generic constructions make no number-theoretic assumptions, one can even construct SIBE schemes without pairings assumptions, with a cost of efficiency lost during secret key extraction.

Finally, we compare our proposal with previous SIBE schemes. We conclude that efficient instantiations of our generic construction are competitive to the most efficient schemes in the literature.

1.3 Preliminary

In this chapter, we review the model of IBE and review several important concepts. We also define some necessary notations.

1.3.1 Conventions

Notations. We use $\vec{x} \leftarrow \mathcal{D}(param, sk, \vec{y})$ to denote that the vector $\vec{x}$ is made up of the plaintexts corresponding to every ciphertext in the vector $\vec{y}$. The term $\hat{\mathcal{M}}$ denotes a subset of message space $\mathcal{M}$, where the elements of $\hat{\mathcal{M}}$ are distributed according to the distribution designated by some algorithm. The function $h : \hat{\mathcal{M}} \rightarrow \{0,1\}^*$ denotes the a-priori partial information about the plaintext, and the function $f : \hat{\mathcal{M}} \rightarrow \{0,1\}^*$ denotes the a-posteriori partial information. $[a]^b$ denotes the first b bits of a string a, and $[a]_b$ denotes the last b bits of a string a.

Experiments. Let A be a probabilistic algorithm, and let $A(x_1, \ldots, x_n; r)$ be the result of running A on inputs $(x_1, \ldots, x_n)$ and coins r. Let $y \leftarrow A(x_1, \ldots, x_n)$ denote the experiment of picking r at random, and let y be $A(x_1, \ldots, x_n; r)$. If S is a finite set, then let $x \leftarrow S$ denote the operation of picking an element randomly and uniformly from S. If α is neither an algorithm nor a set, then let $x \leftarrow \alpha$ denote a simple assignment statement. We say that y can be output by $A(x_1, \ldots, x_n)$ if there is some r such that $A(x_1, \ldots, x_n; r) = y$.

Negligible Function. We say that a function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* if for every constant $c \geq 0$, an integer k_c exists such that $\epsilon(k) < k^{-c}$ for all $k > k_c$.

Random Oracle Model. The random oracle model [6] is an idealized security model to validate the security of certain natural cryptographic constructions, by providing all parties (good and bad alike) with a random function H from strings to strings. Roughly speaking, a random oracle is a function $H : X \rightarrow Y$ chosen at random and uniformly from the set of all functions $\{h : \{0,1\}^* \rightarrow \{0,1\}^\infty\}$. An algorithm can query the random oracle at any point $x \in X$ and receive the value $H(x)$ in response.

R-related Relation. We consider the R-related relation of arity t, where t is polynomial in the security parameter k. Rather than writing $R(x_1, x_2, \ldots, x_t)$, we write $R(x, \vec{x})$, denoting that the first argument is special and bunching the others into a vector $\vec{x}$ where $\left|\vec{x}\right| = t - 1$ and for every $x_i \in \vec{x},\ R(x, x_i)$ holds.

γ-uniformity. A property γ-uniformity is originally defined for conventional public key encryption schemes [25]. Here, we define γ-uniformity for IBE schemes. Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IBE scheme. For a given $id \in \{0,1\}^*$, the corresponding decryption key sk, $x \in \mathcal{M}$ and $y \in \mathcal{C}$, we define $\gamma(param, id, x, y) = \Pr[\sigma \leftarrow \mathsf{COIN}(k) : y = \mathcal{E}(param, id, x; \sigma)]$. We say that Π is γ-uniform, if for any $id \in \{0,1\}^*$, any $x \in \mathcal{M}$ and any $y \in C$, we have $\gamma(param, id, x, y) \leq \gamma$.

1.3.2 Security Models

Algorithms of IBE

Formally, an identity based encryption scheme consists of four algorithms, i.e. $\mathcal{IBE} = (\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D})$, where

- $\mathcal{S}$, the setup algorithm, takes a security parameter k and outputs system parameters $param$ and a master-key, mk. The system parameters include a description of a message space $\mathcal{M}$ and a description of a ciphertext space $\mathcal{C}$. The system parameters should be publicly known, while the mk should be known only by the "private key generator" (PKG).

- $\mathcal{X}$, the extract algorithm, takes three inputs, $param, mk$, and an arbitrary string $id \in \{0,1\}^*$, and outputs a private key, $sk = \mathcal{X}(param, mk, id)$. Here, id will be used as the public encryption key, and sk is the corresponding private decryption key. Intuitively, this algorithm extracts the private key from a given public key.

- $\mathcal{E}$, the encrypt algorithm, takes three inputs, $param$, $id \in \{0,1\}^*$, and a plaintext $x \in \mathcal{M}$. It outputs the corresponding ciphertext $y \in \mathcal{C}$. Since this algorithm might possibly be a probabilistic algorithm, it might take a random seed σ as additional input. The random seed is picked up from a coin space $\mathsf{COIN}(k)$. That is $\sigma \leftarrow \mathsf{COIN}(k)$.

- $\mathcal{D}$, the decrypt algorithm, takes three inputs, $param$, $y \in \mathcal{C}$, and the corresponding private key sk. It outputs $x \in \mathcal{M}$.

The four algorithms must satisfy the standard consistency constraint; i.e., if sk is the private key generated by the extract algorithm with the given id as the public key, then $\forall x \in \mathcal{M} : \mathcal{D}(param, sk, y) = x$, where $y = \mathcal{E}(param, id, x)$.

Algorithms of Stateful IBE

A SBE scheme is specified by five algorithms, i.e., $\mathcal{SIBE}$= (Setup, Ext, NwSt, Enc, Dec).

- Setup: The setup algorithm produces the global system parameter $param$ and the secret global master key mk from the security parameter λ. We write $(param, mk) \leftarrow \mathsf{Setup}(1^\lambda)$.

- Ext: Any recipient who wants to decrypt has to verify himself to the trusted third party, called private key generator (PKG). PKG runs the key extraction algorithms on input $param, mk$ and the user's identity id. We write $sk_{id} \leftarrow \mathsf{Ext}(param, mk, id)$.
- NwSt: The value of sender's statement st is initially generated by the new sate algorithm NwSt. We write $st \leftarrow \mathsf{NwSt}(param)$.
- Enc: The encryption algorithm Enc computes the corresponding ciphertext of a plaintext. We write $C \leftarrow \mathsf{Enc}(param, id, st, m)$.
- Dec$(param, sk, C)$: The decryption algorithm recovers the plaintext from the a ciphertext. We write $m \leftarrow \mathsf{Dec}(param, sk_{id}, C)$.

Algorithms of SPKE

An SPKE scheme is specified by five algorithms. $\mathcal{SPKE}$ = {Setup, KeyGen, NwSt, Enc, Dec}, where

Setup: The randomized setup algorithm takes as input security parameter 1^λ where $\lambda \in \mathbb{N}$. It outputs the system parameters sp. It also specifies the message space $\mathcal{M}$ by sp. ($\mathcal{M}$ may be included in sp.) We write $sp \leftarrow \mathsf{Setup}(1^\lambda)$.

KeyGen: The (possibly randomized) key generation algorithm takes as input sp. It outputs a key pair (pk, sk), where pk is a public key and sk is the corresponding secret key of pk. pk will be published to every participant in the system, while sk will be securely sent to its owner. We write $(pk, sk) \leftarrow \mathsf{KeyGen}(sp)$.

NwSt: The randomized new state algorithm takes as input sp. It outputs a new state st of a sender. We write $st \leftarrow \mathsf{NwSt}(sp)$.

Enc: The randomized encryption algorithm computes the corresponding ciphertext c of a plaintext m on sp, pk and st, where pk is the receiver's public key. We write $c \leftarrow \mathsf{Enc}(sp, pk, st, m)$.

Dec: The deterministic decryption algorithm recovers the plaintext m from the a ciphertext c on sp and sk. We write $m \leftarrow \mathsf{Dec}(sp, sk, c)$.

Model of Symmetric Encryption

Here, we simply review the definition and security requirements of symmetric encryption (SE).

An SE scheme consists of three algorithms, $\mathcal{SE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$. The randomized key generation algorithm K takes as input the security parameter λ and outputs a session key dk. We write $dk \leftarrow \mathsf{K}(\lambda)$. The (possibly randomized) encryption algorithm E takes as input a session key dk and a plaintext m and computes a ciphertext C. We write $C \leftarrow \mathsf{E}(dk, m)$. The decryption algorithm D takes as input a session key dk and a ciphertext C and outputs a plaintext m

(or "$\perp$" for invalid). We write $m/\perp \leftarrow \mathsf{D}(dk, C)$. The standard consistency constraint is that $\forall dk : m \leftarrow \mathsf{D}(dk, \mathsf{E}(dk, m))$.

Symmetric encryption scheme must guarantee indistinguishability against chosen ciphertext attack. We establish an IND-CCA game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ, runs K to obtain a random key dk, and flips a coin $b \leftarrow \{0, 1\}$.

Query: $\mathcal{A}$ issues two types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Left-or-right queries on two messages (m_0, m_1). $\mathcal{C}$ responds with ciphertext $C \leftarrow \mathsf{E}(dk, m_b)$.

$\diamond$ Decrypt-or-reject queries on a ciphertext C. If $b = 1$, then $\mathcal{C}$ responds with the message $m \leftarrow \mathsf{D}(dk, C)$; otherwise $\mathcal{C}$ responds with $\perp$. The restriction is that C must be different from the output from left-or-right queries.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

$\mathcal{A}$'s advantage in this IND-CCA game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SE scheme is secure if the advantage is negligible for any PPT algorithm $\mathcal{A}$. In this paper, we require SE to be multiple time secure, and such SE schemes can be generically built from standard block ciphers and message authentication codes (MAC) [10].

Model of Identity Based Non-interactive Key Exchange

IBNIKE is not a new concept, since it is only a natural extension of its PKI counterpart. The first IBNIKE was proposed by Sakai, Ohgishi and Kasahara [43]. We first review the model of IBNIKE, and then define two security notions. However, here we shall give the definition of type 2 first, since it is more complicated and type 1 security can be viewed as a special case of type 2.

Algorithms. An identity based non-interactive key exchange scheme is specified by three algorithms. $\mathcal{IBNIKE} = \{\mathsf{Setup}, \mathsf{Ext}, \mathsf{Shr}\}$, where

Setup: The randomized setup algorithm takes as input security parameter 1^λ where $\lambda \in \mathbb{N}$. It outputs the system parameters sp and the master key mk. It also specifies the shared key space $\mathcal{SHK}$ by sp. ($\mathcal{SHK}$ may be included in sp.) We write $(sp, mk) \leftarrow \mathsf{Setup}(1^\lambda)$.

Ext: The (possibly randomized) extract algorithm takes as input sp, mk and an identity $id \in \{0, 1\}^n$. It outputs a secret key sk_{id} corresponding to id. We write $sk_{id} \leftarrow \mathsf{Ext}(sp, mk, id)$.

Shr: The deterministic sharing algorithm takes as inputs sp, a private key sk_{id_A} and a user's identities id_B, where $id_A \neq id_B$. It outputs the shared key $K_{A,B} \in \mathcal{SHK}$ between A and B. This algorithm has symmetry. We write $K_{A,B} \leftarrow \mathsf{Shr}(sp, sk_{id_A}, id_B) = \mathsf{Shr}(sp, sk_{id_B}, id_A)$.

Type 2 Security. We establish the T2-IND (type 2 indistinguishability, i.e., indistinguishability against adaptive chosen identity attack and adaptively reveal attack) game for IBNIKE between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of IBNIKE. It passes the resulting system parameter sp to $\mathcal{A}$ and keeps the master key mk to himself.

Phase 1: $\mathcal{A}$ issues two types of oracle queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Extraction queries on an identity id. $\mathcal{C}$ responds with a corresponding secrete key sk_{id}.

$\diamond$ Reveal queries on a pair of identities (id_1, id_2), $\mathcal{C}$ responds with the key $K_{1,2}$ shared between these two identities.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides phase 1 is over, he outputs two target identities id_A, id_B, with restriction that pair (id_A, id_B) has not appeared in previous reveal queries, and neither id_A nor id_B has appeared in previous extraction queries. Then $\mathcal{C}$ flips a coin $b \in \{0, 1\}$. If $b = 0$, $\mathcal{C}$ returns $\mathcal{A}$ a random value from key space $\mathcal{SHK}$; otherwise $\mathcal{C}$ returns the real key $K_{A,B}$.

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Extraction queries on an identity $id \notin \{id_A, id_B\}$. $\mathcal{C}$ responds as in phase 1.

$\diamond$ Reveal queries on a pair of identities $(id_1, id_2) \neq (id_A, id_B)$,$\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

We refer to such an adversary $\mathcal{A}$ as a T2-IND adversary. $\mathcal{A}$'s advantage in this T2-IND security game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b' = b] - 1/2|$. We say that an IBNIKE scheme is secure in the sense of T2-IND if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

Type 1 Security. The T1-IND (type 1 indistinguishability, i.e., indistinguishability against adaptive chosen identity attack) game for IBNIKE is between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. It is similar to T2-IND, but $\mathcal{A}$ can issue only extraction queries, thus $\mathcal{A}$ is not permitted to issue reveal queries.

Let b be the random coin flipped by $\mathcal{C}$ and b' be the output of $\mathcal{A}$. $\mathcal{A}$'s advantage in this T1-IND security game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b' = b] - 1/2|$. We say that an IBNIKE scheme is secure in the sense of T1-IND if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

1.3.3 Complexity Assumptions

Diffie-Hellman assumption. Let g be a group generator of group $\mathbb{G}$ of prime order p. Let $x, y \leftarrow \mathbb{Z}_p^*$. The Diffie-Hellma assumption is that when given g^x and g^y, any probabilistic polynomial time (PPT) algorithm $\mathcal{A}_{\mathbb{G}}$ cannot compute g^{xy} with non-negligible probability. That is, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{dh}}(\mathcal{A}_{\mathbb{G}})$ is negligible.

Gap Diffie-Hellman assumption. The gap Diffie-Hellman assumption requires it is hard to compute g^{xy} even with oracle access to a Diffie-Hellman decision oracle $\mathcal{O}(\cdot,\cdot,\cdot)$ in $\mathbb{G}$. That is, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{gdh}}(\mathcal{A}_{\mathbb{G}}^{\mathcal{O}})$ is negligible. Here, when queried by $\langle g^x, g^y, z\rangle$, $\mathcal{O}$ outputs 1 when $z = g^{xy}$, or outputs 0 otherwise.

Strong twin Diffie-Hellman assumption. The strong twin Diffie-Hellman assumption requires it is hard to compute $(g^{x_1 y}, g^{x_2 y})$ even with oracle access to Diffie-Hellman decision oracle $\mathcal{O}(x_i,\cdot,\cdot)$ where $i = 1, 2$. That is, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{2dh}}(\mathcal{B}_{\mathbb{G}}^{\mathcal{O}})$ is negligible. Here, when queried by $\langle g^{x_i}, g^y, z_i\rangle$, $\mathcal{O}$ outputs 1 when $z_i = g^{x_i y}$, or outputs 0 otherwise.

Relation between DH and 2DH. As proved in [21], the DH assumption holds if and only if the strong twin DH assumption holds. That is, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{2dh}}(\mathcal{B}_{\mathbb{G}}^{\mathcal{O}}) \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{dh}}(\mathcal{A}_{\mathbb{G}}) + Q_d/p$. Here, $\mathcal{B}$ makes at most Q_d queries to $\mathcal{O}$.

Bilinear Diffie-Hellman assumption. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties: (i) *Bilinearity*: For all $x, y \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}$, $e(x^a, y^b) = e(x, y)^{ab}$. (ii) *Non-degeneracy*: $e(g, g) \neq 1$. (iii) *Computability*: There is an efficient algorithm to compute $e(x, y)$ for any $x, y \in \mathbb{G}_1$.

The bilinear Diffie-Hellman (BDH) assumptions is that when given $\langle g^x, g^y, g^w\rangle$ there is no PPT algorithm $\mathcal{A}_{\mathbb{G}_1}$ can compute $e(g, g)^{xyw}$ with non-negligible probability. That is, $\mathbf{Adv}_{\mathbb{G}_1}^{\mathrm{bdh}}(\mathcal{A}_{\mathbb{G}_1})$ is negligible.

Gap bilinear Diffie-Hellman assumption. The gap bilinear Diffie-Hellman assumption is that there is no PPT algorithm $\mathcal{A}_{\mathbb{G}_1}$ can compute $e(g, g)^{wxy}$ with non-negligible probability. even with oracle access to a decision BDH oracle $\mathcal{O}(\cdot,\cdot,\cdot,\cdot)$. That is, $\mathbf{Adv}_{\mathbb{G}_1}^{\mathrm{gbdh}}(\mathcal{A}_{\mathbb{G}_1}^{\mathcal{O}})$ is negligible. Here, when queried by $\langle g^x, g^y, g^w, z\rangle$, $\mathcal{O}$ outputs 1 when $z = e(g, g)^{xyw}$, or outputs 0 otherwise.

Strong twin bilinear Diffie-Hellman assumption. The strong twin bilinear Diffie-Hellman assumption requires it is hard to compute $(e(g, g)^{x_1 yw}, e(g, g)^{x_2 yw})$ even with oracle access to Diffie-Hellman decision oracle $\mathcal{O}(x_i,\cdot,\cdot,\cdot)$ where $i = 1, 2$. That is, $\mathbf{Adv}_{\mathbb{G}_1}^{\mathrm{2bdh}}(\mathcal{B}_{\mathbb{G}_1}^{\mathcal{O}})$ is negligible. Here, when queried by $\langle g^{x_i}, g^y, g^w, z_i\rangle$, $\mathcal{O}$ outputs 1 when $z_i = e(g, g)^{x_i yw}$, or outputs 0 otherwise.

Relation between BDH and 2BDH. As proved in [21], the BDH assumption holds if and only if the strong twin BDH assumption holds. That is, $\mathbf{Adv}^{\mathrm{2bdh}}_{\mathbb{G}_1}(\mathcal{B}^{\mathcal{O}}_{\mathbb{G}_1}) \leq \mathbf{Adv}^{\mathrm{bdh}}_{\mathbb{G}_1}(\mathcal{A}_{\mathbb{G}_1}) + Q_d/p$. Here, $\mathcal{B}$ makes at most Q_d queries to $\mathcal{O}$.

Chapter 2 FRAMEWORK OF SECURITY NOTIONS

Before we discuss and prove security,
we should first define and systematize security notions.

This chapter provides formal definitions of security notions for IBE schemes. This may be considered as a meaningful contribution of this paper, not only because this chapter formalize all the common security notions for IBE schemes, but also because the following chapters of this paper are based on the formalization of this chapter.

2.1 Introduction

Although the basic concept of IBE was proposed by Shamir [44] more than two decades ago, only very recently was the first fully functional scheme proposed [16, 17]. In 2001, Boneh and Franklin defined a security model and gave the first fully functional solution provably secure in the random oracle model [6]. The security notions proposed in their work are natural extensions to the standard ones for PKE, namely indistinguishability-based ones. Subsequent researches further extended to the standard model [20, 14, 13, 48]. Actually, so far in the literature, the security notion IND-ID-CCA2 is widely considered to be the adequate one that captures the essence of security for IBE.

As mentioned above, IND-ID-CCA2 is widely believed to be the "right" security notion for IBE. However, this issue has not been investigated rigorously, *yet*. In this work we aim to establish such an affirmative justification.

Before discussing how to define and achieve the "right" security notion for IBE, we first review the case of IBE.

2.1.1 Security Notions for PKE

A convenient way to formalize security notions for cryptographic schemes is to consider combinations of various *security goals* and possible *attack models*. Three essential security goals being considered in the case of PKE are *indistinguishability* (IND), *semantic security* (SS) [32], and *non-malleability* (NM) [23], i.e. $\mathsf{G}_i \in \{\mathsf{IND},\mathsf{SS},\mathsf{NM}\}$. The attack models are the *chosen plaintext attack* (CPA) [32], the *non-adaptive chosen ciphertext attack* (CCA1) [23] and the *adaptive chosen ciphertext attack* (CCA2) [42], i.e. $\mathsf{A}_j \in \{\mathsf{CPA},\mathsf{CCA1},\mathsf{CCA2}\}$. Their combinations give nine security notions for PKE, e.g. IND-CCA2.

SS is widely accepted as the natural goal of encryption scheme because it formalizes an adversary's inability to obtain any information about the plaintext from a given ciphertext.

The equivalence of SS-CPA and IND-CPA has been proved [32]; and the equivalences between SS-CCA1,2 and IND-CCA1,2 have been proven only recently [47, 31]. On the other hand, NM formalizes an adversary's inability, given a challenge ciphertext y^*, to output a different ciphertext y' in such a way that the plaintexts x and x' underlying these two ciphertexts are meaningfully related, e.g., $x' = x + 1$. The implications from IND-CCA2 to NM under any attack have been proved [5]. For these reasons, along with the convenience of proving security in the sense of IND, in almost all concrete schemes, IND-CCA2 is considered to be the "right" standard security notion for PKE.

2.1.2 Towards Defining Security Notions for IBE

Due to a particular mechanism, the adversaries are granted more power in IBE than in PKE. Essentially, the adversaries can access the *key extraction oracle*, which answers the private key of any queried public key (identity). Including this particular *adaptive chosen identity attack*, we formalize the security notions for IBE, e.g., IND-ID-CCA2, in this way: G_i-ID-A_j, where $\mathsf{G}_i \in \{\mathsf{IND},\mathsf{SS},\mathsf{NM}\}$, ID denotes the particular attack mentioned above, and $\mathsf{A}_j \in \{\mathsf{CPA},\mathsf{CCA1},\mathsf{CCA2}\}$. Boneh and Franklin were the first to define the security notion for IBE, by naturally extending IND-CCA2 to IND-ID-CCA2.

Remark 2.1
Actually, in IBE, besides the security against adaptive chosen identity attack, there is an another kind of weaker security for IBE, called security against selective identity (sID) attack [19]. This kind of security notions are also very useful and have applications in constructing *CCA2*-secure PKE. However, with similar discussions of this paper, one can reach similar results for sID secure IBE. More details are given in the Appendix.

Let us rigorously investigate whether IND-ID-CCA2 could be considered the "right" notion for IBE, besides the intuitive reason that it is analogous to IND-CCA2. The natural approach to justify the appropriateness for IBE is, analogously to the case of PKE, to (i) first define SS- and NM- based security notions for IBE (ii) and then establish the relations among the above security notions. To be more specific, we establish implications from IND-ID-CCA2 to all the other notions; i.e., IND-ID-CCA2 is the *strongest* security notion for IBE.

Intuition tells us that task (i) can be simply achieved by considering the analogy to the case of shifting IND-CCA to IND-ID-CCA as done in Ref. [16], and that task (ii) immediately follows from the relations among the notions as in the case of PKE because we shift all the notions with the same additional attack power (namely, the accessibility to the key extraction oracle). However, we emphasize that the tasks will not follow simply and immediately until rigorous definitions for task (i) and rigorous proofs for task (ii) are presented. We accomplish both tasks in this paper.

First we define three sorts of queries.

$\mathcal{X}$-query(id) Extraction queries. Let $\langle id \rangle$ be an extraction query issued by the adversary. Then the adversary should be responded with the private key sk of id.

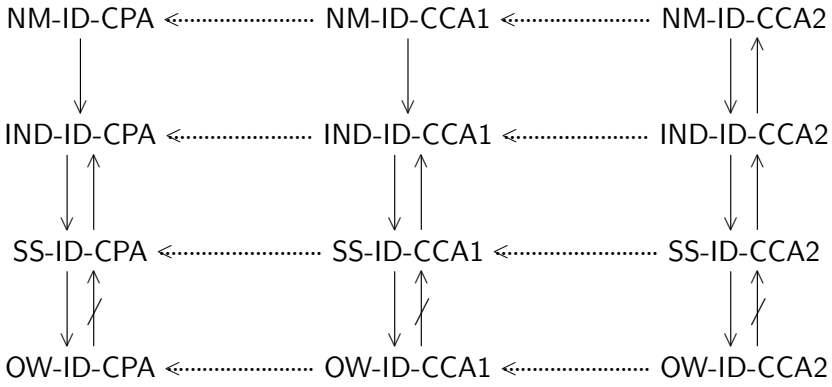


Figure 2.1: Relations among notions of security for $\mathcal{IBE}$

$\mathcal{E}$-query(id, x) Encryption queries. Let $\langle id, x\rangle$ be an encryption query issued by the adversary. Then the adversary should be responded with ciphertext y of x under public key id.

$\mathcal{D}$-query(id, y) Decryption queries. Let $\langle id, y\rangle$ be a decryption query issued by the adversary. Then the adversary should be responded with the plaintext x of y, which is encrypted by public key id.

2.1.3 Contributions

First, we formally present the definitions of the security notions for IBE schemes. The overall definitions are built upon previous work [5, 16, 31]. With our framework, we can formalize all the security notions for IBE.

Second, we rigorously prove the relations among these notions and conclude that, IND-ID-CCA2 is the "right" security notion of security for IBE. Our intuition about those relations turns out to be right: the implication G_1-ID-$\mathsf{A}_1 \Rightarrow \mathsf{G}_2$-ID-$\mathsf{A}_2$ holds in IBE if and only if G_1-$\mathsf{A}_1 \Rightarrow \mathsf{G}_2$-$\mathsf{A}_2$ holds in PKE, where the corresponding security goals G_i and attack models A_j are as mentioned above.

The results of our second contribution are illustrated in Figure 2.1. The vertical *line arrows* represent implications that are explicitly proven, and the horizontal *dot arrows* represent implications that are self-evident. In both cases, an arrow from notion **A** to notion **B** denotes that if an identity based encryption scheme is secure in the sense of **A**, then it is also secure in the sense of **B**. The scripted numbers beside the arrows denote the theorem or lemma in which the implication is proved.

Our results could be considered to have the same flavor as some historical results, to name just one, the equivalence between IND-CCA2 and SS-CCA2 for PKE. There, although IND-CPA and SS-CPA were defined and proved equivalent in 1984 [32], the equivalence between IND-CCA2 and SS-CCA2 was not proved rigorously until 2003 [47]. During this long period of time, people simply believed that shifting the attack power from CPA to CCA2 did not affect the equivalence.

2.2 Definitions of Security Notions

Let $A = (A_1, A_2)$ be an adversary. We say that A is polynomial time if both probabilistic algorithms A_1 and A_2 are polynomial time. In the first stage, given the system parameters, the adversary computes and outputs a challenge template τ. The algorithm A_1 can output some state information s, which will be transferred to A_2. In the second stage, the adversary is issued a challenge ciphertext y^* generated from τ by a probabilistic function, in a manner depending on the goal. We say that the adversary A breaks the scheme if she achieves her goal.

