Research Philosophy of Modern Cryptography[*]

Fuchun Guo[†], *Willy Susilo*, *Xiaofeng Chen*, *Peng Jiang*, *Jianchang Lai*, *Zhen Zhao*

May of 2023 (First Version)

Abstract

Proposing novel cryptography schemes (e.g., encryption, signatures, and protocols) is one of the main research goals in modern cryptography. In this paper, based on more than 800 research papers since 1976 that we have surveyed, we introduce the research philosophy of cryptography behind these papers. We use "benefits" and "novelty" as the keywords to introduce the research philosophy of proposing new schemes, assuming that there is already one scheme proposed for a cryptography notion. Next, we introduce how benefits were explored in the literature and we have categorized the methodology into 3 ways for benefits, 6 types of benefits, and 17 benefit areas. As examples, we introduce 40 research strategies within these benefit areas that were invented in the literature. The introduced research strategies have covered most cryptography schemes published in top-tier cryptography conferences.

[*] This work is based on our Chinese book《数字签名密史：从急需到有趣》(*Cryptologic Research History of Digital Signatures: From 1976 to 2020*) that was first published in November of 2022. The research philosophy has undergone a major revision.

Fuchun Guo and *Willy Susilo* are from the University of Wollongong, *Xiaofeng Chen* and *Zhen Zhao* are from the Xidian University, *Peng Jiang* is from the Beijing Institute of Technology, and *Jianchang Lai* is from the Southeast University.

[†] Supported by ARC Future Fellowship FT220100046 (*fuchun@uow.edu.au*).

Contents

1 Introduction

Throughout human civilization, cryptography has been recognized as a technique for encrypting and decrypting messages to maintain confidentiality for a very long time. In 1976, the concept of cryptography underwent significant evolution, following the publication of the seminal paper "*New Directions in Cryptography*" by Diffie and Hellman [56]. Since then, cryptography has been widely known as modern cryptography. Generally speaking, modern cryptography is the study of mathematical techniques that provide secure communication or computing in the presence of attacks from adversaries. It was developed to meet the increasing demands for security-related services in applications involving computers and networks, which are known as cyberspace nowadays.

Our community has made significant progress in the development of modern cryptography in the past five decades. Firstly, a significant number of cryptography primitives and notions have been invented to provide *confidentiality* or *integrity* services for *data, identity*, and *computing* to secure different application scenarios. Second, numerous organizations such as the International Association for Cryptologic Research (IACR) and individual researchers have successfully conducted conferences and workshops to promote research in the field of modern cryptography. So far, those conferences or workshops with a main interest in cryptography include Crypto, Eurocrypt, Asiacrypt, FSE, TCC, CHES, PKC, CT-RSA, FC, SCN, ACNS, PQCrypt, Latincrypt, SAC, ACISP, Inscrypt, Indocrypt, ICICS, ProvSec, CANS, ISPEC, IWSEC, and ICISC. Finally, a significant number of research papers have been published in the last thirty years. For example, more than 800 papers have been published in the year 2022 in the above conferences and workshops excluding other venues like security conferences and journals. As a result, the influence of cryptography is increasing rapidly, making it a crucial element in the modern era.

As researchers in modern cryptography, we are excited to witness the fast development of cryptography and its increasing impact. However, from another perspective, the rapid development accompanied by constantly updating research problems, technologies, notions, and knowledge has led to lots of beginners being left behind and lost because it is becoming hard for them to keep pace with the rate of published research outcomes. These beginners need to implement self-help in order to study cryptography and explore research. If there is a research philosophy for beginners systemically showing the whole map of cryptography research, it would help them find a way out of their predicament.

CONTRIBUTIONS. In this paper, we introduce the research philosophy of cryptography based on our perspectives on more than 800 academic papers (most are about digital signatures) that were published in cryptography conferences since 1976. We focus on understanding the philosophy for constructing new cryptography schemes (classified as the second scheme), on the assumption that there already exist applicable schemes (classified as the first scheme).

Starting from an application scenario, we explain how the researcher Alice makes contributions when she introduces a new cryptography notion and proposes the first scheme for this notion. We list potential factors in the literature used to evaluate this scheme. Next, we introduce the research philosophy called *"Above and Beyond"* for the researcher Bob when he wants to propose the second scheme. The above (benefit) and beyond (novelty) are to explore novel knowledge for producing more benefits (something in the second scheme better than the first scheme). We introduce the features of benefits in our community and have introduced 3 ways to explore benefits. These three ways have been expanded into 6 types of benefits and 17 benefit areas. Eventually, we introduce 40 research strategies showing what Bob can research in detail in Figure 1. To maintain consistency, in our introduction, we assume that the first scheme was proposed for a new notion called group signature and the second group signature scheme was proposed using different research strategies for different benefits. We also provide concrete examples from the literature. The introduced philosophy and our classification have covered most research outcomes published in peer-review cryptography conferences.

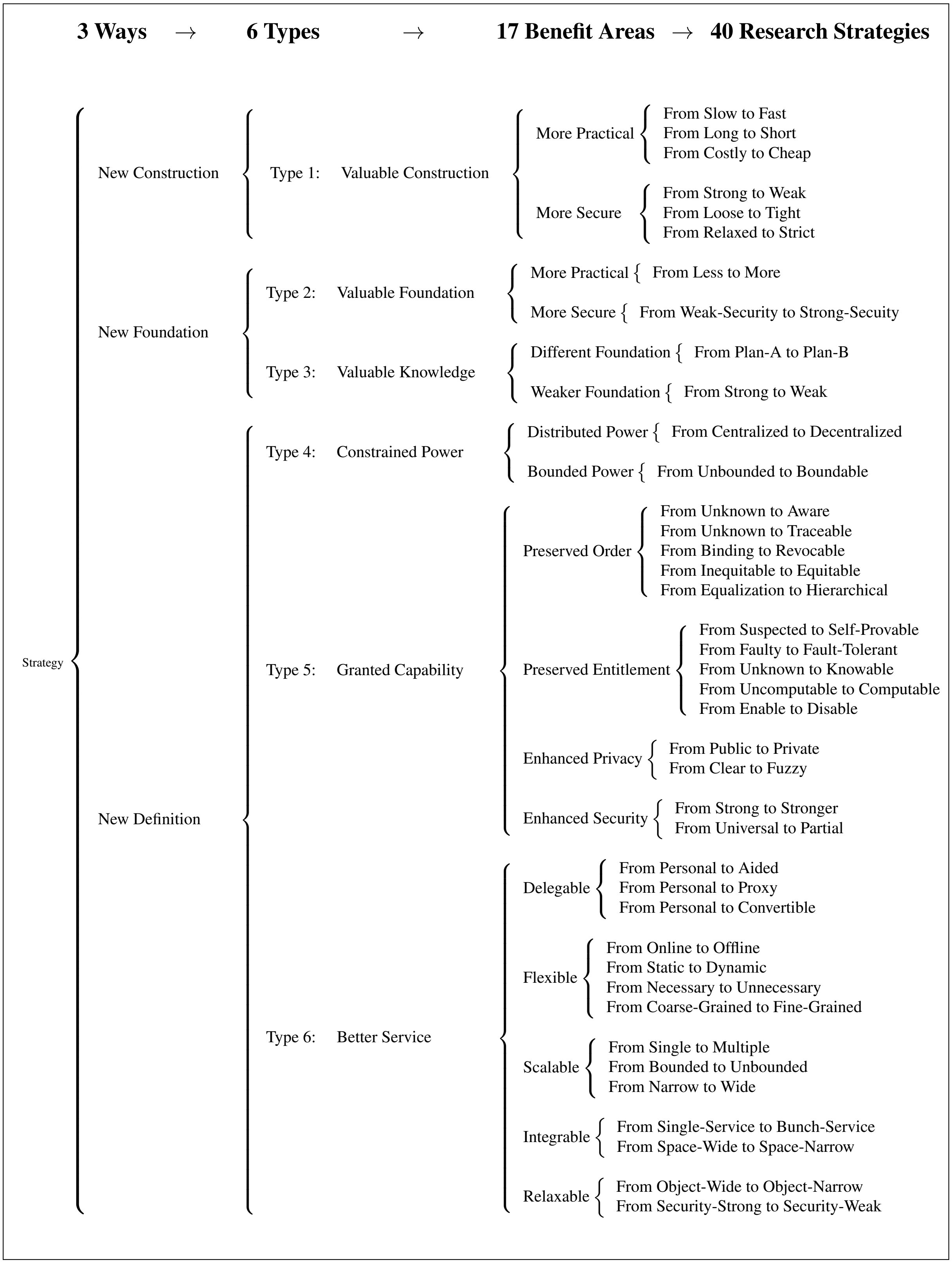

3 Ways → 6 Types → 17 Benefit Areas → 40 Research Strategies
Strategy
New Construction
Type 1: Valuable Construction
More Practical
From Slow to Fast
From Long to Short
From Costly to Cheap
More Secure
From Strong to Weak
From Loose to Tight
From Relaxed to Strict
New Foundation
Type 2: Valuable Foundation
More Practical
From Less to More
More Secure
From Weak-Security to Strong-Secuity
Type 3: Valuable Knowledge
Different Foundation
From Plan-A to Plan-B
Weaker Foundation
From Strong to Weak
New Definition
Type 4: Constrained Power
Distributed Power
From Centralized to Decentralized
Bounded Power
From Unbounded to Boundable
Type 5: Granted Capability
Preserved Order
From Unknown to Aware
From Unknown to Traceable
From Binding to Revocable
From Inequitable to Equitable
From Equalization to Hierarchical
Preserved Entitlement
From Suspected to Self-Provable
From Faulty to Fault-Tolerant
From Unknown to Knowable
From Uncomputable to Computable
From Enable to Disable
Enhanced Privacy
From Public to Private
From Clear to Fuzzy
Enhanced Security
From Strong to Stronger
From Universal to Partial
Type 6: Better Service
Delegable
From Personal to Aided
From Personal to Proxy
From Personal to Convertible
Flexible
From Online to Offline
From Static to Dynamic
From Necessary to Unnecessary
From Coarse-Grained to Fine-Grained
Scalable
From Single to Multiple
From Bounded to Unbounded
From Narrow to Wide
Integrable
From Single-Service to Bunch-Service
From Space-Wide to Space-Narrow
Relaxable
From Object-Wide to Object-Narrow
From Security-Strong to Security-Weak

Figure 1: Classifications of Research Strategies

2 Preliminaries

In general, modern cryptography is applied to provide services with the following distinct results:

- Legal users can do something with some secrets; while
- Illegal users cannot do the same thing without those secrets.