We consider three security goals, IND, SS and NM, and we consider three attack models, ID-CPA,ID-CCA1 and ID-CCA2, in order of increasing strength. The difference among the models is whether A_1 or A_2 is granted access to decryption oracles.

Remark 2.2
Inspecting the similarity between the adaptive chosen identity attack and the selective chosen identity attack, we only discuss in details the former case (ID security). The results can be extended to the latter case (sID security), because the strategies are similar. Roughly speaking, the target public key id should be decided by the adversary in advance, before the challenger runs the setup algorithm. The restriction is that the extraction query on id is prohibited.

Under ID-CPA the adversary can issue $\mathcal{E}$-queries to obtain ciphertexts of plaintexts of her choice. In public key cryptographic schemes, this attack is unavoidable because the adversary always gets access to the encryption function, a.k.a. encryption oracle. Under ID-CCA1, in addition to the public key, the adversary is granted access to an oracle for the decryption function, a.k.a. decryption oracle. The adversary A_1 may use this decryption function to issue $\mathcal{D}$-queries only for the period of time before she is given the challenge ciphertext y^*. (This non-adaptive attack is also named "lunchtime attack".) Under ID-CCA2, in addition to the public key, the adversary again gets access to the decryption oracle, but this time she is permitted to issue only $\mathcal{D}$-queries even on ciphertexts which are chosen after the challenge ciphertext y^* is issued. The only restriction is that A_2 may not ask for the decryption of y^*. At last, under all of the three attacks, the adversary can issue $\mathcal{X}$-queries to obtain private keys of identities, but other than the attack target identity, of her choice.

We describe in Table 2.1 and Table 2.2 the ability with which the adversary in different attack models accesses the *Extraction Oracle* $\mathcal{X}(param, mk, \cdot)$, the *Encryption Oracle* $\mathcal{E}(param, id, \cdot)$ and the *Decryption Oracle* $\mathcal{D}(param, sk, \cdot)$. The only restriction is that in ID-CCA2, A_2 must not ask the decryption oracle for the decryption of the challenge y^*.

When we say $\mathcal{O}_i = \{\mathcal{XO}_i, \mathcal{EO}_i, \mathcal{DO}_i\} = \{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \varepsilon\}$, where $i \in \{1, 2\}$, we mean that $\mathcal{DO}_i$ is a function that returns an empty string ε for any input.
*Remark 1.*To have meaningful definitions, we insist that the target public key id should not be previously queried on; i.e., the definitions are completely meaningless if the adversary already knows the corresponding private key of id.

Table 2.1: Oracle set $\mathcal{O}_1$ in definitions of the notions for IBE

	$\mathcal{O}_1 = \{\mathcal{X}\mathcal{O}_1, \mathcal{E}\mathcal{O}_1, \mathcal{D}\mathcal{O}_1\}$
ID-CPA	$\{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \varepsilon\}$
ID-CCA1	$\{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \mathcal{D}(param, sk, \cdot)\}$
ID-CCA2	$\{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \mathcal{D}(param, sk, \cdot)\}$

Table 2.2: Oracle set $\mathcal{O}_2$ in definitions of the notions for IBE

	$\mathcal{O}_2 = \{\mathcal{X}\mathcal{O}_2, \mathcal{E}\mathcal{O}_2, \mathcal{D}\mathcal{O}_2\}$
ID-CPA	$\{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \varepsilon\}$
ID-CCA1	$\{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \varepsilon\}$
ID-CCA2	$\{\mathcal{X}(param, mk, \cdot), \mathcal{E}(param, id, \cdot), \mathcal{D}(param, sk, \cdot)\}$

2.2.1 One-wayness

As far as we know, only one-wayness against full-identity chosen-plaintext attacks (referred to as OW-ID-CPA in the following definition) has been previously considered in the literature. Here we define one-wayness through a two-stage experiment. The algorithm A_1 is run on the system parameters *param* as input. At the end of A_1's execution she outputs (s, id), such that s is state information (possibly including *param*) that she wants to preserve, and *id* is the public key which she wants to attack. One plaintext x^* is *randomly* selected from the message space $\hat{\mathcal{M}}$ beyond adversary's view. A challenge y^* is computed by encrypting x^* with the public key *id*. The algorithm A_2 tries to computer what x^* was.

Definition 2.1 (OW-ID-CPA, OW-ID-CCA1, OW-ID-CCA2)

Let $\mathcal{IBE} = (\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D})$ be an identity based encryption scheme, and let $A = (A_1, A_2)$ be an adversary. For $\texttt{atk} \in \{\texttt{id-cpa}, \texttt{id-cca1}, \texttt{id-cca2}\}$ and $k \in \mathbb{N}$, let

$$\mathbf{Adv}^{\texttt{ow-atk}}_{\mathcal{IBE}}(\mathcal{A}) = \Pr[\mathbf{Exp}^{\texttt{ow-atk}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \tag{2.1}$$

where for $b, d \in \{0, 1\}$,

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\texttt{ow-atk-b}}_{\mathcal{IBE}}(\mathcal{A}) \\
\quad (param, mk) \leftarrow \mathcal{S}(k); \\
\quad (\hat{\mathcal{M}}, s, id) \leftarrow A_1^{\mathcal{O}_1}(param); \\
\quad x^* \leftarrow \hat{\mathcal{M}}; \\
\quad y^* \leftarrow \mathcal{E}(param, id, x^*); \\
\quad x' \leftarrow A_2^{\mathcal{O}_2}(s, y^*, id); \\
\quad \texttt{if } \; x' = x^* \; \texttt{ then } \; d \leftarrow 1 \; \texttt{ else } \; d \leftarrow 0; \\
\quad \texttt{return } \; d
\end{array}
$$

We say that $\mathcal{IBE}$ is secure in the sense of OW-ATK if $\mathbf{Adv}^{\texttt{ow-atk}}_{\mathcal{IBE}}(\mathcal{A})$ is negligible for any A.

2.2.2 Indistinguishability

This important security notion was first introduced by Goldwasser and Micali [32] for PKE and then described by Boneh and Franklin [16] for IBE. Here, we define indistinguishability through a two-stage experiment. The algorithm A_1 is run on the system parameters *param* as input. At the end of executing A_1, the adversary outputs (x_0, x_1, s, id) such that x_0 and x_1 are plaintexts with the same length, s is state information (possibly including *param*) that she wants to preserve, and id is the public key that she wants to attack. One of x_0 and x_1 is *randomly* selected, say x_b, beyond the adversary's view. A challenge y^* is computed by encrypting x_b with the public key id. The algorithm A_2 tries to distinguish whether y^* was the encryption of x_0 or x_1.

Definition 2.2 (IND-ID-CPA, IND-ID-CCA1, IND-ID-CCA2)

Let $\mathcal{IBE} = (\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D})$ be an identity based encryption scheme and let $A = (A_1, A_2)$ be an adversary. For $\texttt{atk} \in \{\texttt{id-cpa}, \texttt{id-cca1}, \texttt{id-cca2}\}$ and $k \in \mathbb{N}$, let

$$\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A}) = \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \tag{2.2}$$

where for $b, d \in \{0, 1\}$ and $|x_0| = |x_1|$,

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\texttt{ind-atk-b}}_{\mathcal{IBE}}(\mathcal{A}) \\
\quad (param, mk) \leftarrow \mathcal{S}(k); \\
\quad (x_0, x_1, s, id) \leftarrow A_1^{\mathcal{O}_1}(param); \\
\quad y^* \leftarrow \mathcal{E}(param, id, x_b); \\
\quad d \leftarrow A_2^{\mathcal{O}_2}(x_0, x_1, s, y^*, id); \\
\quad \texttt{return } d
\end{array}
$$

We say that $\mathcal{IBE}$ is secure in the sense of IND-ATK, if $\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A})$ is negligible for any A.

2.2.3 Semantic Security

Semantic security (for PKE) was introduced by Goldwasser and Micali [32] and later refined by Goldreich [30]. It captures the security requirement that intercepting the ciphertext gives

an adversary no partial information. We can naturally extend it to the case of IBE. A_1 is given *param* and outputs $(\hat{\mathcal{M}}, h, f, s, id)$. Here, the distribution of $\hat{\mathcal{M}}$ is designated by A_1, and $(\hat{\mathcal{M}}, h, f)$ is the challenge template τ. A_2 receives an encryption y^* of a random message x^* drawn from $\hat{\mathcal{M}}$. The adversary then outputs a value v. She hopes that $v = f(x^*)$. The adversary is successful if she can do this with a probability significantly higher than any *simulator* does. The simulator tries to do as well as the adversary without knowing the challenge ciphertext y^* or accessing any oracle.

Definition 2.3 (SS-ID-CPA, SS-ID-CCA1, SS-ID-CCA2)

Let $\mathcal{IBE} = (\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D})$ be an identity based encryption scheme, let $A = (A_1, A_2)$ be an adversary, and let $A' = (A'_1, A'_2)$ be the simulator. For $\mathtt{atk} \in \{\mathtt{id\text{-}cpa}, \mathtt{id\text{-}cca1}, \mathtt{id\text{-}cca2}\}$ and $k \in \mathbb{N}$, let

$$\mathbf{Adv}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}, \mathcal{A}') = \Pr[\mathbf{Exp}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}') = 1] \tag{2.3}$$

where for $b \in \{0, 1\}$,

$\mathtt{Experiment}\ \mathbf{Exp}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A})$
$(param, mk) \leftarrow \mathcal{S}(k)$;
$(\hat{\mathcal{M}}, h, f, s, id) \leftarrow A_1^{\mathcal{O}_1}(param)$;
$x^* \leftarrow \hat{\mathcal{M}}$;
$y^* \leftarrow \mathcal{E}(param, id, x^*)$;
$v \leftarrow A_2^{\mathcal{O}_2}(s, y^*, h(x^*), id)$;
$\mathtt{if}\ v = f(x^*)$
$\mathtt{then}\ d \leftarrow 1\ \mathtt{else}\ d \leftarrow 0$;
$\mathtt{return}\ d$

$\mathtt{Experiment}\ \mathbf{Exp}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}')$
$(\hat{\mathcal{M}}, h, f, s, id) \leftarrow A'_1(k)$;
$x^* \leftarrow \hat{\mathcal{M}}$;
$v \leftarrow A'_2(s, |x^*|, h(x^*), id)$;
$\mathtt{if}\ v = f(x^*)$
$\mathtt{then}\ d \leftarrow 1\ \mathtt{else}\ d \leftarrow 0$;
$\mathtt{return}\ d$

We say that $\mathcal{IBE}$ is secure in the sense of SS-ATK if for any adversary A a simulator exists such that $\mathbf{Adv}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}, \mathcal{A}')$ is negligible.

We comment here that in the two cases, τ must be distributed identically because both A and A' generate target public key id by themselves, i.e., τ is output individually by A and A'.

2.2.4 Non-malleability

Non-malleability was introduced by Dolev et al. [23]. It roughly requires that an adversary, given a challenge ciphertext, cannot modify it into another, different ciphertext in such a way that the plaintexts underlying the two ciphertexts are meaningfully related. A_1 is given $param$ and outputs a triple $(\hat{\mathcal{M}}, s, id)$. A_2 receives an encryption y^* of a random message x_1 drawn from $\hat{\mathcal{M}}$. The adversary then outputs a description of a relation R and a vector $\vec{y}$ of ciphertexts. We insist that $y \notin \vec{y}$. [1] The adversary hopes that $R(x_1, \vec{x})$ holds. We say that she is successful if she can do this with a probability significantly greater than that with which $R(x_0, \vec{x})$ holds. Here, x_0 is also a plaintext chosen uniformly from $\hat{\mathcal{M}}$, independently of x_1.

Definition 2.4 (NM-ID-CPA, NM-ID-CCA1, NM-ID-CCA2)

Let $\mathcal{IBE} = (\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D})$ be an identity based encryption scheme and let $A = (A_1, A_2)$ be an adversary. For $\mathtt{atk} \in \{\mathtt{id\text{-}cpa}, \mathtt{id\text{-}cca1}, \mathtt{id\text{-}cca2}\}$ and $k \in \mathbb{N}$, let

$$\mathbf{Adv}^{\mathtt{nm\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}) = \Pr[\mathbf{Exp}^{\mathtt{nm\text{-}atk\text{-}1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\mathtt{nm\text{-}atk\text{-}0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \tag{2.4}$$

where for $b \in \{0, 1\}$ and $|x_0| = |x_1|$,

$$\begin{array}{l}
\mathtt{Experiment}\ \mathbf{Exp}^{\mathtt{nm\text{-}atk\text{-}b}}_{\mathcal{IBE}}(\mathcal{A}) \\
\quad (param, mk) \leftarrow \mathcal{S}(k); \\
\quad (\hat{\mathcal{M}}, s, id) \leftarrow A_1^{\mathcal{O}_1}(param); \\
\quad x_0, x_1 \leftarrow \hat{\mathcal{M}}; \\
\quad y^* \leftarrow \mathcal{E}(param, id, x_1); \\
\quad (R, \vec{y}) \leftarrow A_2^{\mathcal{O}_2}(s, y^*, id); \\
\quad \vec{x} \leftarrow \mathcal{D}(param, id, \vec{y}); \\
\quad \mathtt{if}\ y^* \notin \vec{y}\ \wedge\ \perp \notin \vec{x}\ \wedge\ R(x_b, \vec{x}) \\
\quad\quad \mathtt{then}\ d \leftarrow 1\ \mathtt{else}\ d \leftarrow 0; \\
\quad \mathtt{return}\ d
\end{array}$$

We say that $\mathcal{IBE}$ is secure in the sense of NM-ATK, if $\mathbf{Adv}^{\mathtt{nm\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A})$ is negligible for any A.

2.3 Relations among Security Notions for IBE

In this section, we show that security proved in the sense of IND-ID-CCA2 is validly sufficient for implying security in any other sense in IBE. We first extend the relation (equivalence) between IND-ATK and SS-ATK into the IBE environment, then extend the relation

[1] The adversary is prohibited from copying the challenge ciphertext y^*. Otherwise, she could output the equality relation R, where $R(a, b)$ holds if and only if $a = b$, output $\vec{y} = \{y^*\}$, and *always* be successful.

between IND-ATK and NM-ATK into the IBE environment, at last extend the relation (separation) between IND-ATK and OW-ATK into the IBE environment. Because of these relations, the research on identity based encryption schemes has been blossoming over the past several years; thus, we say that these relations are significant.

2.3.1 Equivalence between IND and SS

Theorem 2.1 (IND-ATK $\Leftrightarrow$ SS-ATK)
A scheme $\mathcal{IBE}$ is secure in the sense of IND-ATK *if and only if $\mathcal{IBE}$ is secure in the sense of* SS-ATK*, for any attack* ATK $\in$ {ID-CPA,ID-CCA1,ID-CCA2}.

We prove this theorem by proving two directions, i.e., that IND-ATK implies SS-ATK and that SS-ATK implies IND-ATK.

Lemma 2.1 (IND-ATK $\Rightarrow$ SS-ATK)
If a scheme $\mathcal{IBE}$ is secure in the sense of IND-ATK *then $\mathcal{IBE}$ is secure in the sense of* SS-ATK*, for any attack* ATK $\in$ {ID-CPA,ID-CCA1,ID-CCA2}.

Main Idea of Proof. To clearly show the proof strategy, we describe our main idea as follows. First, according to the definition of SS, to prove that the scheme is secure in the sense of SS-ATK, we show that for any SS-ATK adversary B, a corresponding simulator B' can be constructed with oracle access to B such that B' can do as well as B in an SS-ATK game. To calculate how well the constructed simulator B' can do, we first construct an IND-ATK adversary A with oracle access to B and show that $\mathbf{Adv}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{B}, \mathcal{B}')$ is equal to $\mathbf{Adv}^{\mathtt{ind\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A})$. Because the scheme is secure in the IND-ATK sense, no matter which B is accessed as an oracle, the advantage, $\mathbf{Adv}^{\mathtt{ind\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A})$, of A to break the scheme is *always* negligible. Thus, we claim that the advantage, $\mathbf{Adv}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{B}, \mathcal{B}')$, of B to break the scheme is also negligible; i.e., B' can do as well as B. This is to say that the scheme is secure in the SS-ATK sense. The point is how to prove that the advantage of A in IND-ATK game is equal to the advantage of B in SS-ATK game.
Proof
Let $B' = (B'_1, B'_2)$, $B = (B_1, B_2)$ and $A = (A_1, A_2)$ be SS-ATK simulator, SS-ATK adversary and IND-ATK adversary, respectively. In our construction, both adversaries B and A have access to an oracle set $\mathcal{O}_1$ at their first stage and an oracle set $\mathcal{O}_2$ in their second stages, while the simulator B' has no access to any oracle.

The SS-ATK simulator B' is constructed as follows:

$\texttt{Algorithm } B_1'(k)$
$(param, mk) \leftarrow \mathcal{S}(k)$;
$(\hat{\mathcal{M}}, h, f, s, id) \leftarrow B_1^{\mathcal{O}_1}(param)$;
$\texttt{return } (\hat{\mathcal{M}}, h, f, s, id)$

$\texttt{Algorithm } B_2'(s, |x^*|, h(x^*), id)$
$x' \leftarrow \hat{\mathcal{M}}$ where $|x'| = |x^*|$;
$y' \leftarrow \mathcal{E}(param, id, x')$;
$v \leftarrow B_2^{\mathcal{O}_2}(s, y', h(x^*), id)$;
$\texttt{return } v$

The point that must be emphasized is that because the challenge template $\tau = (\hat{\mathcal{M}}, h, f)$ is chosen by B and B' themselves, τ is distributed identically in the two cases. Thus, B_1' is likely to start by generating $(mk, param) \leftarrow \mathcal{S}(k)$, where $param$ is the same as the system parameters given to B_1.

We comment that the generated master-key mk allows B' not only to *simulate* the extraction oracle, but also to extract the secret key sk corresponding to the target public key id. In this way, B' is able to *simulate* the encryption oracle and decryption oracle.

To calculate how well the simulator B' does, we construct an IND-ATK adversary A and show that $\mathbf{Adv}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}, \mathcal{B}')$ is equal to $\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A})$.

$\texttt{Algorithm } A_1^{\mathcal{O}_1}(param)$
$(\hat{\mathcal{M}}, h, f, s, id) \leftarrow B_1^{\mathcal{O}_1}(param)$;
$x_0, x_1 \leftarrow \hat{\mathcal{M}}$;
$s' \leftarrow (s, h)$;
$\texttt{return } (x_0, x_1, s', id)$

$\texttt{Algorithm } A_2^{\mathcal{O}_2}(x_0, x_1, s', y^*, id)$
where $s' = (s, h)$
$v \leftarrow B_2^{\mathcal{O}_2}(s, y^*, h(x_1), id)$;
$\texttt{if } \; v = f(x_1) \; \texttt{then } d \leftarrow 1 \; \texttt{else } d \leftarrow 0$;
$\texttt{return } d$

Note that in the experiment $\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}')$, the simulator B' invokes the SS-ATK adversary B with a *dummy* encryption y'. This experiment finally outputs 1 only when B captures a-posteriori partial information from this *dummy* encryption. On the other hand, in the experiment $\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{A})$, the adversary IND-ATK A is challenged with the ciphertext y^* corresponding to x_0, invokes B with the a-priori partial information $h(x_1)$, and finally outputs 1 only when the SS-ATK adversary B captures the a-posteriori partial information $f(x_1)$ of x_1. Thus we say in this situation that the encryption y^* is also a *dummy* for B. Hence,

$$\Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] = \Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}') = 1] \tag{2.5}$$

In contrast, in the experiment $\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{A})$, the IND-ATK adversary A is challenged with the ciphertext y^* corresponding to x_1. Focusing on our construction of A, we can see that this experiment obviously outputs 1 only when B captures a-posteriori partial information from this *useful* (no longer *dummy*) encryption; i.e., at the end of B's second stage B outputs v and $v = f(x_1)$. Hence,

$$\Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] = \Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}) = 1] \tag{2.6}$$

We obtain

$$\begin{aligned}
\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A}) &\overset{(1)}{=} \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \\
&\overset{(2)}{=} \Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}) = 1] - \Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}') = 1] \\
&\overset{(3)}{=} \mathbf{Adv}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}, \mathcal{B}')
\end{aligned}$$

Equations $\overset{(1)}{=}$ and $\overset{(3)}{=}$ are according to the definitions of advantages in IND (2.2) and SS (2.3), respectively. Equation $\overset{(2)}{=}$ holds according to Eqs. (2.5) (2.6).

Because $\mathcal{IBE}$ is secure in the IND-ATK sense we know that for the adversary A constructed by any B $\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A})$ is negligible, and hence for any B, $\mathbf{Adv}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{B}, \mathcal{B}')$ is negligible too. Thus we say the constructed simulator B' does as well as *any* adversary B. This concludes the proof of Lemma 2.1. □

Lemma 2.2 (SS-ATK ⇒ IND-ATK)
If a scheme $\mathcal{IBE}$ is secure in the sense of SS-ATK *then $\mathcal{IBE}$ is secure in the sense of* IND-ATK*, for any attack* ATK ∈ {ID-CPA,ID-CCA1,ID-CCA2}.

Main Idea of Proof. Our strategy is as follows. Towards contradiction, we prove that if a scheme is *not* secure in the IND-ATK sense, then it is *not* secure in the SS-ATK as well. So we first assume there exists an IND-ATK adversary B who can successfully break IND-ATK with an advantage that is not negligible, and then we show that we can construct an SS-ATK adversary A who can successfully break SS-ATK with an advantage that is not negligible; i.e., no SS-ATK simulator exists that can do as well as A. We do this by allowing A to call B as an oracle.

Proof

Let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be SS-ATK adversary and IND-ATK adversary respectively.

A is constructed as follows:

$\texttt{Algorithm}\ A_1^{\mathcal{O}_1}(param)$
  $(x_0, x_1, s, id) \leftarrow B_1^{\mathcal{O}_1}(param)$;
  $\hat{\mathcal{M}} \leftarrow \{x_0, x_1\}_U$;
  choose f satisfies $f(x_0) = 0$ and $f(x_1) = 1$;
  choose h satisfies $h(x_0) = h(x_1)$;
  $\texttt{return}\ (\hat{\mathcal{M}}, h, f, s, id)$

$\texttt{Algorithm}\ A_2^{\mathcal{O}_2}(s, y^*, h(x^*), id)$
  $d' \leftarrow B_2^{\mathcal{O}_2}(x_0, x_1, s, y^*, id)$;
  $v \leftarrow d'$;
  $\texttt{return}\ v$

Because either x_0 or x_1 is chosen at a probability of $1/2$, we obtain

$$\Pr[b = 0] = \Pr[b = 1] \quad = \quad \frac{1}{2} \tag{2.7}$$

Recalling the definition of advantages in IND-ATK (2.2), we obtain

$$\Pr[\mathbf{Exp}^{\texttt{ind-atk-b}}_{\mathcal{IBE}}(\mathcal{B}) = 0] + \Pr[\mathbf{Exp}^{\texttt{ind-atk-b}}_{\mathcal{IBE}}(\mathcal{B}) = 1] = 1$$

for $b \in \{0, 1\}$. Furthermore, focusing on our construction, we obtain

$$\begin{aligned}
&\Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \\
&\quad = \Pr[b = 0] \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 0] + \ \Pr[b = 1] \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 1] \\
&\quad \overset{(1)}{=} \frac{1}{2} \cdot (1 - \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 1]) + \ \frac{1}{2} \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 1] \\
&\quad \overset{(2)}{=} \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{B})
\end{aligned} \tag{2.8}$$

Here, equation $\overset{(1)}{=}$ holds according to Eqs. (2.7) (2.8). Equation $\overset{(2)}{=}$ holds according to Eq. (2.2).

On the other hand, recall the definition of SS-ATK (on Page 21). Because the challenge template τ should be distributed identically in the two cases, we observe that in the second stage of the simulator, the input values $\big(s, |x^*|, h(x^*), id\big)$ are independent of the event $x^* = x_b$, where b is chosen randomly and uniformly in $\{0, 1\}$. Hence for any simulator,

$$\Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{A}') = 1] \leq \frac{1}{2} \tag{2.9}$$

This means that A' cannot be successful at a probability more than $1/2$. In this inequality, the equality holds in case A' always outputs a value in $\{0, 1\}$.

According to the definition of advantage in SS-ATK (2.3) and Eq. (2.8) and inequality (2.9), we obtain

$$\begin{aligned}
\mathbf{Adv}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{A}, \mathcal{A}') \quad &= \quad \Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\texttt{ss-atk}}_{\mathcal{IBE}}(\mathcal{A}') = 1] \\
&\geq \quad \frac{1}{2} \cdot \mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{B})
\end{aligned}$$

We have assumed that $\mathbf{Adv}^{\mathtt{ind\text{-}atk}}_{\mathcal{IBE}}(\mathcal{B})$ is not negligible; thus, $\mathbf{Adv}^{\mathtt{ss\text{-}atk}}_{\mathcal{IBE}}(\mathcal{A}, \mathcal{A}')$ is also not negligible. We have reached a contradiction to the hypothesis that $\mathcal{IBE}$ is secure in the SS-ATK sense. Thus, $\mathcal{IBE}$ is also secure in the IND-ATK sense. This concludes the proof of Lemma 2.2.

□

Proof of Theorem 2.1 From Lemma 2.1 and Lemma 2.2, Theorem 2.1 is proven immediately. ■

2.3.2 Relation between IND and NM

Theorem 2.2 (IND-ID-CCA2 ⇒ NM-ID-CCA2)
If a scheme $\mathcal{IBE}$ is secure in the sense of IND-ID-CCA2 *then $\mathcal{IBE}$ is secure in the sense of* NM-ID-CCA2.

Main Idea of Proof. Towards contradiction, we prove that if a scheme is *not* secure in the NM-ID-CCA2 sense, then it is *not* secure in the IND-ID-CCA2 either. We first assume that an NM-ID-CCA2 adversary B exists who can break NM-ID-CCA2 with an advantage that is not negligible, then we show that we can construct an IND-ID-CCA2 adversary A who can break IND-ID-CCA2 with an advantage that is not negligible. We do this by allowing A to call B as an oracle.
Proof
Let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be IND-ID-CCA2 adversary and NM-ID-CCA2 adversary respectively.

A is constructed as follows:

$\mathtt{Algorithm}\ A_1^{\mathcal{O}_1}(param)$
$(\hat{\mathcal{M}}, s, id) \leftarrow B_1^{\mathcal{O}_1}(param)$;
$x_0 \leftarrow \hat{\mathcal{M}}; x_1 \leftarrow \hat{\mathcal{M}}$;
$s' \leftarrow (\hat{\mathcal{M}}, s)$;
$\mathtt{return}\ (x_0, x_1, s', id)$

$\mathtt{Algorithm}\ A_2^{\mathcal{O}_2}(x_0, x_1, s', id, y^*)$
where $s' = (\hat{\mathcal{M}}, s)$
$(R, \vec{y}) \leftarrow B_2^{\mathcal{O}_2}(s, y^*, id)$;
$\vec{x} \leftarrow \mathcal{D}(param, id, \vec{y})$;
$\mathtt{if}\ R(x_0, \vec{x}) \wedge \neg R(x_1, \vec{x})\ \mathtt{then}\ d \leftarrow 0$;
$\mathtt{else\ if}\ \neg R(x_0, \vec{x}) \wedge R(x_1, \vec{x})$
$\mathtt{then}\ d \leftarrow 1$;
$\mathtt{else}\ d \leftarrow \{0,1\}_U$;
$\mathtt{return}\ d$

Table 2.3: Definitions of $p(i,j)$ for $i,j \in \{0,1\}$

	$R(x_0,\vec{x})$	$R(x_1,\vec{x})$	Probability
whether	`false`	`false`	$p(0,0)$
$R(x_b,\vec{x})$	`true`	`false`	$p(1,0)$
holds	`false`	`true`	$p(0,1)$
or not	`true`	`true`	$p(1,1)$

Focusing on our construction we observe that,

$$\begin{aligned}
&\mathbf{Adv}^{\texttt{ind-id-cca2}}_{\mathcal{IBE}}(\mathcal{A}) \\
&\quad \overset{(1)}{=} \Pr[\mathbf{Exp}^{\texttt{ind-id-cca2-1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\texttt{ind-id-cca2-0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \\
&\quad \overset{(2)}{=} \Big[\ p(0,1) + \frac{1}{2} \cdot \big(\ p(0,0) + p(1,1)\ \big)\Big] - \Big[\ p(1,0) + \frac{1}{2} \cdot \big(\ p(0,0) + p(1,1)\ \big)\Big] \\
&\quad = p(0,1) - p(1,0)
\end{aligned}$$

$$\begin{aligned}
&\mathbf{Adv}^{\texttt{nm-id-cca2}}_{\mathcal{IBE}}(\mathcal{B}) \\
&\quad \overset{(3)}{=} \Pr[\mathbf{Exp}^{\texttt{nm-id-cca2-1}}_{\mathcal{IBE}}(\mathcal{B}) = 1] - \Pr[\mathbf{Exp}^{\texttt{nm-id-cca2-0}}_{\mathcal{IBE}}(\mathcal{B}) = 1] \\
&\quad \overset{(4)}{=} \big(\ p(0,1) + p(1,1)\ \big) - \big(\ p(1,0) + p(1,1)\ \big) \\
&\quad = p(0,1) - p(1,0)
\end{aligned}$$

The notations $p(i,j)$, where $i,j \in \{0,1\}$, are defined in Table 2.3. In this way we obtain equations $\overset{(2)}{=}$ and $\overset{(4)}{=}$. Equations $\overset{(1)}{=}$ and $\overset{(3)}{=}$ are according to the definitions of advantages in IND (2.2) and NM (2.4), respectively. Hence,

$$\mathbf{Adv}^{\texttt{ind-id-cca2}}_{\mathcal{IBE}}(\mathcal{A}) = \mathbf{Adv}^{\texttt{nm-id-cca2}}_{\mathcal{IBE}}(\mathcal{B})$$

Under the assumption that $\mathbf{Adv}^{\texttt{nm-id-cca2}}_{\mathcal{IBE}}(\mathcal{B})$ is not negligible, $\mathbf{Adv}^{\texttt{ind-id-cca2}}_{\mathcal{IBE}}(\mathcal{A})$ is also not negligible. We reach a contradiction to the hypothesis that $\mathcal{IBE}$ is secure in the IND-ID-CCA2 sense. Thus $\mathcal{IBE}$ is also secure in the NM-ID-CCA2 sense. This concludes the proof of Theorem 2.2. □

Theorem 2.3 (NM-ATK ⇒ IND-ATK)
If a scheme $\mathcal{IBE}$ is secure in the sense of NM-ATK *then $\mathcal{IBE}$ is secure in the sense of* IND-ATK*, for any attack* ATK ∈ {ID-CPA,ID-CCA1,ID-CCA2}.

Main Idea of Proof. Towards contradiction, we prove that if a scheme is *not* secure in the IND-ATK sense, then it is *not* secure in the NM-ATK as well. We first assume that an IND-ATK adversary B exists who can break IND-ATK with an advantage that is not negligible,

then we show that we can construct an NM-ATK adversary A who can break NM-ATK with an advantage that is not negligible. We do this by allowing A to call B as an oracle.

Proof

Let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be an NM-ATK adversary and an IND-ATK adversary.

A is constructed as follows:

$\texttt{Algorithm } A_1^{\mathcal{O}_1}(param)$
$\quad (x_0, x_1, s, id) \leftarrow B_1^{\mathcal{O}_1}(param);$
$\quad \hat{\mathcal{M}} \leftarrow \{x_0, x_1\}_U;$
$\quad s' \leftarrow (x_0, x_1, s);$
$\quad \texttt{return } (\hat{\mathcal{M}}, s', id)$

$\texttt{Algorithm } A_2^{\mathcal{O}_2}(\hat{\mathcal{M}}, s', y^*, id)$
$\qquad\qquad\qquad\qquad \text{where } s' = (x_0, x_1, s)$
$\quad d \leftarrow B_2^{\mathcal{O}_2}(x_0, x_1, s, id, y^*);$
$\quad y' \leftarrow \mathcal{E}(param, id, (x_d + 1));$
$\quad \vec{y} \leftarrow \{y'\};$
$\quad \texttt{return } (R, \vec{y})$
$\qquad \text{where } R(a, b) = 1 \texttt{ iff } a + 1 = b$

In A_1 the notation $\hat{\mathcal{M}} \leftarrow \{x_0, x_1\}_U$ denotes that $\hat{\mathcal{M}}$ is being assigned the probability space that assigns to each of x_0 and x_1 a probability of $1/2$.

Inspecting either x_0 or x_1 was randomly chosen with a probability of $1/2$, and recalling the definitions of advantages in IND (2.2) and NM (2.4), we obtain

$$\Pr[b = 0] = \Pr[b = 1] \quad = \quad \frac{1}{2} \tag{2.10}$$

$$\Pr[\mathbf{Exp}^{\texttt{ind-atk-b}}_{\mathcal{IBE}}(\mathcal{B}) = 0] + \Pr[\mathbf{Exp}^{\texttt{ind-atk-b}}_{\mathcal{IBE}}(\mathcal{B}) = 1] = 1$$

for $b \in \{0, 1\}$. Furthermore, focusing on our construction, we obtain

$$\begin{aligned}
&\Pr[\mathbf{Exp}^{\texttt{nm-atk-1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \\
&\quad = \Pr[b = 0] \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 0] + \ \Pr[b = 1] \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 1] \\
&\Pr[\mathbf{Exp}^{\texttt{nm-atk-0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \\
&\quad = \Pr[b = 0] \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 1] + \ \Pr[b = 1] \cdot \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 0]
\end{aligned}$$

The event $b = i$, where $i \in \{0, 1\}$, denotes that the challenger chose x_b, encrypted x_b and sent the corresponding ciphertext y^* as a challenge to the NM-ATK adversary A. Hence,

$$
\begin{aligned}
&\mathbf{Adv}^{\texttt{nm-atk}}_{\mathcal{IBE}}(\mathcal{A}) \\
&\quad \overset{(1)}{=} \Pr[\mathbf{Exp}^{\texttt{nm-atk-1}}_{\mathcal{IBE}}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}^{\texttt{nm-atk-0}}_{\mathcal{IBE}}(\mathcal{A}) = 1] \\
&\quad \overset{(2)}{=} \frac{1}{2} \cdot \Big\{ (\Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 0] + \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 1]) \\
&\qquad - \ (\Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 1] + \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 0]) \Big\} \\
&\quad \overset{(3)}{=} \Pr[\mathbf{Exp}^{\texttt{ind-atk-1}}_{\mathcal{IBE}}(\mathcal{B}) = 1] - \Pr[\mathbf{Exp}^{\texttt{ind-atk-0}}_{\mathcal{IBE}}(\mathcal{B}) = 1] \\
&\quad \overset{(4)}{=} \mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{B})
\end{aligned}
$$

Equations $\overset{(1)}{=}$ and $\overset{(4)}{=}$ hold according to the definitions of advantages in NM (2.4) and IND (2.2), respectively. Equation $\overset{(2)}{=}$ holds according to Eqs. (2.10) (2.11) (2.11). Equation $\overset{(3)}{=}$ holds according to Eq. (2.11).

Under the assumption that $\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{B})$ is not negligible, $\mathbf{Adv}^{\texttt{nm-atk}}_{\mathcal{IBE}}(\mathcal{A})$ is also not negligible. We reach a contradiction to the hypothesis that $\mathcal{IBE}$ is secure in the NM-ATK sense. Thus $\mathcal{IBE}$ is also secure in the IND-ATK sense. This concludes the proof of Theorem 2.3. □

2.3.3 Separation between IND and OW

Because the relation between IND and OW is straightforward and the proof is simple, we omit the detailed discussion of proof in this section, but only construct the algorithms.

Theorem 2.4 (IND-ATK ⇒ OW-ATK)
If a scheme $\mathcal{IBE}$ is secure in the sense of IND-ATK *then $\mathcal{IBE}$ is secure in the sense of* OW-ATK*, for any attack* ATK ∈ {ID-CPA,ID-CCA1,ID-CCA2}.

Main Idea of Proof. Towards contradiction, we prove that if a scheme is *not* secure in the OW-ATK sense, then it is *not* secure in the IND-ATK either. We first assume that an OW-ATK adversary B exists who can break OW-ATK with an advantage that is not negligible, then we show that we can construct an IND-ATK adversary A who can break IND-ATK with an advantage that is not negligible. We do this by allowing A to call B as an oracle.
Proof
Let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be IND-ATK adversary and OW-ATK adversary respectively.

A is constructed as follows:

$\texttt{Algorithm } A_1^{\mathcal{O}_1}(param)$
$(\hat{\mathcal{M}}, s, id) \leftarrow B_1^{\mathcal{O}_1}(param);$
$x_0 \leftarrow \hat{\mathcal{M}}; x_1 \leftarrow \hat{\mathcal{M}};$
$s' \leftarrow (\hat{\mathcal{M}}, s);$
$\texttt{return } (x_0, x_1, s', id)$

$\texttt{Algorithm } A_2^{\mathcal{O}_2}(x_0, x_1, s', id, y^*)$
where $s' = (\hat{\mathcal{M}}, s)$
$x' \leftarrow B_2^{\mathcal{O}_2}(s, y^*, id);$
$\texttt{if } x' = x_0 \texttt{ then } d \leftarrow 0;$
$\texttt{else if } x' = x_1$
$\texttt{then } d \leftarrow 1;$
$\texttt{else } d \leftarrow \{0,1\}_U;$
$\texttt{return } d$

It is simple to compute that,

$$\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A}) = \mathbf{Adv}^{\texttt{ow-atk}}_{\mathcal{IBE}}(\mathcal{B})$$

Under the assumption that $\mathbf{Adv}^{\texttt{ow-atk}}_{\mathcal{IBE}}(\mathcal{B})$ is not negligible, $\mathbf{Adv}^{\texttt{ind-atk}}_{\mathcal{IBE}}(\mathcal{A})$ is also not negligible. We reach a contradiction to the hypothesis that $\mathcal{IBE}$ is secure in the IND-ATK sense. Thus $\mathcal{IBE}$ is also secure in the OW-ATK sense. This concludes the proof of Theorem 2.4. □

Theorem 2.5 (OW-ATK $\not\Rightarrow$ IND-ATK)
If a scheme $\mathcal{IBE}$ is secure in the sense of OW-ATK *then there exits another scheme $\mathcal{IBE}'$ which is also secure in the sense of* OW-ATK*, but is not secure in the sense of* IND-ATK*, for any attack* ATK $\in$ {ID-CPA,ID-CCA1,ID-CCA2}.

Main Idea of Proof. To prove this separation from OW-ATK to IND-ATK, we employ another kind of proof technique. We construct a new scheme $\mathcal{IBE}'$ whose algorithms are converted from the original scheme $\mathcal{IBE}$. Our purpose is bare; we want to embed "some" partial information of the plaintexts into the corresponding ciphertexts, in such a way that, the embedded partial information is useless to the OW-ATK adversary while the information could help the IND-ATK adversary to break the scheme $\mathcal{IBE}'$ with a non-negligible probability.
Proof
Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an OW-ATK secure $\mathcal{IBE}$ scheme. Then, we can construct an another $\mathcal{IBE}$ scheme $\Pi' = \{\mathcal{S}', \mathcal{X}', \mathcal{E}', \mathcal{D}'\}$ such that, Pi' is secure in the OW-ATK sense but not secure in the IND-ATK sense. The construction is illustrated in Figure 2.2.

It is obvious that adding the parity-bit into the ciphertext is actually an action to add partial information of the plaintext. Although this action does not give any additional power to the OW-ATK adversary, it does give necessary power to the IND-ATK adversary

Setup $\mathcal{S}'$:
It is as $\mathcal{S}$.
Extraction $\mathcal{X}'$:
It is as $\mathcal{X}$.
Encryption $\mathcal{E}'$:
Let $id \in \{0,1\}^*$ is the encryption key and x is the plaintext.
Let $P(x) \in \{0,1\}$ denote the bitwise-parity of a plaintext x.
It computes $y \leftarrow \mathcal{E}'(param, id, x) \leftarrow \mathcal{E}(param, id, x) \| P(x)$.
Decryption $\mathcal{D}'$:
Let $y = y_1 \| y_2$ be a ciphertext to decrypt, where $\|y_2\| = 1$.
It computes $x \leftarrow \mathcal{D}'(param, sk, y) \leftarrow \mathcal{D}(param, sk, y_1)$.

Figure 2.2: Algorithms of constructed $\mathcal{IBE}$ scheme

$A = (A_1, A_2)$ to break the scheme Π'. This is because at the end of the execution of algorithms A_1, if she outputs two plaintext x_0 and x_1 such that $P(x_0) \neq P(x_1)$, then by observing the additional parity-bit in the challenge y^*, A_2 can immediately know which plaintext was encrypted by the challenger, always. This concludes the proof of Theorem 2.5. □

2.4 Conclusions

This chapter proposed the framework for identity based encryption. This framework turns out to be not only the foundation of this dissertation, but also the foundation of identity based research.

Chapter 3 A FORWARD SECURE SCHEME WITH MASTER KEY UPDATE

After compromise your secret key,
curious attackers only observe your data flow,
but malicious attackers can destroy your business.
What can save you from this terrible circumstance?

This chapter proposes an identity based encryption scheme with forward security. Especially in this scheme, the top secret, called the master key, updates as time evolves. This scheme is provably secure in the sense of FS-IND-ID-CPA based on DBDH assumption in standard model.

3.1 Introduction

Security of digital systems is becoming increasingly critical in our nowaday life. One important primitive is the identity based encryption (IBE) [44]. IBE has the maximum flexibility for assigning user's public key, i.e., any arbitrary string (identity) could be the recipient's public key.

There are (at least[1]) two levels of secret in an IBE scheme. They are the top-level secret, which is called the master key, and the end-level secrets, which are the users' secret keys. In order to minimize damage in case of an adversary successfully expose users' secret keys, forward security [2, 8] has been introduced into IBE [19, 49, 15]. In a forward secure identity based encryption (FSIBE) scheme, the adversary can obtain no information about the compromised user's secret encrypted before the breaking-in time point.

3.1.1 Related Works

One inherent weakness of IBE is the key-escrow problem, which means the trusted center, called private key generator (PKG), possesses the master key. Since the master key is used to generate secret keys corresponding to every identity, compromising the master key equals success of breaking the whole IBE scheme. We can naturally consider a sufficiently motivated adversary will try his best to expose the master key.

Although in historical works [49, 15] forward secrecy of users' secret keys has been perfectly achieved, but forward secrecy of the master key was out of concern. Actually, there exists a constant top secret stored in PKG, and that may become the weakest point of the whole scheme. In this paper, we focus on constructing such an *FSIBE scheme with master key*

[1]For simplicity, here we only consider single layer IBE. And our following discussion affects HIBE case.

update (FSIBEm) that the top-level secret evolves as same as users' secret keys do, so that even if at some time point the adversary compromise the master key, he can no longer generate users's secret keys corresponding to passed time points. Note this attack can be mounted in all the other known works.

In this paper, we focus on how to construct such FSIBEm in standard model.

3.1.2 Contributions

Our first contribution is that we combined Waters' HIBE (Waters) [48] and Boneh-Boyen's HIBE (BonehBoyen) [13] to a hierarchical FSIBE. We employed Waters as the identity hierarchy and BonehBoyen as the time hierarchy.

To achieve FSIBEm's property, we simply let the identity hierarchy be two-level, and force PKG to use a level one secret key as the actual functional master key and to delete the original unevolutional master key.

The security of our FSIBEm could be considered straightforwardly based on Waters and BonehBoyen scheme. Our FSIBEm is secure in the sense of FS-IND-ID-CPA in standard model. We stress here that the security proof is not the main contribution of this paper. We remark that because Waters and BonehBoyen are based on decisional bilinear Diffie-Hellman (DBDH) assumption and in our scheme no additional assumption is introduced, our FSIBEm is also provably secure from DBDH assumption.

Comparing with our scheme, [49] is only secure in the sense of FS-OW-ID-CPA in the random oracle model, which means [49] requires ideal cryptographic hash function, and [15] is secure in the sense of FS-IND-sID-CPA, which means [15] is weak against the adaptive chosen identity attack. Although one can raise [15] to fully security, that will greatly sacrifice security reduction, which means much longer keys have to imported to maintain security level.

3.2 Security Model of FSIBEm

3.2.1 Algorithms of FSIBEm

Definition 3.1
An FSIBEm scheme is specified by six PPT algorithms, i.e., FSIBEm ={Setup, Ext, mkUpd, skUpd, Enc, Dec}.

The functionalities are as follows:

- Setup: The setup algorithm produces the global system parameter $param$ and the initial secret global master key mk^0 from the security parameter λ and the maximum time period t. We write $(param, mk^0) \leftarrow \mathsf{Setup}(1^\lambda)$.

- Ext: At time point τ, after a recipient verifies himself to the trusted third party, called private key generator (PKG), PKG runs the key extraction algorithms on input $param, mk^\tau$ and the user's identity id. The output is the user's secret key sk_{id}^τ corresponding to current time. We write $sk_{id}^\tau \leftarrow \mathsf{Ext}(param, mk^\tau, id, \tau)$.

- mkUpd: The master key update algorithm takes input as system parameter $param$, current time index τ and the current master key mk^τ, and it evolves the master key to $mk^{\tau+1}$ for the next time period.

- skUpd: The secret key update algorithm takes input as system parameter $param$, current time index τ, user's identity id and the current secret key sk_{id}^τ, and it evolves the secret key to $sk_{id}^{\tau+1}$ for the next time period.

- Enc: The encryption algorithm Enc computes the corresponding ciphertext of a plaintext. We write $C \leftarrow \mathsf{Enc}(param, id, m, \tau)$.

- Dec: The decryption algorithm recovers the plaintext from the a ciphertext. We write $m \leftarrow \mathsf{Dec}(param, sk_{id}^\tau, C, \tau)$.

For simplicity, we assume the evolution of the master key and users's secret keys is synchronized. And it is easy to have keys evolve with their own frequency.

3.2.2 Security Notion of FSIBEm

Here, we define FS-IND-ID-CPA game.

Definition 3.2
An FSIBEm scheme is FS-IND-ID-CPA secure if for all polynomial $N(\cdot)$, the advantage of any PPT adversary in the following game is negligible.

Setup: The challenger runs Setup on security parameter $lambda$ and maximum time period 2^t. It passes system parameter $param$ to the adversary and keeps master key mk^0 to itself.

Queries: The adversary issues sk-breakin(id, i) queries, mk-breakin(j), challenge(id^*, m_0, m_1, k) query, with restriction that $0 \leq k < j < N$, and if $id = id^*$ then $0 \leq k < i < N$. These queries are answered as follows:

sk-breakin(id, i)**:** The challenger first runs Ext to compute key sk_{id}^0, and runs skUpd to compute $sk_{id}^{\tau_i}$ as the result.

mk-breakin(j)**:** The challenger runs mkUpd to compute key mk^{τ_j} as the result.

challenge(id^*, m_0, m_1, k)**:** The challenger picks a random bit b and compute $C^* \leftarrow \mathsf{Enc}(param, id^*, m_b, \tau_k)$ as the result.

Guess: The adversary outputs a guess $b' \in \{0, 1\}$. His advantage to win the game is $\mathbf{Adv}^{\texttt{fs-ind-id-cpa}}_{\mathsf{FSIBEm}}(\mathcal{A}) = |\Pr[b' = b] - 1/2|$.

$$mk^{\tau} = \left(\begin{array}{ll} g_2^{\alpha} \cdot h_{mk}^{r} \cdot (g_1^{\tau_{t-1}} \cdot f_1)^{r_1} \cdot (g_1^{\tau_{t-2}} \cdot f_1)^{r_2} \cdots (g_1^{\tau_0} \cdot f_1)^{r_t}, & \\ g^{r}, & \\ g^{r_1}, g^{r_2}, \cdots, g^{r_t}, & \\ g_2^{\alpha} \cdot h_{mk}^{r} \cdot (g_1^{\tau_{t-1}} \cdot f_1)^{r_1} \cdot (g_1^{\tau_{t-2}} \cdot f_1)^{r_2} \cdots (g_1^{\tau_1} \cdot f_1)^{r_{t-1}} \cdot (g_1 \cdot f_t)^{r'_t}, & \text{if } \tau_0 = 0 \\ g_2^{\alpha} \cdot h_{mk}^{r} \cdot (g_1^{\tau_{t-1}} \cdot f_1)^{r_1} \cdot (g_1^{\tau_{t-2}} \cdot f_1)^{r_2} \cdots (g_1 \cdot f_{t-1})^{r'_{t-1}}, & \text{if } \tau_1 = 0 \\ \vdots & \vdots \\ g_2^{\alpha} \cdot h_{mk}^{r} \cdot (g_1^{\tau_{t-1}} \cdot f_1)^{r_1} \cdot (g_1 \cdot f_2)^{r'_2}, & \text{if } \tau_{t-1} = 0 \\ g_2^{\alpha} \cdot h_{mk}^{r} \cdot (g_1 \cdot f_1)^{r'_1}, & \text{if } \tau_{t-1} = 0 \\ g^{r'_1}, g^{r'_2}, \cdots, g^{r'_t}, & \text{restricted to the } g^{r'_i} \text{ such that } \tau_{t-i} = 0 \end{array} \right)$$

Figure 3.1: General form of master key at time period τ.

3.3 A FSIBEm Scheme Based on DBDH Assumption in Standard Model

In this section, we construct our scheme by using two-dimension of HIBE. To be concrete, the first dimension has two-levels and is used for root and user identity hierarchy; while the second one has $\log(T)$ level and is used for time hierarchy in the same manner as Canetti-Halevi-Katz [19] binary tree encryption, where T is the maximum time period. The first HIBE is instantiated by Waters and the second one is by BonehBoyen.

3.3.1 Construction

Let $\mathbb{G}$ be a bilinear group of prime order p, where p is determined by the security parameter. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map. We compute Waters' hash on identities which are bit-strings of length n. As same as all the other identity based encryption schemes, we can employ collision-resistant hash function to expand the identity space. Our scheme is described as follows.

Setup$(1^{\lambda}, 2^t)$**:** It takes inputs of a security parameter λ and a number t representing that the maximum time periods of the scheme is 2^t. Select g as a corresponding generator of $\mathbb{G}$, a random $\alpha \in \mathbb{Z}_p$ and set $g_1 \leftarrow g^{\alpha}$. Choose random elements $(g_2, f_1, \cdots, f_t) \in \mathbb{G}^{t+1}$. Choose random value $(u'_1, u'_2) \in \mathbb{G}^2$ and two random n-length vector $U_j \leftarrow (u_{j,i})$, where $j \in \{1, 2\}$ and the elements of U_j are uniformly distributed in $\mathbb{G}$. Let h_{mk} denote running Waters' hash on the root identity, e.g., the domain of a company. Thus, $h_{mk} \leftarrow u'_1 \prod_{i \in \mathcal{V}} u_{1,i}$, where $\mathcal{V}$ is the set of indicies for which the root identity is set to 1. It generates the master key mk^0 at time-period $\tau = \langle 0| \cdots |0 \rangle$ as in Figure 3.1.

Finally, it deletes α and publishes information except the master key as system parameter.

Ext$(param, mk^{\tau}, id, \tau)$**:** The secret key extraction algorithm generates secret key sk_{id}^{τ} for certified user with identity id. Let h_{id} denote running Waters' hash on the user's

identity, e.g., the user's email address. Thus, $h_{id} \leftarrow u_2' \prod_{i \in \mathcal{W}} u_{2,i}$, where $\mathcal{W}$ is the set of indicies for which the user's identity is set to 1. It works as follows: (1) Pick up random $r^* \leftarrow \mathbb{Z}_p$ and compute $y^* \leftarrow h_{id}^{r^*}$, $a_2 \leftarrow g^{r^*}$. (2) Parse mk^τ as $(a_0, a_1, b_1, \cdots, b_t, c_t, \cdots, c_1, d_1, \cdots, d_t)$, where unless $\tau_{t-i} = 0$. Set $c_i = \perp$,$d_i = \perp$. (3) Output $sk_{id}^\tau \leftarrow (a_0 \cdot y^*, a_1, a_2, b_1, \cdots, b_t, c_t \cdot y^*, \cdots, c_1 \cdot y^*, d_1, \cdots, d_t)$.

mkUpd$(param, mk^\tau, \tau)$**:** It evolves the master key to $mk^{\tau+1}$ for the next time period. The essential part is to use the current BonehBoyen time point to generate cover set for time periods $[\tau + 1, 2^t - 1]$. Due to space limitation, we leave the transformation to the full version.

skUpd$(param, sk_{id}^\tau, id, \tau)$**:** This algorithm computes and returns the evolved user's secret key $sk_{id}^{\tau+1}$. The computation is essentially as same as mkUpd. We omit details in this abstract.