More precisely, modern cryptography can provide services of *confidentiality* or *integrity* for legal users in terms of *data, identity*, or *computing*. For example, cryptography can provide confidentiality during computing using fully homomorphic encryption [71]. That is, users can delegate cloud servers to compute $Enc_k(f(m))$ when only the function f and encrypted data $Enc_k(m)$ are provided (without knowing m). In the following of this paper, modern cryptography is called cryptography for short.

2.1 Cryptography Primitives and Cryptography Notions

The study of cryptography is classified into various cryptography primitives when focusing on specific scenarios and services. Nowadays, the developed cryptography primitives include: (1) symmetric-key encryption, (2) message-authentication codes, (3) public-key encryption, (4) digital signatures, (5) hash functions, and (6) cryptography protocols like zero-knowledge proof (ZKP) and multi-party computations (MPC). These primitives have significant differences in application scenarios or services. For example, encryption was proposed for data confidentiality, digital signatures were invented for data integrity, and MPC was motivated due to the need for interactive computing with confidential inputs.

Each cryptography primitive is composed of multiple pre-defined algorithms. Legal users will enjoy *algorithm functions* provided by these algorithms. For example, in digital signatures, a user can run key generation algorithm to generate a key pair (pk, sk) to become a signer, then this user can run signing algorithm to generate signatures on any messages, and any other users can run verification algorithm to verify signatures. In short, what kinds of services a cryptography primitive can provide is reflected by the defined algorithms and their functionalities.

Generally speaking, given one cryptography primitive, we can evolve variant *cryptography notions* from it after

- revising algorithm definition, and/or
- revising security definition.

A cryptography notion is evolved from early defined algorithms for cryptography primitive to further benefit users in applications via different angles[1]. In short, different cryptography primitives were introduced for various application scenarios, while different cryptography notions have similar application scenarios. For example, digital signatures can be evolved to aggregate signatures [29] which has two additional algorithms (namely adding two algorithms) and one can aggregate multiple signatures into a single one such that the other can still verify the validity of aggregate signatures. Digital signatures can be also evolved to online/offline signatures where the signing algorithm is replaced with two sub-algorithms, where the offline signing algorithm can generate "incomplete signatures" before knowing messages and the online signing algorithm can quickly generate signatures with incomplete signatures.

Before we go to the next introduction, we highlight the transitions of the above descriptions to clarify any confusion. Our purpose is not to give precise definitions of primitives and notions but help clear what we are going to introduce in this paper.

[1] We try to distinguish cryptography primitives from cryptography notions by the way that a primitive is the original of a cryptography notion in this paper, but they are mostly treated as the same outside this work.

Cryptography: Providing security services of confidentiality or integrity in terms of data, identity, and computing for legal users.

Cryptography Primitive: Basic algorithms for particular services or application scenarios motivated by confidentiality or integrity for data, identity, or computing.

Cryptography Notion: Evolved from the basic algorithms for a cryptography primitive with advanced definitions, and the defined algorithms or security are motivated by specific requirements in application scenarios.

We note that cryptography is different from cryptology, where the latter includes cryptography and cryptanalysis. The former focuses on how to construct a scheme (or protocol) for a cryptography notion. The study of cryptanalysis is to analyze existing primitives, notions, constructions, and the relations among them. The research philosophy we introduce is for cryptography only.

2.2 Research Aim and Research Challenge

The aim of cryptography research is mainly to propose a satisfactory scheme for a cryptography notion (including primitive). However, proposing a satisfactory scheme is a daunting task. The challenge is due to the fact that we must consider both the following two factors.

- **Practicality** for legal users (users in short). Users must run algorithms in order to enjoy security services, which are accompanied by time cost, memory cost, storage cost, communication cost, and implementation cost. For example, when a user generates a signature on a message, the size of signatures has impacted the communication cost and the storage cost.
- **Security** against illegal users (known as adversaries). Cryptography is designed for (legal) users to use, but adversaries are assumed to abuse the security services for benefits, who must be stopped. For example, a digital signature should be efficiently computable by a signer who has a signing key, but adversaries without knowing a signing key cannot find other efficient algorithms to generate signatures on behalf of the signer.

There is an inherent tradeoff between practicality and security according to the literature on research in cryptography when proposing schemes. Generally speaking, a practical scheme cannot achieve very strong security, while a very secure scheme is accompanied by inefficiency. It is therefore challenging to propose a perfect cryptography scheme.

3 First Scheme and Its Evaluation

Assuming that Alice has proposed a new cryptography notion $\mathcal{N}$ to secure a particular application. In this section, we will explain how Alice made contributions when proposing the first scheme for $\mathcal{N}$, as well as how to thoroughly evaluate her contributions.

3.1 The First Scheme

When proposing a new cryptography notion to meet a particular application scenario, Alice must conduct her research by following these three steps: Definition (♣), Foundation (♠), and Construction (★). The overview of these three steps will be expanded in Figure 2 as follows.