Enc$(param, id, m, \tau)$**:** To encrypt a plaintext $m \in \mathbb{G}_1$ using id at time-period $\tau \leq 2^t$, first parse τ as $\langle \tau_{t-1} | \cdots | \tau_0 \rangle$, and then pick a random value $s \in \mathbb{Z}_p$ and compute $C = (e(g_1, g_2)^s \cdot m, g^s, h_{mk}^s, h_{id}^s, (g_1^{\tau_{t-1}} \cdot f_1)^s, \cdots, (g_1^{\tau_0} \cdot f_1)^s,) \in \mathbb{G}_1 \times \mathbb{G}^{2+t}$.

Dec$(param, sk_{id}^\tau, C, \tau)$**:** To decrypt a ciphertext C for id at time τ, first parse C as $\langle A, B, D_1, D_2, E_1, \cdots, E_t \rangle$, and parse sk_{id} as $\langle a_0, a_1, a_2, b_1, \cdots, b_t, c_t, \cdots, c_1, d_1, \cdots, d_t \rangle$. Compute the plaintext as follows: $m \leftarrow A \cdot e(a_1, D_1) \cdot e(a_2, D_2) \cdot \prod_{i=1}^t e(d_i, E_i) / e(a_0, B)$.

3.3.2 Security Proof

Theorem 3.1

Our FSIBEm scheme is secure in the sense of FS-IND-ID-CPA if Waters is secure in the sense of IND-ID-CPA and BonehBoyen is secure in the sense of IND-sID-CPA.

Proof

We claim the security of our scheme can be proved straightforwardly from the security of Waters' scheme (Waters) [48] and Boneh-Boyen's scheme (BonehBoyen) [13].

Roughly speaking, the identity-axis and the time-axis of each hierarchy evolve independently in most of the operations. The two axes only meet together in the decryption algorithm.

3.4 Conclusions

This chapter proposed a forward secure identity based encryption scheme with master key update functionality, and this contrasts to previous schemes which only concerned the update of secret key of users but not the master key. Generic construction of such schemes from ordinary IBE could be considered as future work.

Chapter 4 MEANS OF SECURITY ENHANCEMENT

We do not trust the security that we cannot verify.
We cannot manage the security that we do not measure.

In this chapter, we represent the first formal analysis which proves FOPKC, FOCRYPTO and REACT generically enhance weak IBE schemes to IND-ID-CCA secure IBE scheme. It is not hard to straightforwardly apply these enhancements to IBE settings. What we have to do is only replacing the inputs of the original enhancements for PKE with the inputs of the applied enhancements for IBE. Simply speaking, we replace the public key with system parameters *param* and identity *id*.

Although the main ideas of proof in this chapter are as same as the ones in Chapter 2, for convenience, we employ another kind of proof technique which is called "event-based proof". This proof technique is widely used in numerous works in provable security research field.

After showing rigorous proof of these enhancements in IBE, we also represent an observation that the straightforward application of both FOPKC and FOCRYPTO to achieve a strong security is insufficient.

4.1 Introduction

In order to achieve the strongest IND-ID-CCA security, many researches of IBE schemes first build a "basic scheme" with only lower-level security, other than (IND-ID-CCA) security, then *specifically* apply certain security enhancement to upgrade the basic scheme to a new scheme with IND-ID-CCA security. However, these security enhancements are proposed in the PKE environment; e.g., FOPKC [24] is known to enhance IND-CPA secure scheme, FOCRYPTO [25] enhances OW-CPA, and REACT [38] enhances OW-PCA. But, it is still unknown whether these enhancements could *generically* upgrade weak security to IND-ID-CCA security in IBE environment.

Also, in IBE, the generality of security enhancements for PKE should be carefully checked. More exactly, we not only try to confirm the feasibility of such enhancements in the IBE setting, but also focus on gaining *tight* security reduction of such a proof, which is never a trivial job. It is worth reminding that a loose security reduction usually means lower security level with the same key size, and one has to adopt longer keys to compensate this security loss.

4.1.1 On Achieving IND-ID-CCA2 Security

FOcrypto is used to achieve IND-ID-CCA security in Boneh-Franklin's paper [16] for the first time. Galindo [27] has noticed a small flawed step in the proof of [16], however the security reduction in the corrected proof was even looser. In order to achieve a better security reduction, Galindo [27] employed FOpkc. We also note that, in fact, the proof given in [16, 27] did not take account of applying generic FOpkc or FOcrypto transforms, but has mainly considered how to reduce the security of the "full" scheme to that of an IND-CCA secure PKE.

Another variant of Boneh-Franklin scheme with tighter security reduction was given by Libert and Quisquater [37], with a REACT-like appearance by adopting the KEM-DEM idea. We note that this sense of "redundancy" is not the original sense of Phan and Pointchval, since optimistically a point on a curve for bilinear pairing has a length of 171 bits, which is slightly longer than 160 bits of necessary "redundancy". The more important thing is, again, there is no clear discussion on generic transforms for IBE in their paper, since this is not the theme of their work.

4.1.2 Contributions

In this chapter, our first contribution is that we prove these conversions (FOpkc, FOcrypto, REACT) can be applied to IBE *generically* with polynomial security reductions. But in IBE, the reductions of FOpkc and FOcrypto turn significantly worse than in PKE. Recall that in the conventional public key setting, FOpkc conversion can be proven with a "tight" security reduction to its underlying primitives.

Under this circumstance, we propose a slight modification of FOpkc and FOcrypto conversions. Thanks to this modification, we can partially overcome the problem, say, we can obtain better security reductions. The modification is very simple and computationally efficient: just hash the user's identity with other inputs to the random oracle. However, this simple idea actually works! Both the modified FOpkc and the modified FOcrypto admit exactly much tighter reductions as their public key counterparts. This is our second contribution.

On the other hand, the plain REACT already gives a good reduction cost, without any modification. Interestingly, these results may indicate a separation between the chosen plaintext attack (CPA) and plaintext checking attack (PCA) in the IBE setting.

Our third contribution is that in order to intuitively explain how our modification improves the security reduction, we further choose proper concrete parameters, and estimate the average running time of the simulator. For the chosen parameters, using a single PC (or a single dedicated hardware), an IND-ID-CCA adversary breaks the IND-ID-CCA security of "basic Boneh-Franklin scheme + plain FOpkc conversion" with about 10^{24} years in addition to break the IND-ID-CPA security of the basic Boneh-Franklin scheme. This is to say this additional time in plain FOpkc conversion is unacceptable in the realistic world. On the other hand, it needs only additional 10^8 or 10^9 years in the case of the modified FOpkc conver-

sion. Consider possible paralleled computing, say 1 million personal computers, this value decreases to $10^2 \sim 10^3$ years. Furthermore, after applying Moore's law, in 15 years, this value will decrease to 1.30 years, which is acceptable.

4.2 Investigation and Security Proof of Plain Enhancements

In this section, we investigate FOPKC, FOCRYPTO and REACT in IBE environment and show rigorous security proof of these enhancements.

4.2.1 IND-ID-CPA Enhancement

It is absorbing to find out an authentic way to enhance weak IBE schemes to strongly secure ones. In PKE, there exist such conversions, and FOPKC [24] is a good example. It is very efficient and achieves a tight security reduction. Since IBE is a different primitive from PKE, especially the algorithms are different, one may think these conversions are not immediately a solution for IBE. Although in specific case, e.g., [27], FOPKC was employed.

Plain FOpkc for IBE

Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IND-ID-CPA secure IBE scheme. Then, we can use FOPKC to construct an another IBE scheme $\Pi_1 = \{\mathcal{S}_1, \mathcal{X}_1, \mathcal{E}_1, \mathcal{D}_1\}$ as follows: Let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π_1 and $\mathsf{COIN}(k)$ be Π's coin-flipping space. The conversion is illustrated in Figure 4.1.

Proof of Security

Theorem 4.1

Suppose the hash function H is the random oracle, and Π is a γ-uniform IBE scheme. Let $\mathcal{B}$ be an IND-ID-CCA adversary who has advantage $\epsilon(k)$ against Π_1, and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_H H queries, q_E Extraction queries and q_D Decryption queries. Suppose executing $\mathcal{E}$ once needs at most time τ. Then there is an IND-ID-CPA adversary $\mathcal{A}$ who has advantage $\epsilon_1(k)$ against Π. Its running time is $t_1(k)$, where

$$\begin{aligned} \epsilon_1(k) &\geq \big(\epsilon(k) + \frac{1}{2} - \frac{q_H}{2^{l_1 - l_2}}\big)(1 - q_D \cdot \gamma) - \frac{1}{2} \\ t_1(k) &\geq t(k) + q_H \cdot q_D \cdot \tau \end{aligned}$$

Proof

We show how to construct adversary $\mathcal{A}$ by using adversary $\mathcal{B}$ as an oracle. The challenger starts an IND-ID-CPA game by executing $\mathcal{S}$ and generates $param$ and mk. The mk is kept secret by the challenger. $\mathcal{A}$ works by interacting with $\mathcal{B}$ in an IND-ID-CCA game as follows:

Setup: $\mathcal{A}$ gives $param$ to $\mathcal{B}$.

Setup $\mathcal{S}_1$:

It is as $\mathcal{S}$.

In addition, it picks a hash function $H : \{0,1\}^{l_2} \times \{0,1\}^{l_1-l_2} \to \mathsf{COIN}(k)$.

Extraction $\mathcal{X}_1$:

It is as $\mathcal{X}$.

Encryption $\mathcal{E}_1$:

Let $id \in \{0,1\}^*$ is the encryption key and $x \in \{0,1\}^{l_2}$ is the plaintext.

It computes $\mathcal{E}_1(param, id, x; \sigma) \leftarrow \mathcal{E}\big(param, id, x\|\sigma; H(x, \sigma)\big)$,

where σ is a randomly chosen $l_1 - l_2$ bit string.

Decryption $\mathcal{D}_1$:

Let y be a ciphertext to decrypt. Algorithm $\mathcal{D}_1$ works in the following steps:

1. Computes $x' \leftarrow \mathcal{D}(param, sk, y)$ and let $x \leftarrow [x']^{l_2}$ and $\sigma \leftarrow [x']_{l_1-l_2}$.
2. Tests that $\mathcal{E}\big(param, id, x||\sigma; H(x, \sigma)\big) = y$. If not, outputs "reject".
3. Outputs x as the decryption of y.

Figure 4.1: Algorithms of plain FOPKC

Phase 1: Three sorts of queries are answered as follows:

H**-query**(x_i, σ_i)**:** Let $\langle x_i, \sigma_i \rangle$ be an hash query issued by $\mathcal{B}$. $\mathcal{A}$ maintains a list of tuples $\langle x_i, \sigma_i, h_i \rangle$ as explained below. We refer to this list as the H^{list}. The list is initially empty. When $\mathcal{B}$ queries $H(x_i, \sigma_i)$, $\mathcal{A}$ responds as follows:

1. If the query x_i, σ_i already appears on the H^{list} in a tuple $\langle x_i, \sigma_i, h_i \rangle$ then $\mathcal{A}$ responds with h_i.
2. Otherwise, $\mathcal{A}$ picks a random element h_i from $\mathsf{COIN}(k)$ of Π.
3. $\mathcal{A}$ adds the tuple $\langle x_i, \sigma_i, h_i \rangle$ to the H^{list} and returns h_i.

$\mathcal{X}$**-query**(id_i)**:** Let $\langle id_i \rangle$ be an extraction query issued by $\mathcal{B}$. $\mathcal{A}$ inputs $\langle id_i \rangle$ to its own extraction oracle and gets the corresponding decryption key sk_i. $\mathcal{A}$ passes sk_i to $\mathcal{B}$ as the answer of the query.

$\mathcal{D}$**-query**(id_i, y_i)**:** Let $\langle id_i, y_i \rangle$ be a decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds as follows:

1. Find a pair of tuples $\langle x, \sigma, h \rangle$ from the H^{list}, such that $\mathcal{E}(param, id_i, x\|\sigma; h) = y_i$.
2. Outputs x if there exists such a pair of tuples, or outputs "reject" otherwise.

Challenge: Once $\mathcal{B}$ decides that Phase 1 is over it outputs a public key id^* ($id^* \neq id_i$) and two messages x_0, x_1 on which it wishes to be challenged. $\mathcal{A}$ randomly chooses two $l_1 - l_2$ bit strings σ_0 and σ_1. $\mathcal{A}$ sends $\langle \mathsf{ID}^*, x_0\|\sigma_0, x_1\|\sigma_1 \rangle$ to the challenger. The

Symbol	Event
SuccA	$\mathcal{A}$ wins the IND-ID-CPA game in the case that event Fail does not occur
SuccB	$\mathcal{B}$ wins the IND-ID-CCA game in the case that event Fail does not occur
Ask0	$\mathcal{B}$ queries $H(x_b, \sigma_b)$
Ask1	$\mathcal{B}$ queries $H(x_{\bar{b}}, \sigma_{\bar{b}})$
Fail	$\mathcal{A}$ fails to answer a decryption query at some point during the game

Table 4.1: Definitions of events in proof of plain FOPKC

challenger picks a random bit $b \in \{0, 1\}$ and sets $y^* \leftarrow \mathcal{E}(param, id^*, x_b||\sigma_b)$. Then $\mathcal{A}$ gives y^* as the challenge to $\mathcal{B}$.

Phase 2: Three sorts of queries are answered as the same as in Phase 1.

Guess: Once $\mathcal{B}$ decides that Phase 2 is over it outputs a guess b'.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ outputs this bit b' as the answer of the IND-ID-CPA game.

In order to calculate the security reduction, we first define five events in Table 4.1.

Then, we have,

$$\begin{aligned}\Pr[\mathsf{SuccB}] &= \Pr[\mathsf{SuccB}|\mathsf{Ask0}]\Pr[\mathsf{Ask0}] \\ &\quad + \Pr[\mathsf{SuccB}|\neg\mathsf{Ask0} \wedge \mathsf{Ask1}]\Pr[\neg\mathsf{Ask0} \wedge \mathsf{Ask1}] \\ &\quad + \Pr[\mathsf{SuccB}|\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}]\Pr[\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}]\end{aligned}$$

$$\begin{aligned}\Pr[\mathsf{SuccA}] &= \Pr[\mathsf{SuccA}|\mathsf{Ask0}]\Pr[\mathsf{Ask0}] \\ &\quad + \Pr[\mathsf{SuccA}|\neg\mathsf{Ask0} \wedge \mathsf{Ask1}]\Pr[\neg\mathsf{Ask0} \wedge \mathsf{Ask1}] \\ &\quad + \Pr[\mathsf{SuccA}|\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}]\Pr[\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}].\end{aligned}$$

From the specification of $\mathcal{A}$, the following equations hold:

$$\begin{aligned}\Pr[\mathsf{SuccA}|\mathsf{Ask0}] &= 1 \\ \Pr[\mathsf{SuccA}|\neg\mathsf{Ask0} \wedge \mathsf{Ask1}] &= 0 \\ \Pr[\mathsf{SuccB}|\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}] &= \Pr[\mathsf{SuccA}|\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}].\end{aligned}$$

Thus, we have,

$$\begin{aligned}\Pr[\mathsf{SuccA}] - \Pr[\mathsf{SuccB}] &= (1 - \Pr[\mathsf{SuccB}|\mathsf{Ask0}])\Pr[\mathsf{Ask0}] \\ &\quad - \Pr[\mathsf{SuccB}|\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}]\Pr[\neg\mathsf{Ask0} \wedge \neg\mathsf{Ask1}] \\ &\geq -\Pr[\neg\mathsf{Ask0} \wedge \mathsf{Ask1}].\end{aligned}$$

Since

$$\Pr[\neg\mathsf{Ask0} \wedge \mathsf{Ask1}] \leq \frac{q_H}{2^{l_1 - l_2}},$$

we have

$$\Pr[\mathsf{SuccA}] \geq \epsilon(k) + \frac{1}{2} - \frac{q_H}{2^{l_1 - l_2}}.$$

Next, we estimate $\Pr[\neg\mathsf{Fail}]$. The event Fail occurs only when $\mathcal{B}$ submits a Decryption query $\mathcal{D}$-query(id, y) such that $y = \mathcal{E}\big(param, id, x\|\sigma; H(x, \sigma)\big)$ without asking $H(x, \sigma)$. This case happens with probability at most γ , and therefore, we have that

$$\Pr[\neg\mathsf{Fail}] \leq (1-\gamma)^{q_D} \simeq 1 - q_D\gamma.$$

Hence, we have that

$$\epsilon_1(k) \quad \geq \quad \big(\epsilon(k) + \frac{1}{2} - \frac{q_H}{2^{l_1 - l_2}}\big)(1 - q_D \cdot \gamma) - \frac{1}{2}.$$

Finally, we estimate $\mathcal{A}$'s running time. Since in addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_H times for responding to each decryption query, $\mathcal{A}$'s running time is estimated as

$$t_1(k) \quad \geq \quad t(k) + q_H \cdot q_D \cdot \tau.$$

□

4.2.2 OW-ID-CPA Enhancement

As discussed in the above section, FOPKC is proved to be a powerful security enhancement for not only PKE schemes but also IBE schemes. In PKE, there exits another even stronger security enhancement, FOCRYPTO [25] which is also introduce by Fujisaki and Okamoto in 1999. It is known that as security goal, IND is stronger than OW. And FOCRYPTO is the way to enhance the OW-secure PKE schemes to IND-secure PKE schemes. As powerful as it is, FOCRYPTO has an inherent disadvantage, that is, the security reduction is not as ideal as FOPKC , even in PKE environment.

Before this work, it is an open problem whether FOCRYPTO could be generically applied to all OW-secure IBE schemes. Although in actual fact, Boneh and Franklin [16] employed FOCRYPTO in order to obtain their fully secure scheme.

Plain FOcrypto for IBE

Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an OW-ID-CPA secure IBE scheme. Then, we can use FOCRYPTOconstruct an another IBE scheme $\Pi_2 = \{\mathcal{S}_2, \mathcal{X}_2, \mathcal{E}_2, \mathcal{D}_2\}$ as follows: Let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π_2 and $\mathsf{COIN}(k)$ be Π's coin-flipping space. The conversion is illustrated in Figure 4.2.

Setup $\mathcal{S}_2$:

It is as $\mathcal{S}$.

In addition, we pick two hash functions,

$G : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \mathsf{COIN}(k)$ and $H : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.

Extraction $\mathcal{X}_2$:

It is as $\mathcal{X}$.

Encryption $\mathcal{E}_2$:

Let $id \in \{0,1\}^*$ is the encryption key and $x \in \{0,1\}^{l_2}$ is the plaintext.

It computes $\mathcal{E}_2(param, id, x; \sigma) \leftarrow \mathcal{E}\big(param, id, \sigma; G(\sigma, x)\big) \| H(\sigma) \oplus x$,

where σ is a randomly chosen l_2 bit string.

Decryption $\mathcal{D}_2$:

Let $y = y_1 \| y_2$ be a ciphertext to decrypt. Algorithm $\mathcal{D}_2$ works in the following steps:

1. Computes $\sigma \leftarrow \mathcal{D}(param, sk, y_1)$.
2. Computes $x \leftarrow H(\sigma) \oplus y_2$.
3. Sets $r \leftarrow G(\sigma, x)$. Tests that $\mathcal{E}(param, id, \sigma; r) = y_1$. If not, outputs "reject".
4. Outputs x as the decryption of y.

Figure 4.2: Algorithms of plain FOcrypto

Proof of Security

Theorem 4.2

Suppose the hash functions G and H are random oracles, and Π is a γ-uniform IBE scheme. Let $\mathcal{B}$ be an IND-ID-CCA adversary who has advantage $\epsilon(k)$ against Π_2, and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_H H queries, q_G G queries, q_E Extraction queries and q_D Decryption queries. Suppose executing $\mathcal{E}$ once needs at most τ. Then there is an OW-ID-CPA adversary $\mathcal{A}$ who has advantage $\epsilon_2(k)$ against Π. Its running time is $t_2(k)$, where

$$\begin{aligned} \epsilon_2(k) &\geq \frac{1}{q_H + q_G}\left(2\epsilon(k) - q_D\gamma - q_D/2^{l_2}\right) \\ t_2(k) &\geq t(k) + q_G \cdot q_D \cdot \tau \end{aligned}$$

Proof

We show how to construct adversary $\mathcal{A}$ by using adversary $\mathcal{B}$ as an oracle. The challenger starts an OW-ID-CPA game by executing $\mathcal{S}$ and generates $param$ and mk. The mk is kept secret by the challenger. $\mathcal{A}$ works by interacting with $\mathcal{B}$ in an IND-ID-CCA game as follows:

Setup: $\mathcal{A}$ gives $param$ to $\mathcal{B}$.

Phase 1: Four sorts of queries are answered as follows:

G**-query**(σ_i, x_i)**:** $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, x_i, g_i \rangle$ as explained below. We refer to this list as the G^{list}. The list is initially empty. When $\mathcal{B}$ queries $G(\sigma_i, x_i)$, $\mathcal{A}$ responds as follows:

1. If the query σ_i and x_i already appears on the G^{list} in a tuple $\langle \sigma_i, M_i, g_i \rangle$ then $\mathcal{A}$ responds with g_i.
2. Otherwise, $\mathcal{A}$ picks a random element g_i from COIN(k) of Π.
3. $\mathcal{A}$ adds the tuple $\langle \sigma_i, x_i, g_i \rangle$ to the G^{list} and returns g_i.

H**-query**(σ_i)**:** $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, h_i \rangle$ to respond the queries. We refer to this list as H^{list}. The list is initially empty. When $\mathcal{B}$ queries $H(\sigma_i)$, $\mathcal{A}$ responds as following:

1. If the query σ_i already appears on the H^{list} in a tuple $\langle \sigma_i, h_i \rangle$ then $\mathcal{A}$ responds with h_i.
2. Otherwise, $\mathcal{A}$ picks a string h_i from $\{0,1\}^{l_2}$ randomly.
3. $\mathcal{A}$ adds the tuple $\langle \sigma_i, h_i \rangle$ to the H^{list} and returns h_i.

$\mathcal{X}$**-query**(id_i)**:** Let $\langle id_i \rangle$ be an Extraction query issued by $\mathcal{B}$. $\mathcal{A}$ inputs $\langle id_i \rangle$ to its own extraction oracle and gets the corresponding decryption key sk_i. $\mathcal{A}$ passes sk_i to $\mathcal{B}$ as the answer of the query.

$\mathcal{D}$**-query**(id_i, y_i)**:** Let $\langle id_i, y_i \rangle$ be a Decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds as follows:

Symbol	Event
SuccB	$\mathcal{B}$ wins the IND-ID-CCA game in the case that event Fail does not occur
AskB	$\mathcal{B}$ queries $G(\mathcal{D}(param, sk, y_{ch1}), *)$ or $H(\mathcal{D}(param, sk, y_{ch1}))$, where $sk \leftarrow \mathcal{X}(param, mk, id^*)$ and $*$ denotes any l_2-bit string
Fail	$\mathcal{A}$ fails before $\mathcal{B}$ submits a query for $G(\mathcal{D}(param, sk, y_{ch1}), *)$ or $H(\mathcal{D}(param, sk, y_{ch1}))$

Table 4.2: Definitions of events in proof of plain FOcrypto

1. Find a pair of tuples $\langle \sigma, x, g \rangle$ and $\langle \sigma, h \rangle$ from the G^{list} and H^{list}, respectively, such that $\mathcal{E}(param, id_i, \sigma; g) \| h \oplus x = y_i$.
2. Outputs x if there exists such a pair of tuples, or outputs "reject" otherwise.

Challenge: Once $\mathcal{B}$ decides that Phase 1 is over it outputs a public key id^* ($id^* \neq id_i$) and two messages x_0, x_1 on which it wishes to be challenged. $\mathcal{A}$ sends id^* to the challenger and receives a ciphertext y^*. Then, $\mathcal{A}$ generates $y_{ch1} \| y_{ch2}$ where $y_{ch1} = y^*$ and y_{ch2} is a random string whose length is l_2. $\mathcal{A}$ gives $y_{ch1} \| y_{ch2}$ as the challenge to $\mathcal{B}$.

Phase 2: Four sorts of queries are answered as the same as in Phase 1.

Guess: Once $\mathcal{B}$ decides that Phase 2 is over it outputs a guess b'.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ chooses a tuple $\langle \sigma, x, g \rangle$ from the G^{list}, or chooses a tuple $\langle \sigma, h \rangle$ from the H^{list}. Then, $\mathcal{A}$ outputs σ in the tuple as the answer of the OW-ID-CPA game.

In order to calculate the reduction cost, we first define five events in Table 4.2.

Then, we have that

$$\Pr[\mathsf{SuccB} | \neg \mathsf{Fail}] \cdot \Pr[\neg \mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}].$$

Since

$$\Pr[\mathsf{SuccB} | \neg \mathsf{Fail}, \neg \mathsf{AskB}] = \frac{1}{2},$$

we also have

$$\begin{aligned} \Pr[\mathsf{SuccB} | \neg \mathsf{Fail}] &= \Pr[\mathsf{SuccB} | \neg \mathsf{Fail} \wedge \mathsf{AskB}] \cdot \Pr[\mathsf{AskB}] + \frac{1}{2}\Big(1 - \Pr[\mathsf{AskB}]\Big) \\ &\leq \frac{1}{2} \Pr[\mathsf{AskB}] + \frac{1}{2}. \end{aligned}$$

Hence, we have that

$$\Big(\frac{1}{2} \Pr[\mathsf{AskB}] + \frac{1}{2}\Big) \cdot \Pr[\neg \mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}],$$

and therefore,

$$\Pr[\mathsf{AskB}] \geq 2\epsilon(k) - \Pr[\mathsf{Fail}].$$

Next, we estimate $\Pr[\mathsf{Fail}]$. The event Fail occurs only in either

Case 1. $\mathcal{B}$ submits a decryption query $\mathcal{D}$-query$(id, y_1 \| H(\sigma) \oplus x)$ such that $y_1 = \mathcal{E}(param, id, \sigma; G(\sigma, x))$ without asking $G(\sigma, M)$, or

Case 2. $\mathcal{B}$ submits a decryption query $\mathcal{D}$-query$(id, \mathcal{E}(param, id, \sigma; G(\sigma, x)) \| y_2)$ such that $y_2 = H(\sigma) \oplus x$ without asking $H(\sigma)$.

Case 1 and **2** happen with probability at most γ and $1/2^{l_2}$, respectively. Therefore, we have that

$$\Pr[\mathsf{Fail}] \leq 1 - (1 - \gamma - \frac{1}{2^{l_2}})^{q_D}.$$

Hence, we have that

$$\begin{aligned} \epsilon_2(k) &\geq \frac{1}{q_G + q_H} \Pr[\mathsf{AskB}] \\ &\geq \frac{1}{q_G + q_H} \Big(2\epsilon(k) - \big(1 - (1 - \gamma - \frac{1}{2^{l_2}})^{q_D}\big) \Big) \\ &\simeq \frac{1}{q_G + q_H} \big(2\epsilon(k) - q_D\gamma - \frac{q_D}{2^{l_2}} \big). \end{aligned}$$

Finally, we estimate $\mathcal{A}$'s running time. Since in addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_G times for responding to each Decryption query, $\mathcal{A}$'s running time is estimated as

$$t_2(k) \geq t(k) + q_G \cdot q_D \cdot \tau.$$

□

4.2.3 OW-ID-PCA Enhancement

As a non-common security notion, OW-ID-PCA denotes *one-wayness against adaptive chosen identity and plaintext checking attack*. The difference from OW-ID-CPA is that, in this scenario, instead of the $\mathcal{E}$-queries, the adversary is granted the ability of issuing another kind of queries, the $\mathcal{PC}$-queries.

$\mathcal{PC}$-query(id, x, y) Plaintext-Checking queries. Let $\langle id, x, y \rangle$ be a plaintext-checking query issued by the adversary. Then the adversary should be responded with "yes" if y is ciphertext of x under public key id; or else be responded with "no".

Note this kind of queries is not that common as the other three kinds introduced in Chapter 2, but it is still useful and meaningful. Because in some situation, This kind of queries is equal to $\mathcal{E}$-queries, e.g., for deterministic encryptions.

Similar as OW-ID-CPA, the security notion OW-ID-PCA is also a sort of weak security notion. In order to enhance PKE schemes that only possess this security, in 2001 Okamoto and Pointcheval proposed "Rapid Enhanced-security Asymmetric Cryptosystem Transform", a.k.a., REACT [38]. It is very fast while achieves a tight reduction cost. Although REACT

Setup $\mathcal{S}_3$:

It is as $\mathcal{S}$.

In addition it picks two hash functions:

$G : \mathcal{M} \to \{0,1\}^{l_2}, H : \mathcal{M} \times \mathcal{M}' \times \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{l_3}$.

Extraction $\mathcal{X}_3$:

It is as $\mathcal{X}$.

Encryption $\mathcal{E}_3$:

For any massage $x \in \mathcal{M}'$ and random values $\sigma \in \mathcal{M}$, it gets

$y_1 \leftarrow \mathcal{E}(param, id, \sigma; r)$, then it computes

$k \leftarrow G(\sigma), y_2 \leftarrow k \oplus x, y_3 \leftarrow H(\sigma, x, y_1, y_2)$.

The ciphertext consists of the triple $y = (y_1, y_2, y_3)$.

Decryption $\mathcal{D}_3$:

Let $y = (y_1, y_2, y_3)$ be a ciphertext to decrypt.

Algorithm $\mathcal{D}_3$ works in the following steps:

1. Decrypts y_1 and gets σ.
2. Computes $k \leftarrow G(\sigma)$ and $x \leftarrow y_2 \oplus k$.
3. Tests that $y_3 = H(\sigma, x, y_1, y_2)$. If not, outputs "reject".
4. Returns x as the decryption of y.

Figure 4.3: Algorithms of plain REACT

was specifically employed [37], again, it is not known whether REACT can be applied to IBE generically before this work. We investigate the fact in this section. Interestingly plain REACT is not only effective for IBE, but also gives a tight reduction cost, as it does for PKE.

Plain REACT for IBE

Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an OW-ID-PCA secure IBE. Let $\mathcal{M}$ be a message space of Π and $\mathcal{C}$ be a ciphertext space of Π. Then we can use REACT to construct an another IBE scheme $\Pi_3 = \{\mathcal{S}_3, \mathcal{X}_3, \mathcal{E}_3, \mathcal{D}_3\}$ which is secure against IND-ID-CCA. Let $\mathcal{M}'$ be a message space of Π_3 and $\mathcal{C}'$ be a ciphertext space of Π_3. A ciphertext C of Π_3 consists three components y_1, y_2 and y_3. We denote the bit length of these components l_1, l_2 and l_3 respectively. The definition of Π_3 is as follows in Figure 4.3.

Proof of Security

Theorem 4.3

Suppose the hash functions G and H are random oracles. Let $\mathcal{B}$ be an IND-ID-CCA adversary who has advantage $\epsilon(k)$ against Π_3, and its running time is at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_G G-queries, q_H H-queries, q_E Extraction queries and q_D Decryption queries. Then there is an OW-ID-PCA adversary $\mathcal{A}$ who has advantage $\epsilon_3(k)$ against Π. Its running time is $t_3(k)$, where

$$\begin{aligned} \epsilon_3(k) &\geq 2\epsilon(k) - q_D(\frac{1}{2^{l_2}} + \frac{1}{2^{l_3}}) \\ t_3(k) &\geq t(k) + (q_G + q_H) \cdot O(1) \end{aligned}$$

Proof

We show how to construct adversary $\mathcal{A}$ by using adversary $\mathcal{B}$ as an oracle. The challenger starts an OW-ID-PCA game by executing $\mathcal{S}$ and generates $param$ and mk. The mk is kept secret by the challenger. $\mathcal{A}$ works by interacting with $\mathcal{B}$ in an IND-ID-CCA game as follows:

Setup: $\mathcal{A}$ gives $param$ to $\mathcal{B}$.

Phase 1: Four sorts of queries are answered as follows:

G**-query**(σ_i)**:** $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, k_i \rangle$ as explained below. We refer to this list as G^{list}. The list is initially empty. When $\mathcal{B}$ asks $G(\sigma_i)$, $\mathcal{A}$ responds as follows:

1. If query σ_i already appears on the G^{list} in a tuple $\langle \sigma_i, k_i \rangle$ then $\mathcal{A}$ responds with k_i.
2. Otherwise, A picks a random element k_i from $\{0,1\}^{l_2}$.
3. $\mathcal{A}$ adds the tuple $\langle \sigma_i, k_i \rangle$ to the G^{list} and returns k_i.

H**-query**$(\sigma_i, x_i, y_1, y_2)$**:** $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, x_i, y_1, y_2, y_3 \rangle$. We refer this list as the H^{list}. The list is initially empty. When $\mathcal{B}$ queries $H(\sigma, x, y_1, y_2)$, $\mathcal{A}$ responds as follows:

1. If the query $\langle \sigma, x, y_1, y_2 \rangle$ is already appears on the H^{list} in the tuple $\langle \sigma, x, y_1, y_2, y_3 \rangle$ then $\mathcal{A}$ responds with y_3.
2. Otherwise, $\mathcal{A}$ randomly picks a string y_3 from $\{0,1\}^{l_3}$.
3. $\mathcal{A}$ adds the tuple $\langle \sigma, x, y_1, y_2, y_3 \rangle$ to the H^{list} and returns y_3.

$\mathcal{X}$**-query**(id_i)**:** Let $\langle id_i \rangle$ be an Extraction query issued by $\mathcal{B}$. $\mathcal{A}$ inputs $\langle id_i \rangle$ to its own extraction oracle and receives the corresponding decryption key sk_i. $\mathcal{A}$ sends the key sk_i to $\mathcal{B}$ as the answer of the query.

$\mathcal{D}$**-query**(id_i, y_1, y_2, y_3)**:** Let $\langle id_i, y_1, y_2, y_3 \rangle$ be a Decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds as follows:

1. $\mathcal{A}$ picks up a tuple $\langle \sigma', x', y_1', y_2', y_3' \rangle$ from H^{list} such that $y_3 = y_3'$.

2. $\mathcal{A}$ computes $k' \leftarrow G(\sigma')$.
3. Checks if $y_2 = x' \oplus k'$. If this holds, $\mathcal{A}$ issues $\mathcal{PC}$-query(id_i, σ', y_1) to the PC oracle.
4. If the PC oracle answers "yes", $\mathcal{A}$ returns x' to $\mathcal{B}$. Otherwise, $\mathcal{A}$ outputs "reject".

Challenge: Once $\mathcal{B}$ decides that Phase 1 is over it outputs a public key id^* ($id^* \neq id_i$) and two message x_0, x_1 on which it wishes to be challenged. $\mathcal{A}$ sends id^* to the challenger and receives a ciphertext y^*. $\mathcal{A}$ generates a l_2 bit random string y_2 and a l_3 bit random string y_3. $\mathcal{A}$ gives $\langle y^*, y_2, y_3 \rangle$ to $\mathcal{B}$ as a challenge ciphertext.

Phase 2: Four sorts of queries are answered as the same as in Phase 1.

Guess: Once $\mathcal{B}$ decides that Phase 2 is over it outputs a guess b'.

After $\mathcal{B}$ outputs a guess b', $\mathcal{A}$ picks all σs which appear in tuples on the G^{list} and the H^{list}. For each σ, $\mathcal{A}$ queries $\langle id^*, \sigma, y^* \rangle$ to PC oracle. If PC oracle returns "yes", $\mathcal{A}$ outputs the σ as the answer of OW-ID-PCA game.

To estimate the advantage of $\mathcal{A}$, we define the following four events in Table 4.3.

Symbol	Event
SuccA	$\mathcal{A}$ wins the OW-ID-PCA game
SuccB	$\mathcal{B}$ wins the IND-ID-CCA game
AskB	$\mathcal{B}$ asks a query for $G(\sigma^*)$ or $H(\sigma^*, x_b, y_1, y_2)$ at some point during the game
Fail	the simulation fails before the event AskB occurs

Table 4.3: Definitions of events in proof of plain REACT

Then we take the same discussion as in the proof of Theorem 1 and we have that

$$\Pr[\mathsf{SuccB}|\neg\mathsf{Fail}]\Pr[\neg\mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}].$$

Since

$$\Pr[\mathsf{SuccB}|\neg\mathsf{Fail} \wedge \neg\mathsf{AskB}] = \frac{1}{2},$$

we also have

$$\begin{aligned} \Pr[\mathsf{SuccB}|\neg\mathsf{Fail}] &= \Pr[\mathsf{SuccB}|\neg\mathsf{Fail} \wedge \mathsf{AskB}]\Pr[\mathsf{AskB}] + \frac{1}{2}(1 - \Pr[\mathsf{AskB}]) \\ &\leq \frac{1}{2}\Pr[\mathsf{AskB}] + \frac{1}{2}. \end{aligned}$$

Hence, we have that

$$(\frac{1}{2}\Pr[\mathsf{AskB}] + \frac{1}{2})\Pr[\neg\mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}]$$

and therefore,

$$\Pr[\mathsf{AskB}] \geq 2\epsilon(k) - \Pr[\mathsf{Fail}].$$

Next, we estimate $\Pr[\mathsf{Fail}]$. The event Fail occurs only in either

Case 1. $\mathcal{B}$ submits a decryption query $\mathcal{D}$-query$(id, y_1, G(\sigma) \oplus x, y_3)$ such that $y_1 = \mathcal{E}(param, id, \sigma; r)$ and $y_3 = H(\sigma, x, y_1, G(\sigma) \oplus x)$ without asking $G(\sigma)$, or

Case 2. $\mathcal{B}$ submits a decryption query $\mathcal{D}$-query$(id, y_1, y_2, H(\sigma, x, y_1, y_2))$ without asking $H(\sigma, x, y_1, y_2)$.

Case 1 and **2** happen with probability at most 2^{-l_2} and 2^{-l_3}, respectively, and therefore, we have that

$$\Pr[\mathsf{Fail}] \leq 1 - (1 - \frac{1}{2^{l_2}} - \frac{1}{2^{l_3}})^{q_D} \simeq q_D(\frac{1}{2^{l_2}} + \frac{1}{2^{l_3}}).$$

If $\mathcal{B}$ wins the IND-ID-CCA game, then $\mathcal{A}$ also win the OW-ID-PCA game. Therefore,

$$\Pr[\mathsf{SuccA}] \geq \Pr[\mathsf{SuccB}].$$

Hence, we have that

$$\epsilon_3(k) \quad \geq \quad \Pr[\mathsf{SuccA}] \geq \Pr[\mathsf{SuccB}] \simeq 2\epsilon(k) - q_D(\frac{1}{2^{l_2}} + \frac{1}{2^{l_3}}).$$

Finally, we estimate $\mathcal{A}$'s running time. Since in addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to answer the G and H queries. Thus $\mathcal{A}$'s running time is estimated as

$$t_3(k) \geq t(k) + (q_H + q_G) \cdot O(1).$$

□

4.2.4 Discussion

As shown in Theorem 4.1, there exists a polynomial time reduction from $\mathcal{B}$ to $\mathcal{A}$, and consequently, any polynomial time adversary cannot break Π_1 in IND-ID-CCA sense if any polynomial time adversary cannot break Π in IND-ID-CPA sense. However, this result does not immediately imply that any realistic adversary cannot break Π_1 in IND-ID-CCA sense if any realistic adversary cannot break Π in IND-ID-CPA sense. Suppose that $\mathcal{A}$'s computational time is significantly larger than $\mathcal{B}$'s. Then, it might be still infeasible to break Π in practice even if $\mathcal{B}$ can break Π_1 in IND-ID-CCA sense. Bellare and Rogaway [7] proposed the notion of *exact security* for formally dealing with this issue.

Now we focus on the running time of $\mathcal{A}$ and $\mathcal{B}$ in the proof of plain FOPKC for IBE. As shown, there exists a polynomial time reduction from $\mathcal{B}$ to $\mathcal{A}$: in the reduction given above $\mathcal{A}$'s running time is estimated as $t_1(k) = t(k) + q_H \cdot q_D \cdot \tau$, where $t(k)$ is $\mathcal{B}$'s running time.

Assuming that q_H and q_D are estimated as 2^{60} and 2^{40} respectively, $\mathcal{A}$ has to run the encryption algorithm $\mathcal{E}$ for 2^{100} times, which are computationally *infeasible* in practice.

(Notice that a Decryption query requires on-line computation, while a H-query only requires off-line hash computation.) Therefore, $\mathcal{A}$ cannot break IND-ID-CPA security of Π in practice (even if $\mathcal{B}$ works in practical time).

Conclusively, the above straightforward application of FOPKC is insufficient for achieving a strong security. Also, according to Theorem 4.2, we observe that the same problem happens in the straightforward application of FOCRYPTO. On the other hand, as shown in Theorem 4.3, the time reduction of plain REACT is quite satisfiable.

In this section, in order to solve the problem of time reduction efficiency, we propose modified FOPKC and modified FOCRYPTO conversion for IBE schemes. These modifications are with improved time reduction cost, i.e., the simulators need much shorter additional running time but still obtain the same advantage as the simulator does in straightforward applications.

On the other hand, unlike plain FOPKC and plain FOCRYPTO, plain REACT already possesses a tight security reduction for IBE schemes. We remark that this is mainly caused by the PC oracle, which implicitly handles the id by its definition. The significant differences of reduction costs may indicate a separation between these two attack models: CPA and PCA.

4.3 Towards More Efficient Enhancements

4.3.1 More Efficient IND-ID-CPA Enhancement

Observation and basic idea.

The huge running time of $\mathcal{A}$ in Theorem 4.1 is caused by the following reason. In order to respond to a decryption query $\mathcal{D}$-query(id, y), $\mathcal{A}$ has to find a tuple from H^{list} such that its corresponding ciphertext under public key id is identical to y. Because $\mathcal{A}$ does not know id in advance, it is required to carry out re-encryption with public key id for all tuples in H^{list} for every $\mathcal{D}$-query. This results in $q_H \cdot q_D$ times of re-encryption operations. To solve this problem, we add id as one of the inputs to H. We remark that this modification is quite simple in shape, but it is the right thing that we need.

Construction and security proof.

Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IND-ID-CPA secure IBE scheme. Then, we can construct an another IBE scheme $\Pi_4 = \{\mathcal{S}_4, \mathcal{X}_4, \mathcal{E}_4, \mathcal{D}_4\}$ as follows: let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π_4 and COIN(k) be Π's coin-flipping space. The conversion is illustrated in Table 4.4.

Theorem 4.4

Suppose the hash function H is the random oracle, and Π is γ-uniform IBE encryption scheme. Let $\mathcal{B}$ be an IND-ID-CCA adversary who has advantage $\epsilon(k)$ against Π_4, and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_H H-queries, q_E Extraction queries and q_D

Setup $\mathcal{S}_4$:

It is as $\mathcal{S}$.

In addition, it picks a hash function $H : \{0,1\}^{l_2} \times \{0,1\}^{l_1-l_2} \times \{0,1\}^* \to \mathsf{COIN}(k)$.

Extraction $\mathcal{X}_4$:

It is as $\mathcal{X}$.

Encryption $\mathcal{E}_4$:

Let $id \in \{0,1\}^*$ is the encryption key and $x \in \{0,1\}^{l_2}$ is the plaintext.

It computes $\mathcal{E}_2(param, id, x; \sigma) = \mathcal{E}\big(param, id, x\|\sigma; H(x, \sigma, id)\big)$,

where σ is a randomly chosen $l_1 - l_2$ bit string.

Decryption $\mathcal{D}_4$:

Let y be a ciphertext to decrypt. Algorithm $\mathcal{D}_4$ works in the following steps:

1. Computes $x' \leftarrow \mathcal{D}(param, sk, y)$ and let $x \leftarrow [x']^{l_2}$, $\sigma \leftarrow [x']_{l_1-l_2}$.
2. Tests that $\mathcal{E}\big(param, id, x'; H(x, \sigma, id)\big) = y$. If not, outputs "reject".
3. Outputs x as the decryption of y.

Figure 4.4: Algorithms of modified FOPKC

Decryption queries. Suppose executing $\mathcal{E}$ once needs at most τ. Then there is an IND-ID-CPA adversary $\mathcal{A}$ who has advantage $\epsilon_4(k)$ against Π. Its running time is $t_4(k)$, where

$$\begin{aligned} \epsilon_4(k) &\geq \big(\epsilon(k) + \frac{1}{2} - \frac{q_H}{2^{l_1-l_2}}\big)(1 - q_D \cdot \gamma) - \frac{1}{2} \\ t_4(k) &\geq t(k) + q_H \cdot \tau \end{aligned}$$

Proof

To prove this theorem, almost the same strategy as the proof of Theorem 4.1 can be used. That is, assuming IND-ID-CCA adversary $\mathcal{B}$ for Π_4, constructing IND-ID-CPA adversary $\mathcal{A}$ for Π which uses $\mathcal{B}$ as an oracle.

There are two different points between the proof of Theorem 4.1 and Theorem 4.4. The points are how to answer H-queries and $\mathcal{D}$-queries in the IND-ID-CCA game between $\mathcal{A}$ and $\mathcal{B}$. For easiness of comparison, we describe only these different points.

H-query(x_i, σ_i)**:** $\mathcal{A}$ maintains a list of tuples $\langle x_i, \sigma_i, id_i, h_i, y_i\rangle$ as explained below. We refer to this list as the H^{list}. The list is initially empty. When $\mathcal{B}$ queries $H(x_i, \sigma_i, id_i)$, $\mathcal{A}$ responds as follows:

1. If the query x_i, σ_i and id_i already appears on the H^{list} in a tuple $\langle x_i, \sigma_i, id_i, h_i, y_i\rangle$ then $\mathcal{A}$ responds with $H(x_i, \sigma_i, id_i) = h_i$.
2. Otherwise, $\mathcal{A}$ picks a random element h_i from $\mathsf{COIN}(k)$.

3. $\mathcal{A}$ generates a ciphertext $y_i = \mathcal{E}(param, id_i, x_i \| \sigma_i; h_i)$.
4. $\mathcal{A}$ adds the tuple $\langle x_i, \sigma_i, id_i, h_i, y_i \rangle$ to the H^{list} and responds to $\mathcal{B}$ with $H(x_i, \sigma_i, id_i) = h_i$.

$\mathcal{D}$-query(id_i, y_i)**:** Let $\langle id_i, y_i \rangle$ be a decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds this query in the following steps:

1. Finds a tuple $\langle \sigma_j, x_j, id_j, g_j, y_j \rangle$ from the H^{list} such that $id_i = id_j$ and $y_i = y_j$.
2. Outputs x_j if there exists such a tuple, or outputs "reject" otherwise.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ outputs this bit b' as the answer of the IND-ID-CPA game.

The advantage of $\mathcal{A}$ can be evaluate in the same way as in Theorem 4.1, so we omit details here. Since in additional to $\mathcal{B}$' running time, $\mathcal{A}$ has to run $\mathcal{E}$ once when a new H-query is asked. Therefore, $\mathcal{A}$'s running time is estimated as $t(k) + q_H \cdot \tau$.

□

4.3.2 More Efficient OW-ID-CPA Enhancement

Observation and basic idea.

The reason that plain FOCRYPTO leads an inefficient time reduction is as same as the case of plain FOPKC. The huge running time of $\mathcal{A}$ in Theorem 4.2 is caused by the following reason. In order to respond to a decryption query $\mathcal{D}$-query(id, y), $\mathcal{A}$ has to find a pair of tuples from G^{list} and H^{list} such that its corresponding ciphertext under public key id is identical to y. Because $\mathcal{A}$ does not know id in advance, it is required to carry out re-encryption with public key id for all tuples in G^{list} for every $\mathcal{D}$-query. This results in $q_G \cdot q_D$ times of re-encryption operations. To solve this problem, we add id as one of the inputs to G.

Construction and security proof.

Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an OW-ID-CPA secure IBE scheme. Then, we can construct an another IBE scheme $\Pi_5 = \{\mathcal{S}_5, \mathcal{X}_5, \mathcal{E}_5, \mathcal{D}_5\}$ as follows: let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π_5 and COIN(k) be Π's coin-flipping space. The construction is illustrated in Figure 4.5.

Theorem 4.5

Suppose the hash functions G and H are random oracles and Π is a γ-uniform IBE encryption scheme. Let $\mathcal{B}$ be an IND-ID-CCA adversary who has advantage $\epsilon(k)$ against Π_5 and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_H H queries, q_G G queries, q_E Extraction queries and q_D Decryption queries. Suppose executing $\mathcal{E}$ once needs at most τ. Then there

Setup $\mathcal{S}_5$:

It is as $\mathcal{S}$.

In addition, we pick two hash functions,

$G : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \times \{0,1\}^* \to \mathsf{COIN}(k)$ and $H : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.

Extraction $\mathcal{X}_5$:

It is as $\mathcal{X}$.

Encryption $\mathcal{E}_5$:

Let $id \in \{0,1\}^*$ is the encryption key and $x \in \{0,1\}^{l_2}$ is the plaintext.

It computes $\mathcal{E}_2(param, id, x; \sigma) \leftarrow \mathcal{E}\big(param, id, \sigma; G(\sigma, x, id)\big) \| H(\sigma) \oplus x$,

where σ is a randomly chosen l_2 bit string.

Decryption $\mathcal{D}_5$:

Let $y = y_1 \| y_2$ be a ciphertext to decrypt.

Algorithm $\mathcal{D}_2$ works in the following steps:

1. Computes $\sigma \leftarrow \mathcal{D}(param, sk, y_1)$.
2. Computes $x \leftarrow H(\sigma) \oplus y_2$.
3. Sets $r \leftarrow G(\sigma, x, id)$. Tests that $\mathcal{E}(param, id, \sigma; r) = y_1$.

   If not, outputs "reject".
4. Outputs x as the decryption of y.

Figure 4.5: Algorithms of modified FOcrypto

is an OW-ID-CPA adversary $\mathcal{A}$ who has advantage $\epsilon_5(k)$ against Π. Its running time is $t_5(k)$, where

$$\begin{aligned} \epsilon_5(k) &\geq \frac{1}{q_H + q_G}\left(2\epsilon(k) - q_D\gamma - q_D/2^{l_2}\right) \\ t_5(k) &\geq t(k) + q_G \cdot \tau \end{aligned}$$

Proof
To prove This theorem, almost same strategy as the proof of Theorem 4.2 can be used. That is, assuming IND-ID-CCA adversary $\mathcal{B}$ for Π_5, constructing OW-ID-CPA adversary $\mathcal{A}$ for Π which uses $\mathcal{B}$ as an oracle.

There are two different points between the proof of Theorem 4.2 and Theorem 4.5. The points are how to answer G-queries and decryption-queries in the IND-ID-CCA game between $\mathcal{A}$ and $\mathcal{B}$. For easiness of comparison, we describe only these different points.

G-query(σ_i, x_i, id_i)**:** $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, x_i, id_i, g_i, y_i \rangle$ as explained below. We refer to this list as the G^{list}. The list is initially empty. When $\mathcal{B}$ queries $G(\sigma_i, x_i, id_i)$, $\mathcal{A}$ responds as follows:

1. If the query σ_i, x_i and id_i already appears on the G^{list} in a tuple $\langle \sigma_i, x_i, id_i, g_i, y_i \rangle$ then $\mathcal{A}$ responds with $G(\sigma_i, x_i, id_i) = g_i$.
2. Otherwise, $\mathcal{A}$ picks a random element g_i from $\mathsf{COIN}(k)$.
3. $\mathcal{A}$ generates a ciphertext $y_i \leftarrow \mathcal{E}(param, id_i, \sigma_i; g_i) \| H(\sigma_i) \oplus x_i$.
4. $\mathcal{A}$ adds the tuple $\langle \sigma_i, x_i, id_i, g_i, y_i \rangle$ to the G^{list} and responds to $\mathcal{B}$ with $G(\sigma_i, x_i, id_i) = g_i$.

$\mathcal{D}$-query(id_i, y_i)**:** $\mathcal{A}$ responds this query in the following steps:

1. Finds a tuple $\langle \sigma_j, x_j, id_j, g_j, y_j \rangle$ from the G^{list} such that $id_i = id_j$ and $y_i = y_j$.
2. Outputs x_j if there exists such a tuple, or outputs "reject" otherwise.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ chooses a tuple $\langle \sigma, x, id, g, y \rangle$ or $\langle \sigma, h \rangle$ from the G^{list} or the H^{list} randomly and outputs σ in the tuple as the answer of the OW-ID-CPA game.