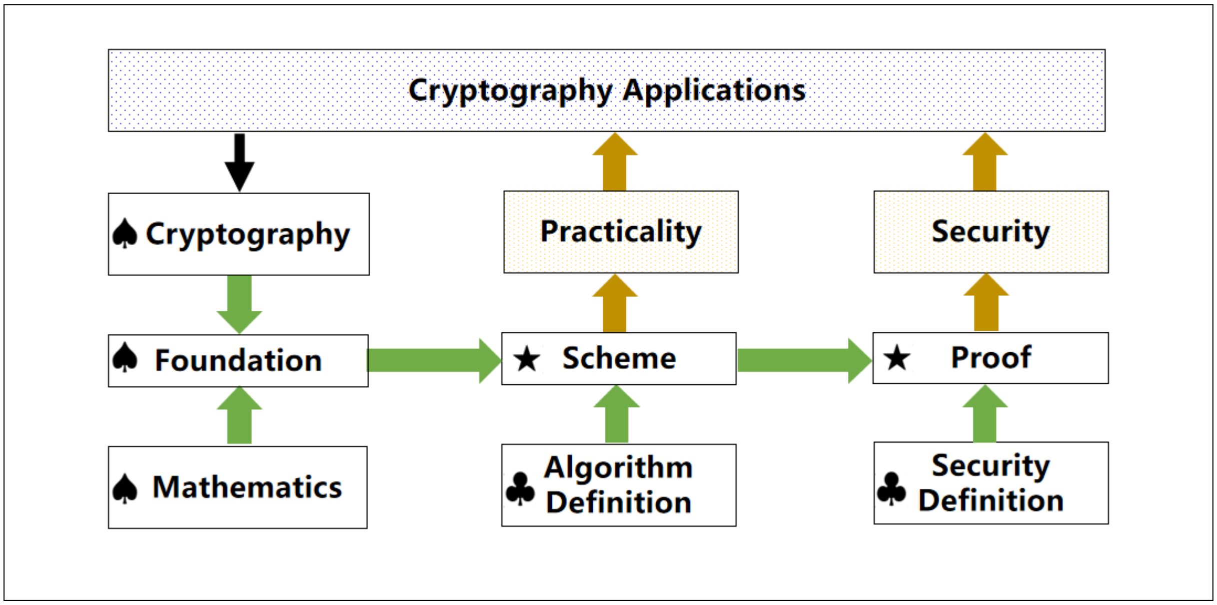


Figure 2: First Scheme and Its Evaluation.

When Alice wants to propose a new cryptography notion, the first step is to formalize the cryptography notion $\mathcal{N}$ with formal definitions including algorithm definition and security definition.

- **Algorithm definition** pre-defines all needed algorithms that are sound for users in applications. Most importantly, the input and output of all potential objects (e.g., message, signing key, public key, and signature) of each algorithm should be clearly stated. Generally speaking, after defining all algorithms, correctness is needed to guarantee that all algorithms are consistent and can work together to meet the requirements of applications.

- **Security definition** pre-defines the strength of security that a scheme to be proposed should achieve. The security definition is also known as the definition of security model, and it has three components: (1) the computing ability of adversaries that are mainly referred to as probabilistic polynomial-time (PPT) adversaries or quantum polynomial-time (QPT) adversaries[2], (2) what the adversary is allowed to know (query) before the attack, and (3) the goal of the attack launched by the adversary. A security definition ends with a negligible advantage of a successful attack.

When definitions are completed, the second step is to find a usable foundation denoted by F as the building block to "build" scheme. Notice that a scheme is probably constructed from multiple foundations. For simplicity, we treat adopted foundations as a single one only. There are two types of foundations: all potential mathematics and all existing cryptography.

[2] Here, adversaries are equivalent to Turing machines.

- **Mathematics** are the direct candidates that are used as building blocks to construct cryptography schemes. The developed mathematics in the literature include integer ring, cyclic groups, Elliptic-curve groups, bilinear pairing, error-correcting codes, multivariate, lattice, and isogeny. The mathematics define the rules of operations over well-defined elements. For example, in the integer ring, the operable elements are integers and the operations include modular addition and modular multiplication. Each mathematical foundation has its specific features and limitations when applied in scheme construction.

- **Cryptography** can be also served as foundations in scheme construction. That is, we can use all existing cryptography schemes as foundations to construct new cryptography schemes. We can also use existing cryptography primitives or notions as foundations to propose schemes that is known as generic construction, and it means that we can use any scheme from this primitive or notion to construct schemes. For example, a generic construction of a digital signature scheme from identification protocol [63]. We note that the most fundamental primitive of cryptography is one-way function.

Once definitions and foundations are set down, the last step is to construct a scheme from the foundation F and then prove its security in a security model. When we say "constructing a scheme", it refers to the process of describing computations of all pre-defined outputs from pre-defined inputs step by step for all pre-defined algorithms.

- **Scheme** is the process of algorithm construction (design operations inside algorithms step by step) S from the chosen foundation F, namely $\mathsf{F} \rightarrow \mathsf{S}$. The proposed algorithms must be friendly for users. Most importantly, when running algorithms, the computation time and communication cost must be acceptable (known as polynomial time and polynomial size).

- **Proof** is the process of analyzing the security of S under a hardness assumption denoted by P, namely $\mathsf{S} \succ \mathsf{P}$[3]. That is, if P is indeed hard, then S must be secure (in the defined security model). Our community has invented many formal methods for proving security such as security reduction in game-based proof for schemes like encryption and signature schemes, and simulation-based proof for protocols like MPC protocol.

$$\text{Research Steps} = \begin{cases} \text{Step 1: Definition} & \begin{cases} \text{Algorithm Definition} \\ \text{Security Definition} \end{cases} \\ \\ \text{Step 2: Foundation} & \begin{cases} \text{Foundation from Mathematics} \\ \text{Foundation from Cryptography} \end{cases} \\ \\ \text{Step 3: Construction} & \begin{cases} \text{Scheme Construction} \\ \text{Security Proof} \end{cases} \end{cases}$$

We note that scheme construction and security proof in the literature were mostly treated as two simultaneous process. That is when we construct a scheme, we must also consider how to prove its

[3] Our description is equivalent to "P is reducible to S" in the research community of computational complexity theory.

security at the same time. Otherwise, the process might be stuck when proving the security of proposed schemes. In other words, we might have to adjust the scheme construction to meet provable security.