The advantage of $\mathcal{A}$ can be evaluate in the same way as in Theorem 4.2. So, we omit the details here. Since in addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_G times to make the G^{list}. Hence, the running time of $\mathcal{A}$ is estimated as $t(k) + q_G \cdot \tau$. □

4.3.3 Discussion

We simply compare the Running time of $\mathcal{A}$ in plain applications and modified applications. We especially focus on times to run the encryption algorithm $\mathcal{E}$ which is required for each

simulation. It is believed that if a simulator has to run $\mathcal{E}$ for more than 2^{80} times, then it does not properly work in a realistic time. Now, we have that

$$\#\mathcal{E}(\Pi_1)(\sim 2^{100}) \gg 2^{80} \gg \#\mathcal{E}(\Pi_4)(\sim 2^{60})$$
$$\#\mathcal{E}(\Pi_2)(\sim 2^{100}) \gg 2^{80} \gg \#\mathcal{E}(\Pi_5)(\sim 2^{60})$$

where $\#\mathcal{E}(\cdot)$ denotes the times to run $\mathcal{E}$ in the simulation. This implies that the running time of the simulator for Π_4 and Π_5 is considered realistic. This is to say, our modifications for FOPKC and FOCRYPTO works.

4.4 Numerical Explanation by Encryption Time

In this section, we compare the cost of security reductions in the plain FOPKC and the modified FOPKC by numerical evaluation, and discuss what this result means in the realistic world.

As shown in Theorem 4.1, the plain FOPKC gives a polynomial time reduction, and looking at the coefficient of $\epsilon(k)$, the reduction seems tight This means that the adversary's advantage against the underlying weak scheme and that against the enhanced strong scheme are close. Therefore, at the first glance, merit of the our modified FOPKC seems not considerable.

However, when we focus on the fact that the other terms except for ϵ term have a significant influence on the reduction cost in plain FOPKC, the problem comes out as described at the end of Section 4.2.1. So, here, we compare the plain FOPKC and the modified FOPKC by strictly estimating

$$\begin{cases} T_P \leftarrow t_1(k)/\epsilon_1(k), \text{ in plain FOPKC} \\ T_M \leftarrow t_4(k)/\epsilon_4(k), \text{ in modified FOPKC} \end{cases}$$

where T_P (or T_M) is intuitively the average expected computational time for adversary to succeed in breaking the basic IND-ID-CPA secure IBE scheme Π, in the proof of plain FOPKC (or modified FOPKC). In order to establish this evaluation, let Boneh and Franklin's IBE scheme (BF-IBE) [16] be the underlying scheme.

4.4.1 Parameter Setting

Let $T \leftarrow t(k)/\epsilon(k)$, where T is the average expected computational times to succeed in breaking the enhanced IND-ID-CCA secure scheme. We review here that a $t_1(k)$-time adversary can break the underlying scheme with advantage $\epsilon_1(k)$ in plain FOPKC; a $t_4(k)$-time adversary can break the underlying scheme with advantage $\epsilon_4(k)$ in modified FOPKC; and a $t(k)$-time adversary can break the enhanced scheme with advantage $\epsilon(k)$ in both FOPKC.

If the value T_P (or T_M) is close to T, then the reduction is said to be tight, or efficient; contrarily, if T_P (or T_M) is much larger than T, then the reduction is said to be non-tight, or inefficient. If the security reduction if inefficient, then the adversary might not break the underlying IBE scheme in practical running time. This means the proof is less sound. We

derive the relation between T_P and T, the relation between T_M and T, and we compare the two relations.

In our evaluation, we set q_H, q_D and γ be $2^{60}, 2^{40}$ and 2^{-160}, respectively. Because q_H denotes the times to issue off-line hash queries, q_D denotes the time to issue on-line Decryption queries, and γ is identical to the inverse of the order of the underlying group in BF-IBE. We set l_2 be 2^{160} which is the bit length of the plaintext of both FOPKC conversions.

Regarding τ, in BF-IBE, to encrypt one message requires one *pairing computation* and this is the dominant part. The latest trustable researches show that, the running time of fastest pairing algorithms in software implementation [4] and hardware implementation [35] are about

$$\begin{cases} 4.33 \text{ miliseconds} & \text{in software implementation (AthlonXP 2GHz)} \\ 0.85 \text{ miliseconds} & \text{in hardware implementation (FPGA 15MHz)} \end{cases}$$

Thus, we set the running time τ of encryption to be these values.

4.4.2 T_P of Plain FOpkc

In the above setting, we evaluate T_P of plain FOPKC for BF-IBE:

$$\begin{aligned} T_P &\leq \frac{t + q_H q_D \tau}{(\epsilon + \frac{1}{2} - \frac{q_H}{2^{(l_1 - l_2)}})(1 - q_D\gamma) - \frac{1}{2}} \\ &\simeq \frac{t + 2^{100} \times \tau}{(\epsilon + \frac{1}{2} - 2^{60}/2^{140})(1 - 2^{40} \cdot 2^{-160}) - \frac{1}{2}} \simeq \frac{t + 2^{100} \times \tau}{\epsilon - 2^{-80}} \\ &\simeq T + 2^{110} \times \tau. \end{aligned}$$

After substituting τ, we obtain the additional cost to break BF-IBE scheme $(T_P - T)$ is

$$\begin{cases} 1.00 \times 2^{102} & \text{seconds } (\simeq 1.32 \times 10^{24} \text{ years}) \text{ in software implementation} \\ 0.85 \times 2^{100} & \text{seconds } (\simeq 0.11 \times 10^{24} \text{ years}) \text{ in hardware implementation,} \end{cases}$$

respectively. Each of them needs too long time to break BF-IBE, of course, it is impossible to calculate in the real world.

4.4.3 T_M of Modified FOpkc

In the above setting, we evaluate T_M of modified FOPKC for BF-IBE:

$$\begin{aligned} T_M &\leq \frac{t + q_H \tau}{(\epsilon + \frac{1}{2} - q_H/2^{l_1 - l_2})(1 - q_D\gamma) - \frac{1}{2}} \\ &\simeq \frac{t + 2^{60} \times \tau}{(\epsilon + \frac{1}{2} - 2^{60}/2^{140})(1 - 2^{40} \cdot 2^{-160}) - \frac{1}{2}} \simeq \frac{t + 2^{60} \times \tau}{\epsilon - 2^{-80}} \\ &\simeq T + 2^{70} \times \tau. \end{aligned}$$

After substituting τ, we obtain the additional cost to break BF-IBE scheme $(T_M - T)$ is

$$\begin{cases} 1.00 \times 2^{62} & \text{seconds } (\simeq 1.33 \times 10^{9} \text{ years}) \text{ in software implementation} \\ 0.85 \times 2^{60} & \text{seconds } (\simeq 2.83 \times 10^{8} \text{ years}) \text{ in hardware implementation,} \end{cases}$$

	$T_P - T$ of Plain FOPKC case	$T_M - T$ of Modified FOPKC case
Case 1	0.11×10^{23} years	2.83×10^{8} years
Case 2	1.32×10^{24} years	1.33×10^{9} years
Case 3	1.29×10^{15} years	1.30 years

Table 4.4: The Result of the Numerical Explanation

where **Case 1** denotes hardware implementation; **Case 2** denotes software implementation; **Case 3** denotes application of Moore's Law for software implementation using 10^6 PCs in 15 years after.

respectively. Thus the additional time cost in modified FOPKC is much smaller than that in plain FOPKC case.

4.4.4 Discussion

Since almost the whole additional time cost is pairing calculations in encryptions, these computations are easily parallelized. Nowadays, it is not difficult at all to gather computing resources such as million-order PCs [33] or to produce a number of specialized IC chips. For example, consider the case that an adversary, who is in software implementation, can gather only one million PCs' computation ability. In this case, the additional time cost is 1.33×10^3 years at present. By Moore's Law, this time cost will decrease to about 1.30 years in less than 15 years. Thus, this additional time cost will be feasible computable in near future.

Remark 4.1

The original Moore's Law derives from a speech given by Gordon Moore, later a founder of Intel, in 1965, in which he observed that the number of microcomponents that could be placed in an integrated circuit (microchip) of the lowest manufacturing cost was doubling every year and that this trend would likely continue into the future. As this observation and prediction began to be frequently cited, it became known as Moore's Law. In later years, the Law was occasionally reformulated to mean that rate. The pace of change having slowed down a bit over the past few years, the definition has changed (with Gordon Moore's approval) to reflect that the doubling occurs only every 18 months.

Due to the above discussion, we see that if there exists an adversary who can break our modified FOPKC in a realistic time, then it is also possible to break the underlying IBE scheme in almost the same computational time. On the other hand, as shown in Table 4.4, it is not clear whether the plain FOPKC provides the same level of security or not. Consequently, we can say that the modified FOPKC achieves *exact security* in a strict sense while the plain FOPKC does not.

4.5 Numerical Explanation by Group Size

In this section, we compare the group size required by the plain FOPKC and the modified FOPKC, and discuss what this result means in the realistic world.

In 2006, Gentry pointed out that concrete security and tight reductions are "utmost practical important" [29]. For example, since exponentiations in a group whose elements can be represented in r bits takes roughly $\mathcal{O}(r^3)$ time, this means that performing five 112-bit group exponentiations can be faster than one 192-bit group exponentiation. Thus finding as-tight-as-possible security reduction is very important for implementation.

To investigate how long the group needs to be for a security reduction, very recently research by Bellare and Ristenpart [12] proposed an evaluation framework. Their used a similar index of our previous results, say, they evaluated how efficient an adversary can break his target cryptographic scheme or hard problem. The requirement is: $T_P \leq T_Z$ and $T_M \leq T_Z$, where,

$$\begin{cases} T_P \leftarrow t_1(k)/\epsilon_1(k), \text{ in plain FOPKC} \\ T_M \leftarrow t_4(k)/\epsilon_4(k), \text{ in modified FOPKC} \\ T_Z \leftarrow t_z(k)/\epsilon_z(k), \text{ the adversary is against DBDH.} \end{cases}$$

Here, we again let Boneh and Franklin's IBE scheme (BF-IBE) [16] be the underlying scheme.

4.5.1 Parameter Setting

Let $T \leftarrow t(k)/\epsilon(k)$, where T is the average expected computational times to succeed in breaking the enhanced IND-ID-CCA secure scheme. We review here that a $t_1(k)$-time adversary can break the underlying scheme with advantage $\epsilon_1(k)$ in plain FOPKC; a $t_4(k)$-time adversary can break the underlying scheme with advantage $\epsilon_4(k)$ in modified FOPKC; and a $t_z(k)$-time adversary can break DBDH with advantage $\epsilon_z(k)$.

As a result,

$$\begin{aligned} T_P &\leftarrow e \cdot q_E \cdot (2^k + q_H \cdot q_D \cdot \tau \cdot \epsilon^{-1}) \\ T_M &\leftarrow e \cdot q_E \cdot (2^k + q_H \cdot \tau \cdot \epsilon^{-1}) \\ T_Z &\leftarrow 0.88\sqrt{p} \cdot T_{op}(\mathbb{G}_1) \end{aligned}$$

For concrete values, as our previous evaluation, we set γ be 2^{-160}, and set l_2 be 2^{160}. Furthermore, in our 6-D matrix for FOPKC on BF-IBE, we set security parameter be $\{60, 70, 80, 100, 128, 192\}$; group size be $\{80, 112, 128, 192, 256\}$; $\log \epsilon$ be $\{-10, -20, -30, -40\}$; $\log q_D$ be $\{10, 20, 30, 40\}$; $\log q_E$ be $\{10, 20, 30, 40\}$; and $\log q_H$ be $\{60\}$.

The most different part between this evaluation and previous evaluation is, here we focus on computational orders, not on concrete time.

4.5.2 Main Result

After a full computation on our matrix, 1920 elements have been verified. Generally speaking, all results are positive, i.e., our modified FOPKC only requires the same or even shorter group size than the plain FOPKC.

Table 4.5 shows comparison of pairing setups required to provably ensure security of plain FOPKC and modified FOPKC for various security level at security parameter k and values of ϵ, q_D and q_E. Here s(pFOPKC) represents plain FOPKC and s(mFOPKC) represents modified FOPKC.

4.6 Conclusions

This chapter investigated existing security enhancement means, which were developed for PKE, in the IBE environment. After observing the essence of inefficient part of plainFOPKC and FOCRYPTO, we proposed the cure: by adding identity information to the input of random oracle, one can easily achieve more efficient enhancements without any cost. At last, an intuitive numerical explanation was given to help readers understand the significance of our proposal.

k	$\log \epsilon$	$\log q_D$	$\log q_E$	s(pFOPKC)	s(mFOPKC)
60	−10	30	30	112	80
60	−10	40	20	112	80
60	−10	40	30	112	80
60	−20	20	30	112	80
60	−20	30	20	112	80
60	−20	30	30	112	80
60	−20	40	10	112	80
60	−20	40	20	112	80
60	−20	40	30	112	80
60	−20	40	40	128	112
60	−30	10	30	112	80
60	−30	20	20	112	80
60	−30	20	30	112	80
60	−30	30	10	112	80
60	−30	30	20	112	80
60	−30	30	30	112	80
60	−30	30	40	128	112
60	−30	40	10	112	80
60	−30	40	20	112	80
60	−30	40	30	128	80
60	−30	40	40	192	112
60	−40	10	20	112	80
60	−40	20	10	112	80
60	−40	20	20	112	80
60	−40	20	40	128	112
60	−40	30	10	112	80
60	−40	30	20	112	80
60	−40	30	30	128	112
60	−40	30	40	192	112
60	−40	40	10	112	80
60	−40	40	20	128	80
60	−40	40	30	192	112
60	−40	40	40	192	112
80	−30	40	40	192	112
100	−30	40	40	192	128
128	−30	40	40	192	192
192	−30	40	40	256	256

Table 4.5: Comparison of Pairing Setups.

Chapter 5 ON DESIGN OF EFFICIENT SCHEMES

If keeping 150-byte secret in your disk
will give you 20% faster encryption but no security risk,
won't you do that? Anyway, I will.

The concept of stateful encryption was introduced by Bellare et al. in 2006. Compared with a conventional public key encryption scheme, stateful encryption can surprisingly achieve much better encryption performance.

In this chapter, we introduce a related primitive called stateful identity based key encapsulation mechanism (SIBKEM). Together with IND-CCA secure symmetric encryption, SIBKEM implies stateful identity based encryption. We then demonstrate there is a generic construction of SIBKEM from a wide class of identity based non-interactive key exchange schemes. Also, we illustrate several instantiations.

It is preferable to construct cryptographic schemes based on only weak assumption. However, previous proposals of stateful encryption schemes are either based on strong assumptions; or admitting very *loose* security reductions. In this chapter, we improve these aspects by presenting a stateful identity based encryption scheme with tight security reduction to the computational bilinear Diffie-Hellman assumption. It is worth reminding that it is always desirable to have the proofs with tight reductions such that the actual schemes can be practically-meaningful.

At last, we note that our techniques of both formalizing SIBE and reducing assumption to weak one can be also applied to conventional public key settings. We propose a new primitive named stateful key encapsulation mechanism, and then show how to achieve stateful encryption by composing our primitive and symmetric encryption. Finally, we propose stateful public key encryption scheme based on computational Diffie-Hellman assumption.

5.1 Introduction

Public key encryption (PKE) is a very important tool for securing digital communicabilities. On the opposite of convenient key management functionalities, PKE schemes are often very slow compared with *symmetric encryption* (SE). In resource-constrained environment like mobile communications and sensor networks, this disadvantage of PKE will be quite undesirable, since system performance will drop greatly due to the high computational cost from frequent discrete modular exponentiations.

To improve the encryption performance of PKE, Bellare, Kohno and Shoup [9] introduced the concept of *stateful PKE* (SPKE) in ACM-CCS'06, where a sender maintains some state

information. Without loss of generality, the state information is divided into two parts: the secret part and the public part. Then the encryption algorithm takes as input not only a message and the public key of receiver, but also his current secret state to produce a ciphertext. As a result, the sender's computational cost for encryption is dramatically reduced. Decryption performance remains unchanged from stateless scheme, and the receivers need not even necessarily to notice if the sender is stateful if the public state is included in the ciphertext. Note that no such state information is required for either the sender or the receiver in conventional public key encryption schemes.

Regarding the security notions, the standard chosen ciphertext security (CCA) [32, 42] is modified to adjust a single-sender-multiple-receiver network, which in turn implies security of more general settings. According to whether the adversary is required to know the secret keys of the players other than its target, the model is further classified into *known secret key* (KSK) and *unknown secret key* (USK) settings. Apparently, the USK model is stronger and seems more realistic.

An *identity based encryption* (IBE) scheme is a special public key encryption scheme, where public keys can be arbitrary strings, advocated by Shamir [44] to simplify public key certificate management. The first fully functional construction was given by Boneh and Franklin [16], and many other researches followed [19, 13, 48, 29]. All the currently known efficient (IBE) schemes, are designed from pairing computation, which is known to be even heavier than discrete-exponentiation computation. Inspirited by [9], Phong, Matsuoka and Ogata [41] recently proposed the first "stateful identity based encryption" (StIBE) scheme based on the Boneh-Franklin IBE scheme [16]. Currently there is no known generic construction of SIBE schemes, and in this paper, we provide the first one.

5.1.1 Related Works

Bellare, Kohno and Shoup introduced the model of SPKE and proposed two constructions based on DHIES [1] and Kurosawa-Desdmet [36], respectively. Subsequently, Baek, Zhou and Bao [3] proposed a "generic" construction, and demonstrated many efficient instantiations. We remark that the "generic" construction requires additionally that underlying *key encapsulation mechanism* (KEM) [45] meets two non-standard properties: "partitioned" and "reproducibility". Thus their approach is not necessarily a real simplification for scheme designing. Paterson and Srinivasan [39] proposed a transformation from IBNIKE to a CPA secure IBE.

5.1.2 Contributions

In this chapter, we focus on SIBE, where the chosen identity security of SIBE schemes implies USK security [41]. We first introduce a simpler primitive called *stateful identity based KEM* (SIBKEM), which eventually enables a modular design approach for SIBE schemes, together with IND-CCA secure symmetric encryption. We formally give a composition theorem for such approach.

Next, we give a generic construction for SIBKEM based on so-called *identity based non-interactive key exchange* (IBNIKE). As its name suggests, an IBNIKE scheme is a non-interactive key exchange scheme that two players set up their shared key. Our construction is in a totally black-box manner: given any IBNIKE scheme, we can construct an SIBKEM scheme without essential modifications of the algorithms nor resorting to random oracles.

It has been known that NIKE schemes are closely related to PKE schemes. To illustrate, the well-known Diffie-Hellman key exchange is exactly the base of ElGamal encryption. However, this seems not so clear for stateful encryption schemes due to the introduction of state and chosen ID security into the model. In this paper, we present an affirmative answer to this question. More exactly, we show a large class of IBNIKE schemes is sufficient to build SIBKEM schemes, and therefore, SIBE schemes.

Next, we demonstrate several instantiations of our generic constructions and compare them with known stateful PKE schemes. Since our generic constructions make no number-theoretic assumptions, one can even construct SIBE schemes without pairings assumptions, with a cost of efficiency lost during secret key extraction.

Finally, we compare our proposal with previous SIBE schemes. We conclude that efficient instantiations of our generic construction are competitive to the most efficient schemes in the literature.

5.1.3 Security Notions of Stateful Encryption

Security Notion of SIBE

We establish the IND-ID-CCA (indistinguishability against adaptive chosen identity attack and adaptive chosen ciphertext attack) game for SIBE between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this games, the PPT adversary $\mathcal{A}$ tries to distinguish which plaintext was encrypted. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of SIBE. It passes the the resulting system parameters sp to $\mathcal{A}$ and keeps the masker key mk to himself. The state st is decided a-priori by $\mathcal{C}$.

Phase 1: $\mathcal{A}$ issues three types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Extraction queries on an identity id. $\mathcal{C}$ responds with a corresponding secret private key sk_{id} of id.

$\diamond$ Encryption queries on an identity and a message (id, m). $\mathcal{C}$ responds with ciphertext c of m under public key id and the current state st.

$\diamond$ Decryption queries on an identity and a ciphertext (id, c). $\mathcal{C}$ responds with the plaintext m of c, which is encrypted under the public key id.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that phase 1 is over, he outputs two equal length plaintext m_0, m_1 and an id^* on which he wishes to be challenged. The only restriction is that

id^* must not appear in any extraction query in phase 1. Then $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and sets $c^* \leftarrow \mathsf{Enc}(sp, id^*, st, m_b)$. $\mathcal{C}$ returns c^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Extraction queries on an identity $id \neq id^*$. $\mathcal{C}$ responds as in phase 1.

$\diamond$ Encryption queries on an identity and a message (id, m). $\mathcal{C}$ responds as in phase 1.

$\diamond$ Decryption queries on an identity and a ciphertext $(id, c) \neq (id^*, c^*)$. $\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. $\mathcal{A}$'s advantage in this IND-ID-CCA game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SIBE scheme is secure in the sense of IND-ID-CCA if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

Security Notion of SPKE

The first SPKE scheme was shown by Bellare, Kohno and Shoup [9]. Here, we review the model and then define the IND-CCA security in the USK model. Note that currently there is no SPKE scheme considering security in the CPA sense.

We establish the IND-CCA (indistinguishability against adaptive chosen ciphertext attack) game for SPKE between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the PPT adversary $\mathcal{A}$ tries to distinguish which plaintext was encrypted. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of SPKE. It then runs KeyGen to obtain a key pair (pk_1, sk_1) as the target. It passes the the resulting system parameters sp and the target public key pk_1 to $\mathcal{A}$ and keeps the secret key sk_1 as secret. $\mathcal{C}$ also sends all of the other secret keys $\{sk_2, \cdots, sk_n\}$ in the system to $\mathcal{A}$, where $sk_i \neq sk_1$. This captures the fact that $\mathcal{A}$ may corrupt all the entities other than his attack target. The state st is decided a-priori by $\mathcal{C}$.

Phase 1: $\mathcal{A}$ issues two types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Encryption queries on a public key and a message (pk_i, m), where $1 \leq i \leq n$. $\mathcal{C}$ responds with ciphertext c of m under public key pk_i and the current state st.

$\diamond$ Decryption queries on a ciphertext c. $\mathcal{C}$ responds with the plaintext m of c, which is encrypted under the target public key pk_1.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that phase 1 is over, he outputs two equal length plaintext m_0, m_1. Then $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and sets $c^* \leftarrow \mathsf{Enc}(sp, pk_1, st, m_b)$. $\mathcal{C}$ returns c^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Encryption queries on a public key and a message (pk_i, m). $\mathcal{C}$ responds as in phase 1.
$\diamond$ Decryption queries on a ciphertext $c \neq c^*$. $\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

We refer to such an adversary $\mathcal{A}$ as an IND-CCA adversary. $\mathcal{A}$'s advantage in this IND-CCA game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SPKE scheme is secure in the sense of IND-CCA if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

5.2 Stateful Identity Based Key Encapsulation Mechanism

In this section, we introduce the model and security notions of SIBKEM. Roughly speaking, SIBKEM is the "stateful version" of *conventional identity based key encapsulation mechanism* (IBKEM). In particular, in SIBKEM, the sender maintains a state information. And for a specified identity, the session key encapsulated by the sender remains the same unless the state is updated. Since it is deterministic, SIBKEM is weaker than IBKEM, i.e., the adversary can issue neither encapsulation query nor decapsulation query on the target identity.

5.2.1 Algorithms of SIBKEM

An SIBKEM scheme is specified by five algorithms. $\mathcal{SIBKEM} = \{$Setup, Ext, NwSt, Enc, Dec$\}$.

Setup: The randomized setup algorithm takes as input security parameter 1^λ where $\lambda \in \mathbb{N}$. It outputs the system parameters sp and the master key mk. It also specifies the key space $\mathcal{SHK}$ by sp. ($\mathcal{SHK}$ may be included in sp.) We write $(sp, mk) \leftarrow \mathsf{Setup}(1^\lambda)$.

Ext: The (possibly randomized) key extraction algorithms takes as input sp, mk and a user's identity id. It outputs a secret key sk_{id} corresponding to id. We write $sk_{id} \leftarrow \mathsf{Ext}(sp, mk, id)$.

NwSt: The randomized new state algorithm takes as input sp. It outputs a new state st of a sender. We write $st \leftarrow \mathsf{NwSt}(sp)$.

Enc: The deterministic encapsulation algorithm takes as input sp, id and st, where id is the receiver's identity. It outputs the corresponding ciphertext c of a session key dk. We write $(c, dk) \leftarrow \mathsf{Enc}(sp, st, id)$.

Dec: The deterministic decapsulation algorithm takes as sp, sk_{id} and a ciphertext c. It outputs the session key dk. We write $dk \leftarrow \mathsf{Dec}(sp, sk_{id}, c)$.

5.2.2 Security Notion of SIBKEM

We establish the IND-ID-CCA (indistinguishability against adaptive chosen identity attack and adaptive chosen ciphertext attack) game for SIBKEM between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of SIBE. It passes the the resulting system parameters sp to $\mathcal{A}$ and keeps the masker key mk to himself. The state st is decided a-priori by $\mathcal{C}$.

Phase 1: $\mathcal{A}$ issues three types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Extraction queries on an identity id. $\mathcal{C}$ responds with a corresponding secret private key sk_{id} of id.

$\diamond$ Encapsulation queries on an identity id. $\mathcal{C}$ responds with ciphertext c and a decryption key dk under id and the current state st.

$\diamond$ Decapsulation queries on an identity and a ciphertext (id, c). $\mathcal{C}$ responds with the decryption key dk of c, which is encapsulated under id.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that phase 1 is over, he outputs an id^* on which he wishes to be challenged. The only restriction is that id^* must not appear in any query in phase 1. Then $\mathcal{C}$ computes a valid key-ciphertext pair (c^*, dk_1^*) and flips a coin $b \in \{0, 1\}$. If $b = 0$, then $\mathcal{C}$ chooses a random key dk_0^* from the key space and returns (c^*, dk_0^*) to $\mathcal{A}$; otherwise $\mathcal{C}$ returns (c^*, dk_1^*).

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Extraction queries on an identity $id \neq id^*$. $\mathcal{C}$ responds as in phase 1.

$\diamond$ Encapsulation queries on an identity $id \neq id^*$. $\mathcal{C}$ responds as in phase 1.