In summary, the contributions made by Alice are composed of three parts: (1) a proper definition for the cryptography notion $\mathcal{N}$, (2) an applicable foundation F that has been successfully identified by Alice, and (3) a scheme construction from S that is provably secure. We emphasize that Alice's contributions are not just the concrete scheme construction but all knowledge towards successfully proposing the first scheme.

3.2 Evaluation

The evaluation of Alice's contributions depends on what she has contributed. Since she proposed the new cryptography notion and the first scheme, the evaluation can also be classified into three categories, known as Definition, Foundation, and Construction. Next, we expand on what to evaluate in these three categories.

The evaluation of *Definition* is to evaluate the algorithm functions that have been defined for the notion $\mathcal{N}$ and the strength of security models defined for the notion $\mathcal{N}$.

- **Algorithm Definition.** Each algorithm provides one function for users (e.g. key generation algorithms for generating a key pair) and each cryptography notion always provides a limited number of algorithms. Further, the input and output of each algorithm decide the scalability and flexibility of running algorithms. Taking the verification algorithm of digital signatures denoted by $\mathsf{Verify}(pk, m, \sigma_m)$ as the example, the algorithm can only input one signature for verification and the signature must be in plaintext (e.g., cannot be encrypted). The algorithm definition could be found no longer suitable when the application scenario has been slightly changed.

- **Security Model.** A security model defines who the adversary is, what the adversary knows, and what the adversary wants to attack. The evaluation of the security model usually revisits whether the definition of "who the adversary is", "what the adversary knows", and "what the adversary wants to attack" matches the application scenario or not. The security definition could be too weak and needs to be strengthened, or too strong and can be relaxed.

The evaluation of *Foundation* is to evaluate how good the adopted foundation is when it is applied in the scheme construction. There are two ways to measure the foundation: practicality and hardness.

- **Practicality.** The adopted foundation affects the practicality of the proposed scheme. For example, a scheme constructed from RSA-like integer ring [127] is usually less efficient than that from elliptic curve cyclic group [97] in terms of the output size. Generally speaking, a scheme is constructed from multiple foundations and the practicality could be impacted due to using one less-practical foundation. For example, a hash function $H : \{0,1\}^* \rightarrow \mathbb{S}$ must be employed in the scheme construction and $\mathbb{S}$ refers to the set of large prime numbers [70] or group elements [32], which is harder to be constructed than $\mathbb{S}$ referring to fixed-length bit strings.

- **Hardness.** The adopted foundation also affects the security of the proposed scheme. The adversary cannot break a proposed scheme from the foundation because some computing problems over the foundations are computationally hard, but the hardness based on the different foundations is not the same. For example, the hardness of the Discrete logarithm problem from modular multiplicative group and from elliptic curve group with the same size of group elements are not identical [97]. How hard a problem is based on the adopted foundation is to be evaluated in this way.

The evaluation of *Construction* is to evaluate how good the first scheme is in terms of practicality and security. The practicality is about the price to pay for users to enjoy each algorithm function. The security describes how hard for adversaries to break the proposed scheme S or how convincing the security of the proposed scheme is. Our community mostly evaluates the practicality and the security of S via the following factors.

- **Computation time** of obtaining the output from its input. In many cases, a theoretical analysis is clear enough. In the complexity level, the time is classified into constant, logarithm, or linear in the size of the input and the operation unit depends on the chosen foundation such as one exponentiation or one hash operation. When it is constant, our community is interested in knowing a more precise cost, such as two exponentiations or more than two.
- **Output size** for representing the output from its input. Similarly, in the complexity level, the size is classified into constant, logarithm, or linear in the size of the input and the size unit depends on the chosen foundation such as one group element or one hash value. When the size is constant, our community is also interested in knowing a more precise size, such as three group elements or more than three. The output size impacts the storage cost or communication cost.
- **Interaction cost** for completing running an algorithm. Usually, this kind of algorithm exits in cryptography protocols where multiple parties with secret individual inputs interactively run an algorithm. Our community cares about two kinds of interaction costs: communication costs sent among users and communication rounds. One of an example of considering interaction cost is the blind signatures [68] that need interactions between the signer and receiver when they generate a blind signature. We note that the communication cost of the interaction is about the intermedia communication cost consumed by algorithms and they are different from the output size.
- **Security model** adopted in security proof for the first scheme. There could be more than one security model with different levels of security strength proposed for the cryptography notion. Our community is interested in knowing how strong the security model is adopted for the proposed scheme S. The strength of the security model significantly affects the difficulty of constructing efficient schemes and proving security.
- **Hard problem** adopted as the underlying hardness of breaking the proposed scheme S. The complexity of solving different computing problems over the same foundation is mostly different. Some underlying hardness assumptions are standard in use but some of them like q-type assumptions [25] or oracle-based assumptions [123] are very strong. Further, some underlying hardness assumptions such as lattice-based problems are still believed-to-be hard against quantum computers, while some of them such as the factoring problem are easy using quantum computers.
- **Proof model** adopted to model adversary in computing. In the standard proof model, there is no restriction on the adversary except advantage, time, and computing power (such as PPT or QPT). Unfortunately, it could be hard to prove security with such a proof model. Our community is trying to use some relaxed proof models to prove security. The relaxed proof models in the literature include Random Oracle Model [18], Generic Group Model [132], Algebraic Group Model [66], and Common-Reference String Model [64].
- **Proof result** mainly refers to the concrete loss during the security reduction. A security reduction is tight if the prover can solve a hard problem with an advantage very close to that of breaking the proposed scheme from adversaries [17]. A tight reduction guarantees that the lower bound

complexity of breaking the proposed scheme is close to that of solving the hard problem. A security reduction is called memory-tight [11] if the amount of working memory used by the prover is close to the adversary. It has been shown that some security reductions yield less meaningful security guarantees if the memory cost is not tight. The proof result therefore decides the concrete security of the proposed scheme.

It is worth noting that the security of a scheme is usually proven against adversaries who launch queries sequentially in a security model. Our community also considers concurrent security in a security model especially for protocols, because some sequentially secure schemes (protocols) were found to be not concurrently secure [60]. In terms of the proof model, universal composability (UC) framework [40] was invented to analyze the security of protocols in modular, and the UC model can be seen as a model better or stronger than a standalone model.

$$\text{Evaluation} = \begin{cases} \text{Evaluation of Definition} & \begin{cases} \text{Algorithm Definition} \\ \text{Security Model} \end{cases} \\ \\ \text{Evaluation of Foundation} & \begin{cases} \text{Practicality} \\ \text{Hardness} \end{cases} \\ \\ \text{Evaluation of Construction} & \begin{cases} \text{Computation Time} \\ \text{Output Size} \\ \text{Interaction Cost} \\ \text{Security Model} \\ \text{Hard Problem} \\ \text{Proof Model} \\ \text{Proof Result} \end{cases} \end{cases}$$

Our evaluation has some overlaps between the evaluation of the hardness of the foundation and the hard problem adopted for security proof. The former is to evaluate the security of the foundation (e.g., evaluate the most fundamental hard problem Discrete Logarithm problem) while the latter is to evaluate the security of the proposed scheme (e.g., the adopted q-SDH problem for provable security). Our evaluation also has some overlap between the evaluation of the security definition and the security model. Similarly, the former is to evaluate the notion (e.g., the strongest and the weakest security models) while the latter is to evaluate the security of the proposed scheme (e.g., the adopted security model is neither the strongest nor the weakest).

4 Second Scheme and Its Research Philosophy

Bob, another researcher, is interested in finding solutions to secure the X application, but Alice has already introduced the cryptography notion $\mathcal{N}$ to secure this application and contributed $\mathsf{F} \rightarrow \mathsf{S} \succ \mathsf{P}$ in the first scheme construction. An inherent question is:

What is Bob's motivation for continuing the research on X application after Alice?

A straightforward answer is to improve Alice's first scheme to be more friendly for users to use the scheme, and (or) harder for adversaries to break the scheme. But this is not what all Bob can do. In this section, we introduce the proper research philosophy for Bob to continue his research.

4.1 Research Philosophy

Based on our observations on the literature, a proper research philosophy suitable for cryptography is called *above and beyond*. The former represents benefits and the latter represents novelty. More precisely, Bob is to explore new knowledge for constructing the second scheme, such that

- (**Above**) the second scheme brings more benefits for certain users than the first scheme, and
- (**Beyond**) the second scheme brings more benefits because of contributing novel knowledge.

The "benefit" is something good for users who run cryptography algorithms. For example, it is faster for users to generate signatures and harder for adversaries to forge signatures without secret keys in the second scheme than in the first scheme. Further, there could be more than one type of users, such as signers and verifiers, in the application scenario secured by a cryptography notion. We can explore the benefits for one type of users only. For example, how to improve the signing efficiency for signers in digital signatures. We notice that whether the benefits are convincing or not in the second scheme depends on the narrative created by Bob, especially when the benefits are not obvious to reviewers (readers). We will expand on the detail of benefits in the next subsection.

The "novelty" is about how much new the contributed knowledge is used to produce more benefits in the second scheme. However, the definition of novelty is rather vague without a clear quantity measurement, and it heavily depends on the reviewers' deep knowledge and feeling when the second scheme is submitted for publication consideration. Here, we give two opposite examples. The novelty of the second scheme is probably low if it is constructed from existing methods after straightforward modifications, no matter how many benefits the second scheme brings. While the novelty of the second scheme is probably high if it has successfully solved a long-term open problem which no one knows how to solve with all existing methods.

In summary, research is to produce novel knowledge and the additional benefits[4] are the direct evidence showing the novelty of knowledge. Nevertheless, it's important to note that bringing benefits does not necessarily mean that the knowledge must be novel. In this study, we will not be focusing on the concept of novelty in the field of cryptography research, as there is no standard method, but rather on understanding the benefits due to novel knowledge.

The "above and beyond" is a less restrictive and more precise philosophy compared to "better than the first scheme". By focusing on the contribution of novel knowledge, novelty is recognized as bringing additional benefits for certain users. Ultimately, it is hoped that the collective efforts of individual researchers will lead to significant changes that benefit the entire human population. For example, the knowledge for constructing signature schemes with faster signing and the knowledge for constructing signature schemes with faster verification can be later qualitatively merged into novel knowledge used to construct a more practical signature scheme that benefits both signers and verifiers.

4.2 Developed Benefits

The aims of proposing new schemes are to produce more benefits related to practicality or security with the help of novel knowledge. In this section, we briefly summarize what the developed benefits look like in our community.