$\diamond$ Decapsulation queries on an identity and a ciphertext $(id, c) \neq (id^*, c^*)$. $\mathcal{C}$ responds as in phase 1. Note that since the decapsulation algorithm is deterministic on fixed id and st, the restriction is actually $id \neq id^*$.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. $\mathcal{A}$'s advantage in this IND-ID-CCA game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SIBKEM scheme is secure in the sense of IND-ID-CCA if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

5.2.3 Composition Theorem

By combining an IND-ID-CCA secure $\mathcal{SIBKEM}$ = {SIBKEM.Setup, SIBKEM.Ext, SIBKEM.NwSt, SIBKEM.Enc, SIBKEM.Dec} and an IND-CCA secure $\mathcal{SE}$ = {SE.K, SE.E, SE.D}, we can obtain an IND-ID-CCA secure $\mathcal{SIBE}$ = {Setup, Ext, NwSt, Enc, Dec}. We

omit composition details since it is straightforward. At a high level, the SIBE sender uses SE.E to encrypt a message by using the key dk encapsulated by SIBKEM.Enc, and the SIBE receiver runs SE.D to decrypt with dk recovered by SIBKEM.Dec.

Theorem 5.1
Suppose $\mathcal{SIBKEM}$ is IND-ID-CCA secure, and $\mathcal{SE}$ is IND-CCA secure. Then the hybrid encryption scheme $\mathcal{SIBE}$ is IND-ID-CCA secure.

Proof of Security. We employ the game-based proof technique.
Game 0. Fix an efficient adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. We define Game 0 to be the attack game by $\mathcal{A}$ in the definition of IND-ID-CCA for SIBE. For proof convenience, we describe Game 0 as follows.

$$(sp, mk) \leftarrow \mathsf{Setup}(1^\lambda); st \leftarrow \mathsf{NwSt}(sp); (m_0, m_1, id^*) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp); b \leftarrow \{0,1\};$$
$$(c^*, dk_1^*) \leftarrow \mathsf{Enc}(sp, st, id^*); C^* \leftarrow \mathsf{E}(dk_1^*, m_b); b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, c^*, C^*)$$

In the above, we define E_0 to be the event that $b' = b$. Thus $\mathcal{A}$'s advantage is $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[E_0] - 1/2|$.
Game 1. The difference from Game 0 is that instead of encrypt m_b with dk_1^*, we encrypt it with randomly chosen $dk_0^* \in \mathcal{SHK}$. We describe Game 1 as follows. The box shows the difference.

$$(sp, mk) \leftarrow \mathsf{Setup}(1^\lambda); st \leftarrow \mathsf{NwSt}(sp); (m_0, m_1, id^*) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp); b \leftarrow \{0,1\};$$
$$(c^*, dk_1^*) \leftarrow \mathsf{Enc}(sp, st, id^*); \boxed{dk_0^* \leftarrow \mathcal{SHK}; C^* \leftarrow \mathsf{E}(dk_0^*, m_b);} b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, c^*, C^*)$$

Let E_1 be the event that $b' = b$ in Game 1.

Claim 1. $|\Pr[E_1] - 1/2| = \mathbf{Adv}_{\mathcal{B}_1}(\lambda)$. Here $\mathbf{Adv}_{\mathcal{B}_1}(\lambda)$ is the advantage of an adversary against $\mathcal{SE}$, and this advantage is assumed to be negligible. This follows from the fact that in Game 1, the encryption key dk_0^* is completely randomly distributed in $\mathcal{SHK}$.

Claim 2. $|\Pr[E_0] - \Pr[E_1]| = \mathbf{Adv}_{\mathcal{B}_2}(\lambda)$. Here $\mathbf{Adv}_{\mathcal{B}_2}(\lambda)$ is the advantage of an adversary against $\mathcal{SIBKEM}$, and this advantage is assumed to be negligible. The proof of Claim 2 is essentially the observation that in Game 0, the pair (c^*, dk_1^*) is real output from encapsulation algorithm, while in Game 1, a random dk_0^* is given instead. In this case, $\mathcal{A}$ should not notice the difference under the assumption that $\mathcal{SIBKEM}$ is secure. Rigorously, we construct a distinguishing algorithm $\mathcal{B}_2$ as follows.

Distinguisher $\mathcal{B}_2(c^*, dk^*)$
$(sp, mk) \leftarrow \mathsf{SIBKEM.Setup}(1^\lambda); st \leftarrow \mathsf{SIBKEM.NwSt}(sp);$
$(m_0, m_1, id^*) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp); b \leftarrow \{0,1\}; C^* \leftarrow \mathsf{E}(dk^*, m_b); b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, c^*, C^*);$
if $b' = b$ then output 1 else output 0

It is obvious that $\mathcal{B}_2$ interpolates between Game 0 and Game 1. If the input of $\mathcal{B}_2$ is the real output from encapsulation algorithm, then it works as same as Game 0. If the input of $\mathcal{B}_2$ is a ciphertext and a random key, then it works as same as Game 1.

Thus, the advantage of $\mathcal{B}_2$ against $\mathcal{SIBKEM}$ is equal to $|\Pr[E_0] - \Pr[E_1]|$. This completes the proof of Claim 2.

Combining Claim 1 and Claim 2, we have that $\mathbf{Adv}_{\mathcal{A}}(\lambda) = \mathbf{Adv}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}_{\mathcal{B}_2}(\lambda)$. Since $\mathcal{SIBKEM}$ and $\mathcal{SE}$ are secure, thus $\mathcal{A}$' advantage $\mathbf{Adv}_{\mathcal{A}}(\lambda)$ against $\mathcal{SIBE}$ is negligible. This completes the proof of Theorem 1. □

5.2.4 Generic Construction of SIBE

In this section, we propose a generic construction of stateful identity based key encapsulation mechanism. Our building block is identity based non-interactive key exchange (with mild requirements[1]). By applying our generic construction to various IBNIKE schemes, we can obtain SIBKEM schemes which provide various abstracting functionality.

Preparation

As described above, an IBNIKE scheme is specified by three basic algorithms, Setup, Ext, and Shr. To show the generic construction, in addition to these three basic algorithms, we require three additional algorithms which can be derived from the basic algorithms.

Sample: The randomized sample algorithm takes input as sp and output a temporary key pair $(pk, sk) \in \{\mathcal{PK}\} \times \{\mathcal{SK}\}$, where sk is the corresponding secret key to the public key pk. And the identifier of pk cannot be revealed. One can imagine that pk is the image of a virtual identifier id, and id must not be in collision with other realistic identities in the identity space.

Shr': If a party B has neither an identity nor an secret key, and B wants to exchange a key to a target party A with identity id_A, then Shr' takes as input (sp, sk_B, id_A), where sk_B is B's temporary secret key generated in Sample. It outputs a key $K_{A,B}$. Shr' is a deterministic algorithms.

Shr'': If a party A with identity id_A and secret key sk_{id_A} wants to exchange a key with a party B who does not have an identity but a temporary public key pk_B, then Shr'' takes as input (sp, sk_{id_A}, pk_B), where pk_B is generated in Sample. It outputs a key $K_{A,B}$. Shr'' is a deterministic algorithms.

We require the consistency of Shr' and Shr'' algorithms, i.e., if sk_{id_A} is secret key of id_A, and sk_B is secret key of pk_B, then $\mathsf{Shr'}(sp, sk_B, id_A) = \mathsf{Shr''}(sp, sk_{id_A}, pk_B)$, where $(pk_B, sk_B) \leftarrow \mathsf{Sample}(sp)$ and $sk_{id_A} \leftarrow \mathsf{Ext}(sp, mk, id_A)$.

At the first glance, these algorithms seem to require special properties to IBNIKE schemes, but as far as our best knowledge, it is easy to construct such algorithms for almost all currently known IBNIKE schemes. As an example, we illustrate a concrete construction below.

[1]Similar conditions to convert an IBNIKE scheme to an IND-ID-CPA secure IBE scheme can be found in [39].

From IBNIKE to SIBKEM

Let $\mathcal{IBNIKE}$ = {Setup, Ext, Shr, Sample, Shr', Shr''} be an IBNIKE scheme. By employing $\mathcal{IBNIKE}$ as buiding block, we show a generic construction of an SIBKEM scheme $\mathcal{SIBKEM}$ = {K.Setup, K.Ext, K.NwSt,K.Enc,K.Dec} as follows:

K.Setup: It takes as input 1^λ, and runs Setup of $\mathcal{IBNIKE}$ to obtain sp, mk, where sp contains a description of the shared key space $\mathcal{SHK}$. The output is (sp, mk).

K.Ext: It takes as input (sp, mk, id), and runs Ext of $\mathcal{IBNIKE}$ on (sp, mk, id) to obtain sk_{id} of an identity. The output is sk_{id}.

K.NwSt: It takes as input sp, and runs Sample of $\mathcal{IBNIKE}$ to obtain a temporary key pair $(\hat{pk}, \hat{sk})$. It sets $st \leftarrow (\hat{pk}, \hat{sk})$ and outputs st.

K.Enc: It takes as input (sp, id, st), parses st as $(\hat{pk}, \hat{sk})$, and then runs Shr' of $\mathcal{IBNIKE}$ on input $(sp, \hat{sk}, id)$ to obtain a key K. It sets the ciphertext $c \leftarrow \hat{pk}$, $dk \leftarrow K$, and outputs (c, dk).

K.Dec: It takes as input sp, sk_{id}, c, and runs Shr'' on input (sp, sk_{id}, c) to obtain the key K. It sets $dk \leftarrow K$, and outputs dk. According to the consistency of Shr' and Shr'', dk is the valid key outputed by K.Enc.

Security Proof

Here, we analyze the security of our generic construction. For proof convenience, we use the simulation-based proof technique. As described below, our proof has perfect simulation.

Theorem 5.2
Suppose $\mathcal{IBNIKE}$ is T2-IND secure. Then $\mathcal{SIBKEM}$ is IND-ID-CCA secure.

Main idea of the proof. Our strategy is as follows. Towards contradiction, we prove that if a scheme $\mathcal{SIBKEM}$ we constructed is *not* secure in the IND-ID-CCA sense, then the underlying scheme $\mathcal{IBNIKE}$ is *not* secure in the T2-IND. So we first assume there exists an IND-ID-CCA adversary $\mathcal{A}$ who can successfully break IND-ID-CCA with an advantage which is not negligible, then we show that we can construct a T2-IND adversary $\mathcal{B}$ who can successfully break T2-IND with an advantage which is not negligible.
Proof. Let $\mathcal{A}$ be an adversary against $\mathcal{SIBKEM}$, and $\mathcal{A}$'s advantage is $Adv_{\mathcal{A}}(\lambda)$. By using $\mathcal{A}$ as a subroutine, we construct an algorithm $\mathcal{B}$ who attempts to break T2-IND of $\mathcal{IBNIKE}$, and $\mathcal{B}$'s advantage is $\mathbf{Adv}_{\mathcal{B}}(\lambda)$. $\mathcal{B}$ plays T2-IND game interactively with a challenger $\mathcal{C}$.

The challenger $\mathcal{C}$ runs Setup of $\mathcal{IBNIKE}$, and obtains (sp, mk). $\mathcal{C}$ passes the system parameters sp to $\mathcal{B}$ and keeps the master key mk as secret.

$\mathcal{B}$ receives from $\mathcal{C}$ sp of $\mathcal{IBNIKE}$, and uses $\mathcal{A}$ to play against $\mathcal{C}$. $\mathcal{B}$ begins by drawing $id_B \leftarrow \{0,1\}^*$ and computing (pk_B, sk_B) where $pk_B \in \mathcal{PK}$ is the temporary public key of id_B, and $sk_B \in \mathcal{SK}$ is the temporary secret key. It sets $st \leftarrow (pk_B, sk_B)$, where st is

considered as the sender's current state. $\mathcal{B}$ then passes sp to $\mathcal{A}$. This ends the setup phase of $\mathcal{SIBKEM}$.

In phase 1 of $\mathcal{SIBKEM}$, $\mathcal{A}$'s extraction queriy on an identity id_A are answered by $\mathcal{B}$ by passing id_A to $\mathcal{C}$ as an extraction query in the T2-IND game. $\mathcal{A}$'s decapsulation query on an identity id_A and a ciphertext c are answered by $\mathcal{B}$ by first passing (id_B, id_A) to $\mathcal{C}$ as a reveal query in the T2-IND game. After obtaining $K_{A,B}$, $\mathcal{B}$ answers $\mathcal{A}$ with $dk \leftarrow K_{A,B}$.

When $\mathcal{A}$ submits challenge id^*, $\mathcal{B}$ submits its own challenge to $\mathcal{C}$ on input (id_B, id^*). Let b denote the random bit chosen by $\mathcal{C}$ in responding to $\mathcal{B}$'s challenge. $\mathcal{B}$ receives a key $K^* \in \mathcal{SHK}$ as result, and passes (K^*, pk_B) as the answer to $\mathcal{A}$. At last, $\mathcal{A}$ outputs its answer b'. $\mathcal{B}$ outputs this bit b' as his own answer to $\mathcal{C}$. The restriction is that id_B is distinct from id^*, and id^* is not involved in any query. Thus, $\mathcal{B}$'s advantage $\mathbf{Adv}_{\mathcal{B}}(\lambda)$ is as same as $\mathcal{A}$'s advantage $\mathbf{Adv}_{\mathcal{A}}(\lambda)$.

We have assumed that $\mathcal{A}$'s $\mathbf{Adv}_{\mathcal{A}}(\lambda)$ is not negligible, thus $\mathcal{B}$'s $\mathbf{Adv}_{\mathcal{B}}(\lambda)$ is also not negligible. We reach a contradiction to the hypothesis that $\mathcal{IBNIKE}$ is secure in the T2-IND sense. Thus $\mathcal{SIBKEM}$ is secure in the IND-ID-CCA sense. This completes the proof of Theorem 2. □

Theorem 5.3
Suppose the IBNIKE scheme $\mathcal{IBNIKE}$ is T1-IND secure. Then the SIBKEM scheme $\mathcal{SIBKEM}$ is IND-ID-CPA secure.

Proof.
This proof is similar as the proof of Theorem 1, thus we omit details. At a high level, the only difference is that since the adversary against $\mathcal{IBNIKE}$ cannot access to his Reveal oracle, he cannot simulate the Decryption queries from $\mathcal{SIBKEM}$. This results that $\mathcal{SIBKEM}$ is secure in the sense of IND-ID-CPA.

5.2.5 Instantiations and Comparisons

In this section, we demonstrate several instantiations of our generic construction from IBNIKE to SIBKEM. After applying our technique to Sakai, Ohgishi and Kasahara's IBNIKE scheme [43], we briefly discuss pairing-free SIBKEM (and so SIBE) schemes. At last we compare our technique to other related works.

Assumptions with bilinear maps. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties: (i) *Bilinearity*: For all $x, y \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}$, $e(x^a, y^b) = e(x, y)^{ab}$. (ii) *Non-degeneracy*: $e(g, g) \neq 1$. (iii) *Computability*: There is an efficient algorithm to compute $e(x, y)$ for any $x, y \in \mathbb{G}_1$.

The bilinear Diffie-Hellman (BDH) assumptions is that when given $\langle g^a, g^b, g^c \rangle$ there is no PPT algorithm $\mathcal{A}_{\mathbb{G}_1}$ can compute $e(g, g)^{abc}$ with non-negligible probability. That is, $\mathbf{Adv}^{\mathrm{bdh}}_{\mathbb{G}_1}(\mathcal{A}_{\mathbb{G}_1})$ is negligible. The decisional bilinear Diffie-Hellman (DBDH) assumption is that given $\langle g^a, g^b, g^c \rangle$ there is no PPT algorithm $\mathcal{A}_{\mathbb{G}_1}$ can distinguish $e(g, g)^{abc}$ from T

($G \leftarrow_R \mathbb{G}_2$) with non-negligible probability. The gap bilinear Diffie-Hellman assumption (GBDH) is that there is no PPT algorithm $\mathcal{A}_{\mathbb{G}_1}$ can compute $e(g,g)^{wxy}$ with non-negligible probability, even $\mathcal{A}$ is given oracle access to a decision BDH oracle $\mathcal{O}(\cdot,\cdot,\cdot,\cdot)$. That is, $\mathbf{Adv}^{\text{gbdh}}_{\mathbb{G}_1}(\mathcal{A}^{\mathcal{O}}_{\mathbb{G}_1})$ is negligible. Here, when queried by $\langle g^x, g^y, g^w, z\rangle$, $\mathcal{O}$ outputs 1 when $z = e(g,g)^{xyw}$, or outputs 0 otherwise.

SOK-based SIBKEM Instantiation

In 2000, Sakai, Ohgishi and Kasahara [43] proposed the first IBNIKE scheme SOK. Later in 2001, Boneh and Franklin [16] proposed the first IBE scheme. Although these two schemes aimed at different solutions, but the ideas were similar. The Type 2 security of SOK-IBNIKE can be reduced to decisional bilinear Diffie-Hellman assumption (DBDH).

The SOK scheme is described as follows.

$\mathsf{Setup}(1^\lambda)$: Choose a random generator $P \in \mathbb{G}_1$. Pick a random $s \in \mathbb{Z}^*_q$, and set $P_1 \leftarrow sP$. Choose cryptographic hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*_1$, and $H_2 : \mathbb{G}^*_2 \rightarrow \mathcal{SHK}$ as random oracles, where $\mathcal{SHK}$ is the key space. The system parameters are $sp \leftarrow \langle \mathbb{G}_1, \mathbb{G}_2, e, P, P_1, H_1, H_2\rangle$. The master key is $mk \leftarrow s$. Output (sp, mk).

$\mathsf{Extract}(sp, mk, id)$: For a given identity $id \in \{0,1\}^*$, set and output the secret key $sk_{id} \leftarrow sH_1(id)$.

$\mathsf{Shr}(sp, sk_{id_A}, id_B)$: Compute and output $K_{A,B} \leftarrow H_2(e(sk_{id_A}, H_1(id_B)))$.

Additional algorithms. Our three additional algorithms are constructed as follows.

$\mathsf{Sample}(sp)$: Choose a random $r \leftarrow \mathbb{Z}^*_q$, and set the temporary public key $pk_B \leftarrow rP$. The temporary secret key is $sk_B \leftarrow r$. Output (pk_B, sk_B).

$\mathsf{Shr'}(sp, sk_B, id_A)$: Compute and output $K_{A,B} \leftarrow H_2(e(sk_B \cdot P_1, H_1(id_A))$.

$\mathsf{Shr''}(sp, sk_{id_A}, pk_B)$: Compute and output $K_{A,B} \leftarrow H_2(e(pk_B, sk_{id_A}))$.

The Instantiation $\mathsf{Setup}(1^\lambda)$: Choose a random generator $P \in \mathbb{G}_1$. Pick a random $s \in \mathbb{Z}^*_q$, and set $P_1 \leftarrow sP$. Choose cryptographic hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*_1$, and $H_2 : \mathbb{G}^*_2 \rightarrow \mathcal{SHK}$ as random oracles, where $\mathcal{SHK}$ is the key space for symmetric encryption. The system parameters are $sp \leftarrow \langle \mathbb{G}_1, \mathbb{G}_2, e, P, P_1, H_1, H_2\rangle$. The master key is $mk \leftarrow s$.

$\mathsf{Extract}(sp, mk, id)$: For a given identity $id \in \{0,1\}^*$, set and output the secret key $sk_{id} \leftarrow sH_1(id)$.

$\mathsf{NwSt}(sp)$: Choose a random $r \leftarrow \mathbb{Z}^*_q$, and set the temporary public key $\hat{pk} \leftarrow rP$. The temporary secret key is $\hat{sk} \leftarrow r$. Set and output the new state $st \leftarrow (\hat{pk}, \hat{sk})$.

$\mathsf{Enc}(sp, id, st)$: Compute $dk \leftarrow H_2(e(\hat{sk} \cdot P_1, H_1(id))) = H_2(e(sP, H_1(id))^r)$. Set $c \leftarrow \hat{pk}$. Output (c, dk).

$\mathsf{Dec}(sp, sk_{id}, c)$: Compute and output $dk \leftarrow H_2(e(c, sk_{id})) = H_2(e(rP, sH_1(id)))$.

Pairing-Free SIBKEM Instantiations As built on numbers of historical works and recently formalized in [39], secure IBNIKE could be built from any trapdoor discrete log group.

The assumption is pairing-free, say, only *computational Diffie-Hellman* (CDH) assumption and random oracle. In particular, RSA-based IBNIKE arises from [34, 39]. The CDH-based IBNIKE is first proposed in [26], and subsequentially developed in [46]. Due to limitation of space, we omit the details.

By employing these IBNIKE as underlying schemes, we can obtain SIBKEM (and so SIBE) schemes without pairing. But the extraction algorithm will become greatly inefficient. Although in applications where the private key generator has great computational power this trade-off might be acceptable, we do not treat these instantiations as our main contribution in this paper.

Comparisons

Here, we compare our generic construction and instantiation with known results.

We first compare our generic construction (Ours1) with Beak et al.'s generic construction [3] (BZB08).

Scheme	BZB08	Ours1
Encryption algorithm type	Public key based	Identity based
security: *additional assumption*	KSK/IND-CCA: *None* USK IND-CCA: *random oracle*	IND-ID-CCA: *none*

Figure 5.1: Comparisons of Generic Constructions

Note that since our generic construction is in the identity based environment, the adversary can issue extraction queries to obtain secret keys of any identity other than the attack target. This means that the security of Ours1 should always considered in the USK model.

We then compare our SOK-based instantiation (Ours2) with Phong et al.'s scheme [41] (PMO08) and Boneh et al's scheme [16] (BF01).

Scheme	PMO08	BF01	Ours2
Stateful IBE?	Yes	No	Yes
Assumption	GBDH	BDH	DBDH
Tight security reduction?	Yes	No	Yes
Computation cost (encryption/decryption)	1 p/1 p	1 e + 1 p/1 p	1 p/1 p

For simplicity, the computation of the map-to-point hash functions (once for each scheme) is not evaluated due to light computation cost.

Figure 5.2: Comparisons of Stateful Identity-Based Encryption Schemes

The security of PMO08 is reduced to the GBDH assumption, however, we are not aware of any practical implementation for GBDH, since the decision problem is usually hard in $\mathbb{G}_2$.

We note that one can further weaken the underlying assumptions, with the price of either a loose security reduction by re-encryption checking techniques (e.g. FO-transform [25]), or a larger public key size due to the twin public key technique [21].

5.3 How to Remove Gap Assumptions and Maintaining Tight Reductions

In this section, we show a concrete stateful identity based encryption scheme whose IND-ID-CCA security can be reduced to the strong twin bilinear Diffie-Hellman assumption. This scheme is based on Boneh-Franklin [16] and is provably secure in the random oracle model.

5.3.1 Construction

$\mathsf{Setup}(1^\lambda)$**:** With input the security parameter 1^λ, generate two groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order p, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Choose a random group generater $g \in \mathbb{G}_1$, choose two random values $x_1 \in \mathbb{Z}$, $x_2 \in \mathbb{Z}$ and set $g_1 = g^{x_1}$, $g_2 = g^{x_2}$. Selects two hash function $H : \{0,1\}^n \leftarrow \mathbb{G}_1^*$ and $G : \mathbb{G}_2 \leftarrow \{0,1\}^k$. Let $param$ be $(\mathbb{G}_1, \mathbb{G}_2, p, e, n, g, g_1, g_2, H, G)$. Let mk be (x_1, x_2). Return $(param, mk)$ as the system parameter and the master key.

$\mathsf{Ext}(param, mk, id)$**:** For a given bit-string $id \in \{0,1\}^n$, compute $sk_{id,1} \leftarrow H(id)^{x_1}$, $sk_{id,2} \leftarrow H(id)^{x_2}$. Output $(sk_{id,1}, sk_{id,2})$ as the secret key corresponding to id.

$\mathsf{NwSt}(param)$**:** Pick r randomly from $\mathbb{Z}_p^*$ and set $R \leftarrow g^r$. Return (r, R) as output.

$\mathsf{Enc}(param, id, st, m)$**:** Parses st as (r, R), and set $y_i \leftarrow e(H(id), g_i)^r$ for i=1,2. Let $K \leftarrow G(R, y_1, y_2, id)$ and $c \leftarrow \mathsf{SEnc}(K, m)$. Return $C \leftarrow (c, R)$.

$\mathsf{Dec}(param, sk_{id}, C)$**:** Parse C as (c, R) and parse sk_{id} as $(sk_{id,1}, sk_{id,2})$. Let $y_i \leftarrow e(sk_{id,i}, R)$ for i=1,2. Compute $K \leftarrow G(R, y_1, y_2, id)$; $m \leftarrow \mathsf{SDec}(K, c)$.

5.3.2 Security Proof

Theorem 5.4

Let StIBE be the stateful identity based encryption scheme associated to group $\mathbb{G}_1$ and symmetric encryption scheme SE. Let $\mathcal{A}$ be an IND-ID-CCA adversary against StIBE. Then there exists a BDH adversary $\mathcal{A}_{\mathbb{G}_1}$ and an IND-CCA adversary $\mathcal{A}_{\mathsf{SE}}$ against SE such that

$$\begin{aligned} \boldsymbol{Adv}^{ind\text{-}id\text{-}cca}_{\mathsf{StIBE}}(\mathcal{A}) \quad &\leq \quad e \cdot (Q_{id}+1) \cdot \Big(\boldsymbol{Adv}^{bdh}_{\mathbb{G}_1}(\mathcal{A}_{\mathbb{G}_1}) + \\ &\qquad \boldsymbol{Adv}^{ind\text{-}cca}_{\mathsf{SE}}(\mathcal{A}_{\mathsf{SE}}) + \frac{2Q_g + Q_d}{p} \Big). \end{aligned}$$

Proof

The security proof of our proposal is quite similar to the one of the twin Boneh-Franklin, which is carefully discussed in appendix C of [21]. Thus we omit details here. The proof will be given in the full version of this paper.

5.4 Extension♯1: Stateful Key Encapsulation Mechanism

This section aims at the key encapsulation part of SPKE. We formalize this part as a cryptographic primitive named *stateful key encapsulation mechanism* (SKEM), which eventually enables a modular design approach for SPKE schemes, together with IND-CCA secure symmetric encryption. We formally give a composition theorem for such approach.

5.4.1 Algorithms of SKEM

In this section, we introduce the model and security notions of SKEM. Roughly speaking, SKEM is the "stateful version" of *conventional key encapsulation mechanism* (KEM). In particular, in SKEM, the sender maintains a state information. And for a specified public key, the session key encapsulated by the sender remains the same unless the state is updated. Since it is deterministic, SKEM seems to being capturing different security aspect from KEM, i.e., the adversary can issue neither encapsulation query nor decapsulation query on the target public key.

A SKEM scheme is specified by five algorithms. $\mathcal{SKEM} = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{NwSt}, \mathsf{Enc}, \mathsf{Dec}\}$.