[4] The benefits from the second scheme subtract those from the first scheme.

- Developed Benefits
  - Claimed Benefits
    - Quantity Improvement
    - Desired Property
  - Produced Benefits
    - Can be little
    - Can be blurred
    - Can have tradeoff

The claimed benefits in the second scheme are to show more benefits. There are two ways of parsing the word "more" which originally means a greater quantity or something additional.

- **The first parse is called quantity improvement**. In particular, the price paid by users to enjoy security services is reduced and therefore benefits users, or the price paid by adversaries to abuse security services is increased. For example, users can complete computations faster, users can store and send a short ciphertext instead of a long ciphertext, or users can pay a lower price to purchase products due to lower-cost implementation. Another example is tight security proof [17] with a smaller reduction loss as it is related to the efficiency of computation, output size, and communication cost.
- **The second parse is called desired property**. The desired property refers to some nice features that correspond to the benefits of practicality or security. If the second scheme has a property while the first one does not have this, it implies that this scheme owns particular benefits from this property. For example, security proof without random oracles in the second scheme makes its security more convincing [41] than that using random oracles in the first scheme.

The quantity improvement and desired property are two concrete claims applied to argue that the second scheme is somehow more practical or more secure than the first scheme. We note that quantity improvement can be also treated as desired property. For example, improving signing efficiency is a quantity improvement but efficient signing compared to inefficient signing becomes a desired property.

Assuming that the second scheme has contributed novel knowledge. An inherent question is: how obvious the additional benefits are in the second scheme when compared to the first scheme? The literature shows that the second scheme could produce benefits that are easily understandable and obvious, but also produce non-straightforward benefits. There are three types of non-obvious produced benefits in our community.

The produced benefits in our community allow it to be little. This is because it is technically hard to have any incremental improvements. One of examples showing little benefit but novel knowledge is tight security proof for some special cryptography schemes where the loss was proven to be bounded with the query number q from the adversary in a security model for all known constructions [52]. A scheme that can bypass impossibility and has security proof with any loss less than q must contribute significant knowledge.

The produced benefits in our community allow it to be blurred and conditionally true only. The first example showing blurred benefits but novel knowledge is the research of identity-based encryption [131]. Assuming that the first scheme was constructed from cyclic groups with pairing [28] and the second scheme was constructed from cyclic groups without pairing [57]. Even though the second scheme is less practical and no more secure than the first scheme, the second scheme still brings benefits and the benefits will become huge if all constructions of pairings were found to be insecure in the future. The second example is the construction of digital signatures from a one-way function. Assuming that the first scheme was the RSA signature scheme [127] and the second scheme was digital signatures from any one-way function [115]. The second scheme uses a weaker foundation in constructing the signature scheme because the RSA scheme is based on a specific one-way trapdoor permutation. The benefits will become huge if it is hard to find a secure one-way trapdoor permutation to implement RSA. We note that this is happening in our academic community on the assumption that quantum computers are possible in the future. The second scheme has contributed a lot because a hash-based signature scheme instantiated from the second scheme can currently resist quantum attacks [19].

The produced benefits in our community allow the existence of tradeoffs. In our community, the tradeoff is conditionally acceptable when the second scheme brings more benefits in terms of some factors (among quantity improvement and desired properties) at the price of losing benefits of some other factors. This is because the novelty of knowledge cannot be directly reflected by the number of benefits or the number of factors. Taking digital signatures as an example, when digital signatures are applied to provide integrity on data sent from clients to a remote server where clients serve as signers and the server is the verifier, the bottleneck efficiency is dominated by how fast the server can respond to requests from thousands of clients when clients are queuing for receiving services [117]. It is acceptable even if the second signature scheme has to sacrifice the signing efficiency in order to significantly improve the verification efficiency in this application scenario. We also found that many proposed schemes in the literature have different desired properties. Assuming that X,Y, Z denote some desired properties, the first scheme has properties X and Y, while the second scheme has Y and Z properties without X. The tradeoff exists but the knowledge is novel because it is unknown how to use existing knowledge or it is difficult to use current knowledge to construct a scheme with Y and Z properties at the same time. This kind of tradeoff is quite often when considering the factors of security proof. For example, the first scheme is proven secure without random oracles under standard hardness assumption, while the second scheme is proven secure under standard hardness assumption and against quantum attacks in the random oracle model.