Setup: The randomized setup algorithm takes as input security parameter 1^λ where $\lambda \in \mathbb{N}$. It outputs the system parameters sp which will be announced to all party involved in the system. It also specifies the key space $\mathcal{SHK}$ by sp. ($\mathcal{SHK}$ may be included in sp.) We write $sp \leftarrow \mathsf{Setup}(1^\lambda)$.

KeyGen: The randomized key generation algorithm takes as input sp. It outputs a key pair (pk, sk), where pk is a public key, and sk is the corresponding secret key.

NwSt: The randomized new state algorithm takes as input sp. It outputs a new state st of a sender. We write $st \leftarrow \mathsf{NwSt}(sp)$.

Enc: The deterministic encapsulation algorithm takes as input sp, pk and st, where pk is the receiver's public key. It outputs the corresponding ciphertext c of a session key dk. We write $(c, dk) \leftarrow \mathsf{Enc}(sp, pk, st)$.

Dec: The deterministic decapsulation algorithm takes as sp, sk and a ciphertext c. It outputs the session key dk. We write $dk \leftarrow \mathsf{Dec}(sp, sk, c)$.

5.4.2 Security Notion of SKEM

We establish the IND-CCA (indistinguishability against adaptive chosen ciphertext attack) game for SKEM between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the PPT adversary $\mathcal{A}$ tries to distinguish if $\mathcal{C}$ gives him a valid session key or a random key. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of SPKE. It then runs KeyGen to obtain a key pair (pk_1, sk_1) as the target. It passes the the resulting system parameters sp and the target public key pk_1 to $\mathcal{A}$ and keeps the secret key sk_1 as secret. $\mathcal{C}$ also sends all of the other secret keys $\{sk_2, \cdots, sk_n\}$ in the system to $\mathcal{A}$, where $sk_i \neq sk_1$. This captures the fact that $\mathcal{A}$ may corrupt all the entities other than his attack target. The state st is decided a-priori by $\mathcal{C}$.

Phase 1: $\mathcal{A}$ issues two types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Encapsulation queries on a public key pk_i, where $1 \leq i \leq n$. $\mathcal{C}$ responds with ciphertext c and a decryption key dk under id and the current state st.
$\diamond$ Decapsulation queries on a ciphertext c. $\mathcal{C}$ responds with the decryption key dk of c, which is encapsulated under the target public key pk_1.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that phase 1 is over, $\mathcal{C}$ computes a valid key-ciphertext pair (c^*, dk_1^*) and flips a coin $b \in \{0, 1\}$. If $b = 0$, then $\mathcal{C}$ chooses a random key dk_0^* from the key space and returns (c^*, dk_0^*) to $\mathcal{A}$; otherwise $\mathcal{C}$ returns (c^*, dk_1^*).

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Encapsulation queries on a public key pk_i. $\mathcal{C}$ responds as in phase 1.
$\diamond$ Decapsulation queries on a ciphertext $c \neq c^*$). $\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

We refer to such an adversary $\mathcal{A}$ as an IND-CCA adversary. $\mathcal{A}$'s advantage in this IND-CCA game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SKEM scheme is secure in the sense of IND-CCA if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

5.4.3 Composition Theorem

By combining an IND-CCA secure $\mathcal{SKEM} = \{$SKEM.Setup, SKEM.KeyGen, SKEM.NwSt, SKEM.Enc, SKEM.Dec$\}$ and an IND-CCA secure $\mathcal{SE} = \{$SE.K, SE.E, SE.D$\}$, we can obtain an IND-CCA secure $\mathcal{SPKE} = \{$Setup, KeyGen, NwSt, Enc, Dec$\}$. We omit composition details since it is straightforward. At a high level, the SPKE sender uses SE.E to encrypt a message by using the key dk encapsulated by SKEM.Enc, and the SPKE receiver runs SE.D to decrypt with dk recovered by SKEM.Dec.

Theorem 5.5
Suppose $\mathcal{SKEM}$ is IND-CCA secure, and $\mathcal{SE}$ is IND-CCA secure. Then the hybrid encryption scheme $\mathcal{SPKE}$ is IND-CCA secure.

Proof

We employ the game-based proof technique.

Game 0. Fix an efficient adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. We define Game 0 to be the attack game by $\mathcal{A}$ in the definition of IND-CCA for SPKE. For proof convenience, we describe Game 0 as follows.

$$
\begin{aligned}
&sp \leftarrow \mathsf{Setup}(1^\lambda);\\
&(pk_1, sk_1) \leftarrow \mathsf{KeyGen}(sp); \cdots ; (pk_n, sk_n) \leftarrow \mathsf{KeyGen}(sp);\\
&st \leftarrow \mathsf{NwSt}(sp);\\
&(m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp, sk_2, \cdots, sk_n);\\
&b \leftarrow \{0,1\};\\
&(c^*, dk_1^*) \leftarrow \mathsf{Enc}(sp, st, pk_1);\\
&C^* \leftarrow \mathsf{E}(dk_1^*, m_b);\\
&b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, sk_2, \cdots, sk_n, c^*, C^*)
\end{aligned}
$$

In the above, we define E_0 to be the event that $b' = b$. Thus $\mathcal{A}$'s advantage is $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[E_0] - 1/2|$.

Game 1. The difference from Game 0 is that instead of encrypt m_b with dk_1^*, we encrypt it with randomly chosen $dk_0^* \in \mathcal{SHK}$. We describe Game 1 as follows. The box shows the difference.

$$
\begin{aligned}
&sp \leftarrow \mathsf{Setup}(1^\lambda);\\
&(pk_1, sk_1) \leftarrow \mathsf{KeyGen}(sp); \cdots ; (pk_n, sk_n) \leftarrow \mathsf{KeyGen}(sp);\\
&st \leftarrow \mathsf{NwSt}(sp);\\
&(m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp, sk_2, \cdots, sk_n);\\
&b \leftarrow \{0,1\};\\
&(c^*, dk_1^*) \leftarrow \mathsf{Enc}(sp, st, pk_1);\\
&\boxed{dk_0^* \leftarrow \mathcal{SHK}; C^* \leftarrow \mathsf{E}(dk_0^*, m_b);}\\
&b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, sk_2, \cdots, sk_n, c^*, C^*)
\end{aligned}
$$

Let E_1 be the event that $b' = b$ in Game 1.

Claim 1. $|\Pr[E_1] - 1/2| = \mathbf{Adv}_{\mathcal{B}_1}(\lambda)$. Here $\mathbf{Adv}_{\mathcal{B}_1}(\lambda)$ is the advantage of an adversary against $\mathcal{SE}$, and this advantage is assumed to be negligible. This follows from the fact that in Game 1, the encryption key dk_0^* is completely randomly distributed in $\mathcal{SHK}$.

Claim 2. $|\Pr[E_0] - \Pr[E_1]| = \mathbf{Adv}_{\mathcal{B}_2}(\lambda)$. Here $\mathbf{Adv}_{\mathcal{B}_2}(\lambda)$ is the advantage of an adversary against $\mathcal{SKEM}$, and this advantage is assumed to be negligible. The proof of Claim 2 is essentially the observation that in Game 0, the pair (c^*, dk_1^*) is real output from encapsulation algorithm, while in Game 1, a random dk_0^* is given instead. In this case, $\mathcal{A}$ should not notice the difference under the assumption that $\mathcal{SKEM}$ is secure. Rigorously, we construct a

distinguishing algorithm $\mathcal{B}_2$ as follows.

$$
\begin{aligned}
&\text{Distinguisher } \mathcal{B}_2(c^*, dk^*)\\
&\quad sp \leftarrow \mathsf{SKEM.Setup}(1^\lambda);\\
&\quad st \leftarrow \mathsf{SKEM.NwSt}(sp);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp, sk_2, \cdots, sk_n);\\
&\quad b \leftarrow \{0,1\}; C^* \leftarrow \mathsf{E}(dk^*, m_b);\\
&\quad b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, sk_2, \cdots, sk_n, c^*, C^*);\\
&\quad \text{if } b' = b \text{ then output } 1 \text{ else output } 0
\end{aligned}
$$

It is obvious that $\mathcal{B}_2$ interpolates between Game 0 and Game 1. If the input of $\mathcal{B}_2$ is the real output from encapsulation algorithm, then it works as same as Game 0. If the input of $\mathcal{B}_2$ is a ciphertext and a random key, then it works as same as Game 1.

Thus, the advantage of $\mathcal{B}_2$ against $\mathcal{SKEM}$ is equal to $|\Pr[E_0] - \Pr[E_1]|$. This completes the proof of Claim 2.

Combining Claim 1 and Claim 2, we have that $\mathbf{Adv}_{\mathcal{A}}(\lambda) = \mathbf{Adv}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}_{\mathcal{B}_2}(\lambda)$. Since $\mathcal{SKEM}$ and $\mathcal{SE}$ are secure, thus $\mathcal{A}$' advantage $\mathbf{Adv}_{\mathcal{A}}(\lambda)$ against $\mathcal{SPKE}$ is negligible. This completes the proof of Theorem 1. □

5.5 Extension♯2: SPKE Based on Weak Assumption

In this section, we show a concrete stateful public key encryption scheme whose IND-CCA security can be reduced to the strong twin Diffie-Hellman assumption. This scheme is provably secure in the random oracle model and the unknown secret key model.

5.5.1 Construction

$\mathsf{Setup}(1^\lambda)$: Choose a random group generater $g \in \mathbb{G}$, and select a hash function H. Return (g, H) as the output $param$.

$\mathsf{KG}(param)$: Parse $param$ as (g, H) and choose two random elements x_1, x_2 from $\mathbb{G}$. Define the generated secret key $sk \leftarrow (x_1, x_2)$ and the corresponding public key $pk \leftarrow (y_1, y_2)$, where $y_1 \leftarrow g^{x_1}$ and $y_2 \leftarrow g^{x_2}$. Return (pk, sk).

$\mathsf{NwSt}(param)$: Pick r from $\mathbb{Z}_p$ at random. Parse $param$ as (g, H) and compute $R \leftarrow g^r$. Returns (r, R) as output.

$\mathsf{Enc}(param, pk, st, m)$: Parse $param = (g, H)$. Compute $K \leftarrow H(R, y_1^r, y_2^r)$; $c \leftarrow \mathsf{SEnc}(K, m)$; $C \leftarrow (c, R)$. Return $C = (c, R)$.

$\mathsf{Dec}(param, sk, C)$: Parse $C = (c, R)$. Compute $K \leftarrow H(R, R^{x_1}; R^{x_2})$; Return $m \leftarrow \mathsf{SDec}(K, c)$.

5.5.2 Security Proof

Theorem 5.6

Let StPKE be the stateful public key encryption scheme associated to group $\mathbb{G}$ and symmetric encryption scheme SE. Let $\mathcal{A}$ be an IND-CCA adversary against StPKE in the USK model. Then there exists a DH adversary $\mathcal{A}_{\mathbb{G}}$ and an IND-CCA adversary $\mathcal{A}_{\mathsf{SE}}$ against SE such that

$$\boldsymbol{Adv}^{ind\text{-}cca}_{StPKE}(\mathcal{A}) \leq \boldsymbol{Adv}^{dh}_{\mathbb{G}}(\mathcal{A}_{\mathbb{G}}) + \boldsymbol{Adv}^{ind\text{-}cca}_{SE}(\mathcal{A}_{SE}) + \frac{Q_h}{p}.$$

Proof

The security proof of our proposal is quite similar to the one of the twin ElGamal encryption scheme, which is carefully discussed in appendix B of [21]. The only thing should be considered is how to simulate random oracle.

5.6 Conclusions

In this chapter, we firstly proposed a cryptographic primitive called stateful identity based key encapsulation mechanism (SIBKEM). We defined the security notion, and showed that by combining secure SIBKEM and secure symmetric encryption, we can obtain secure stateful identity based encryption.

Secondly, we showed how to generically construct such SIBKEM scheme from a well-studied cryptographic primitive named identity based non-interactive key exchange (ID-NIKE). Although our discussion was only in identity based settings, but we note that part of our results could be applied to conventional public key settings.

We then showed how to remove gap-assumptions from stateful IBE schemes. Our constructions are efficient and with tight security reductions. However, we emphasize that all our discussions are within the random oracle model. How to build efficient stateful IBE schemes in the standard model under the same security definition is still open. Moreover, it would be interesting to reduce computational cost in encryption and decryption.

At last, we presented two extensions of our technique to stateful PKE settings. We introduced a cryptographic primitive named stateful key encapsulation mechanism. We also discussed how to achieve a stateful public key encryption scheme by composing this primitive and an IND-CCA secure symmetric key encryption. We then proposed a new SPKE scheme, trading assumptions/generality with computation costs.

Chapter 6 CONCLUSION

In this paper, we first proposed the definition framework and formally presented the definitions of the notions of security for $\mathcal{IBE}$ schemes, then rigorously proved the relations among these notions and concluded IND-ID-CCA2 is the adequate notion of security for $\mathcal{IBE}$ schemes. The significance of this result is that, from now on we have scientific evidence to claim proving $\mathcal{IBE}$ scheme secure in the sense of IND-ID-CCA2 is enough.

At the second stage, we first formalized the security model of forward secure identity based encryption scheme with master key update functionality. And then we proposed such a scheme whose security is based on basic complexity assumption and without relying on the random oracle model.

At the third stage, we confirmed the generic security of security enhancements (FOPKC, FOCRYPTO, and REACT) in $\mathcal{IBE}$ settings, and investigated the fact that there exists a significantly inefficient security reduction in the straightforward applications of both FOPKC and FOCRYPTO. Under this circumstance, we modified FOPKC and FOCRYPTO and reduced the security reductions down to acceptable values. In order to intuitively explain our solution, we presented numerical analysis by substituting proper concrete values. This "average success time" evaluation idea is quite important and common in cryptography research field, because security reduction is composed of two portions: advantage reduction and time reduction. We showed the modified FOPKC could achieve a satisfiable provable security, while the plain FOPKC could not.

At last, we studied application of stateful encryption in IBE settings. We formalized the essential part (SIBKEM) of SIBE and then showed a generic construction. Then we focused on concrete scheme and traded assumptions/generality with computation costs. Interestingly, our methodology does not stop only in SIBE field: it also affects SPKE research. By employing our technique, one can achieve SPKE scheme based on weak assumption, and also can formalize stateful (public key) key encapsulation mechanism.

Bibliography

[1] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In *CT-RSA'01*, volume 2020 of *LNCS*, pages 143–158. Springer, 2001.

[2] R. Anderson. Two remarks on public key cryptology. In *Invited Lecture, ACM-CCS '97*, 1997. `http://www.cl.cam.ac.uk/ftp/users/rja14/forwardsecure.pdf`.

[3] Joonsang Baek, Jianying Zhou, and Feng Bao. Generic constructions of stateful public key encryption and their applications. In *ACNS'08*, volume 5037 of *LNCS*, pages 75–93, 2008.

[4] P.S.L.M. Barreto. A note on effcient computation of cube roots in characteristic 3. Cryptology ePrint Archive, Report 2004/305, 2004. `http://eprint.iacr.org/2004/305`.

[5] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology - CRYPTO '98*, volume 1462 of *LNCS*, pages 26–45. Springer, 1998.

[6] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*. ACM, 1993.

[7] M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.

[8] M. Bellare and B. Yee. Forward security in private-key cryptography. In *Topics in Cryptology – CT-RSA '03*, volume 2612 of *LNCS*, pages 1–18. Springer, 2003. Preliminary version at `http://eprint.iacr.org/2001/035/`.

[9] Mihir Bellare, Tadayoshi Kohno, and Victor Shoup. Stateful public-key cryptosystems: how to encrypt with one 160-bit exponentiation. In *ACM CCS'06*, pages 380–389. ACM, 2006.

[10] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *ASIACRYPT'00*, volume 1976 of *LNCS*, pages 531–545, 2000.

[11] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008. Preliminary version appeared in [10].

[12] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' ibe scheme. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 407–424. Springer, 2009.

[13] D. Boneh and X. Boyen. Efficient selective-id identity based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT '04*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[14] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology - CRYPTO '04*, volume 3152 of *LNCS*, pages 443–459. Springer, 2004.

[15] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology - EUROCRYPT '05*, volume 3494 of *LNCS*, pages 440–456. Springer, 2005.

[16] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[17] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. Full version of [16].

[18] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, pages 647–657. IEEE Computer Society, 2007.

[19] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology - EUROCRYPT '03*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.

[20] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology - EUROCRYPT '04*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.

[21] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In *EUROCRYPT'08*, volume 4965 of *LNCS*, pages 127–145. Springer, 2008.

[22] Jean-Sébastien Coron. On the exact security of full domain hash. In *CRYPTO'00*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer, 2000.

[23] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography (extended abstract). In *STOC '91*, pages 542–552, 1991.

[24] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography - PKC '99*, volume 1560 of *LNCS*, pages 53–68. Springer, 1999.

[25] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.

[26] Steven D. Galbraith, Florian Hess, and Nigel P. Smart. Extending the ghs weil descent attack. In *EUROCRYPT'02*, volume 2332 of *LNCS*, pages 29–44. Springer, 2002.

[27] D. Galindo. Boneh-Franklin identity based encryption revisited. In *Proc. of 32nd ICALP*, volume 3580 of *LNCS*, pages 791–802. Springer, 2005.

[28] C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *Advances in Cryptology - ASIACRYPT '02*, volume 2501 of *LNCS*, pages 548–566. Springer, 2002.

[29] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.

[30] O. Goldreich. *Foundations of cryptography, Volumn II (revised, posted version Nr. 4.2).* Cambridge University Press, 2003. `http://www.wisdom.weizmann.ac.il/ oded/`.

[31] O. Goldreich, Y. Lustig, and M. Naor. On chosen ciphertext security of multiple encryptions. Cryptology ePrint Archive, Report 2002/089, 2002. `http://eprint.iacr.org/`.

[32] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[33] Website Information. RSA security – cryptographic challenges. `http://www.rsasecurity.com/`, RSA Security, available on Feb. 11, 2005.

[34] Masao Kasahara and Yasuyuki Murakami. Murakami-Kasahara ID-based key sharing scheme revisited - in comparison with Maurer-Yacobi schemes. Cryptology ePrint Archive, Report 2005/306, 2005.

[35] T. Kerins, W.P. Marnane, E.M. Popovici, and P.S.L.M. Barreto. Efficient hardware for the Tate pairing calculation in characteristic three. In *CHES 2005*, volume 3659 of *LNCS*, pages 412–426. Springer, 2005.

[36] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO '04*, volume 3152 of *LNCS*, pages 426–442. Springer, 2004.

[37] B. Libert and J.J. Quisquater. Identity based encryption without redundancy. In *ACNS*, volume 3531 of *LNCS*, pages 285–300. Springer, 2005.

[38] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology – CT-RSA '01*, volume 2020 of *Lecture Notes in Computer Science*, pages 159–174. Springer, 2001.

[39] Kenneth G. Paterson and Sriramkrishnan Srinivasan. On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. Journal version at DCC [40].

[40] Kenneth G. Paterson and Sriramkrishnan Srinivasan. On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. *Designs, Codes and Cryptography*, 52:219–241, 2009. Preliminary versions at Cryptology ePrint Archive: Report 2007/453 [39].

[41] Le Trieu Phong, Hiroto Matsuoka, and Wakaha Ogata. Stateful identity-based encryption scheme: Faster encryption and decryption. In *ASIACCS'08*, pages 381–388. ACM, 2008.

[42] C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '91*, volume 576 of *LNCS*, pages 433–444. Springer, 1991.

[43] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *SCIS'00*, pages 26–28, 2000.

[44] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.

[45] Victor Shoup. A standard for public-key encryption. ISO 18033-2, 2006.

[46] Edlyn Teske. An elliptic curve trapdoor system. J. Cryptology, 19(1):115–133, 2006.

[47] Y. Watanabe, J. Shikata, and H. Imai. Equivalence between semantic security and indistinguishability against chosen ciphertext attacks. In *Public Key Cryptography - PKC '03*, volume 2567 of *LNCS*, pages 71–84. Springer, 2003.

[48] B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT '05*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

[49] Danfeng Yao, Nelly Fazio, Yevgeniy Dodis, and Anna Lysyanskaya. Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In *ACM CCS*, pages 354–363. ACM, 2004.

Publicatoins

ジャーナル論文

<1> P. Yang, G. Hanaoka, Y. Cui, R. Zhang, N. Attrapadung, K. Matsuura, and H. Imai. Relations among notions of security for identity based encryption schemes. In **情報処理学会論文誌**, Vol.47, No.8, pages 2417–2429, 2006.

査読つき国際会議論文

<2> P. Yang, T. Kitagawa, R. Zhang, G. Hanaoka, K. Matsuura, and H. Imai. Applying Fujisaki-Okamoto to identity-based encryption. In *16th Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC '06)*, volume 3857 of *LNCS*, pages 183–192. Springer, Las Vegas, NV, USA, Feb. 2006.

<3> N. Attrapadung, Y. Cui, D. Galindo, G. Hanaoka, I. Hasuo, H. Imai, K. Matsuura, P. Yang, and R. Zhang (アルファベット音順に掲載). Relations among notions of security for identity based encryption schemes. In *7th Latin American Theoretical Informatics (LATIN '06)*, volume 3887 of *LNCS*, pages 130–141. Springer, Valdivia, Chile, Mar. 2006.

<4> T. Kitagawa, P. Yang, R. Zhang, G. Hanaoka, H. Watanabe, K. Matsuura, and H. Imai. Generic transforms to acquire CCA-security for identity based encryptions: the cases of FOPKC and REACT. In *11th Australasian Conference on Information Security and Privacy (ACISP '06)*, volume 4058 of *LNCS*, pages 348–359. Springer, Melbourne, Austrilia, Jul. 2006.

<5> P. Yang, T. Kitagawa, R. Zhang, G. Hanaoka, H. Watanabe, K. Matsuura, and H. Imai. A simple approach to evaluate fujisaki-okamoto conversion in identity based encryption. In *The 2006 International Symposium on Information Theory and its Applications (ISITA '06)*. Seoul, Korea, Oct. 2006.

<6> P. Yang, R. Zhang, and K. Matsuura. Stateful public key encryption: How to remove gap assumptions and maintaining tight reductions. In *The 2008 International Symposium on Information Theory and its Applications (ISITA '08)*, IEEE. Auckland, New Zealand, Dec. 2008.

<7> P. Yang and K. Matsuura. A forward secure identity based encryption scheme with master key update. In *The 2008 International Symposium on Information Theory and its Applications (ISITA '08)*, IEEE. Auckland, New Zealand, Dec. 2008.

<8> P. Yang, R. Zhang, K. Matsuura, and H. Imai. Generic Construction of Stateful Identity Based Encryption. In *The 12th Information Security Conference (ISC '09)*. Pisa, Italy, Sep. 2009. 採録済.

調査報告

<9> 楊鵬, 北田亘. ID ベース暗号の高機能化に関する動向調査. In *NTT* **受託調査報告書第二部**, pages 6–41, Mar. 2007.

査読無し国内発表論文

<10> 楊鵬, 花岡悟一郎, 崔洋, 張鋭, Nuttapong Attrapadung, 松浦幹太, and 今井秀樹. ID ベース暗号の安全性定義とそれらの関係. In *IEICE Technical Report, Vol.105, No.194*, pages 25–32. 電子情報通信学会, 盛岡, 日本, Jul. 2005.

<11> P. Yang, T. Kitagawa, R. Zhang, G. Hanaoka, K. Matsuura, and H. Imai. Towards security enhancement with efficient reduction for identity based encryption. In **第** *28* **回情報理論とその応用シンポジウム** *(SITA '05)*, pages 163–166, 沖縄, 日本, Nov. 2005.

<12> T. Kitagawa, P. Yang, R. Zhang, G. Hanaoka, K. Matsuura, and H. Imai. Applying fo-pkc99 and react to identity-based encryption. In *2006* **年暗号と情報セキュリティシンポジウム** *(SCIS '06)*. IEICE, 広島, 日本, Jan. 2006.

<13> 楊鵬, 松浦幹太. 上位秘密鍵生成センター構造法について. In *2006* **年コンピュータセキュリティシンポジウム** *2006 (CSS '06)*. IPSJ, 京都, 日本, Oct. 2006.

<14> P. Yang, T. Kitagawa, G. Hanaoka, R. Zhang, H. Watanabe, K. Matsuura and H. Imai. Security tightness evaluation of Fujisaki-Okamoto conversions in identity based encryption. In **第** *29* **回情報理論とその応用シンポジウム** *(SITA '06)*, 北海道, 日本, Nov. 2006.

<15> T. Kitagawa, P. Yang, G. Hanaoka, R. Zhang, H. Watanabe, K. Matsuura, and H. Imai. Means of Security Enhancement and Their Evaluation for Identity Based Encryption. In *2007* **年暗号と情報セキュリティシンポジウム** *(SCIS '07)*. IEICE, 長崎, 日本, Jan. 2007.

<16> P. Yang. Security Enhancement for Identity Based Encryption. In *Workshop: Research and Presentation*, Vol.19, pp150-176. 2007.

<17> P. Yang, K. Mizayaki, G. Hanaoka, K. Matsuura, and H. Imai. Security Notions and Proof of A Bit-wise Sanitizable Signature Scheme from Any One-way Permutation. In *2008* **年暗号と情報セキュリティシンポジウム** *(SCIS '08)*. IEICE, 宮崎, 日本, Jan. 2008.

<18> P. Yang and K. Matsuura. A Forward Secure Identity Based Encryption Scheme with Master Key Update. In **第** *31* **回情報理論とその応用シンポジウム** *(SITA '08)*, 栃木, 日本, Oct. 2008.

<19> P. Yang and K. Matsuura. A forward secure identity based encryption scheme with master key update. In **生産研究**, 60 巻, 5 号, pages 115–117, Sep. 2008.

<20> P. Yang, R. Zhang, K. Matsuura, and H. Imai. Generic Construction of Stateful Identity Based Encryption. In *2009* **年暗号と情報セキュリティシンポジウム** *(SCIS '09)*. IEICE, 滋賀, 日本, Jan. 2009.

<21> P. Yang, R. Zhang, K. Matsuura, and H. Imai. Stateful Key Encapsulation Mechanism. In **第** *46* **回コンピュータセキュリティ研究会研究発表会**. IPSJ, 秋田, 日本, Jul. 2009.

<22> G. Hanaoka, K. Miyazaki, P. Yang. Sequential Bitwise Sanitizable Signatures. In *The 1st Meeting for Cyrptology Frontier Group (***平成** *21* **年** *8* **月第** *1* **回暗号フロンティア研究会***)*. 石川, 日本, Aug. 2009. 発表予定.