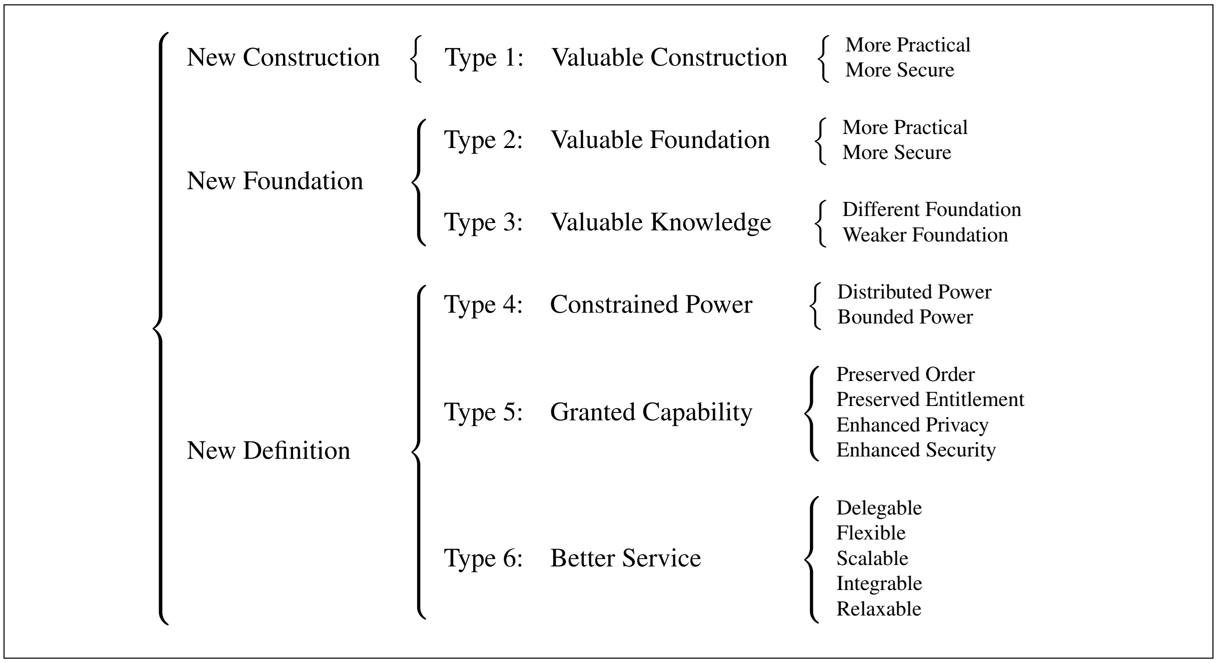
When the produced benefits are not obvious, presenting a well-crafted narrative can effectively communicate the novelty of the knowledge used in constructing the second scheme. The literature suggests using specific scenarios and technique difficulties to make a compelling argument. How to argue novelty from little/blurred/tradeoff benefits is outside the scope of this paper.

4.3 Ways for Benefits and Types of Benefits

Recalling that the contributions made by Alice comprise of (1) definition for the cryptography notion $\mathcal{N}$, (2) finding the applicable foundation F for scheme construction, and (3) the construction of the first provably secure scheme S. In this section, we introduce how Bob can explore benefits in detail.

The literature shows 3 ways that Bob can conduct the research based on Alice's contributions. These 3 ways can be further expanded into 6 types of benefits and 17 benefit areas in total.

3 Ways → *6 Types of Benefits* → *17 Benefit Areas*



New Construction	Type 1:	Valuable Construction	More Practical More Secure
New Foundation	Type 2:	Valuable Foundation	More Practical More Secure
	Type 3:	Valuable Knowledge	Different Foundation Weaker Foundation
New Definition	Type 4:	Constrained Power	Distributed Power Bounded Power
	Type 5:	Granted Capability	Preserved Order Preserved Entitlement Enhanced Privacy Enhanced Security
	Type 6:	Better Service	Delegable Flexible Scalable Integrable Relaxable

The three ways are "New Construction", "New Foundation", and "New Definition"[5]. They are explained as follows.

- **New Construction.** Without changing the definition and the foundation given in the first scheme, the second scheme contributes to a new construction. For example, assuming that the BLS signature scheme [32] constructed from bilinear pairing was treated as the first scheme. The second scheme [140] by Waters is a new construction of a digital signature scheme from the same definition and foundation.

- **New Foundation.** Without changing the definition in the first scheme, the second scheme aims to use a foundation different from that being used to construct the first scheme. For example, assuming that the RSA signature scheme [127] from integer ring was treated as the first signature scheme. The ElGamal signature scheme [67] from cyclic group can be seen as the second scheme from a new foundation.

- **New Definition.** In this way, the second scheme is constructed based on new definitions. The new definition can refer to a new algorithm definition or a new security definition. For example, assuming that the RSA signature scheme was treated as the first signature scheme [127]. The batch RSA scheme [61] can be seen as the second scheme aiming to improve signing efficiency by introducing a new algorithm that can sign multiple signatures at the same time.

Although all developed benefits ultimately aim to be more practical or secure in their applications, the benefits that can be derived from each of these ways are not identical. In order to clarify the specific benefits of each way, we expand the benefits of practicality or security into the following 6 types (*Valuable Construction, Valuable Foundation, Valuable Knowledge, Constrained Power, Granted Capability, and Better Service*), where each type can be further categorized into several benefit areas.

[5] We list in this order because it is relatively easier to make contributions via new construction especially for beginners.

The Type 1 benefits (Valuable Construction) are to show the value of the new construction via two areas: more practical and more secure. That is, the second scheme is to claim as more practical or more secure than the first scheme according to those evaluation factors introduced in Section 3.2.

The Type 2 benefits (Valuable Foundation) are to show the value of adopting the new foundation via two areas: more practical and more secure the same as type 1. But the additional benefits are due to the successful application of the new foundation. That is, the construction of the second scheme could probably borrow the main idea in the first scheme.

The Type 3 benefits (Valuable Knowledge) are to show the knowledge of applying the new foundation via two areas: different foundation and weak foundation. These two areas are eventually linked to show the value of the new foundation, but the corresponding benefits are little, blurred, or having tradeoffs.

The benefits from a new definition have significant differences when compared to those from a new construction and from a new foundation. The new definition is motivated by the application scenario which is becoming complex or needs some enriched security services. To clarify the remaining three types of benefits to be introduced, we have abstracted all application scenarios into the scene that *Person(s) A needs to do some secret computation and the result is known or received by the person(s) B*.

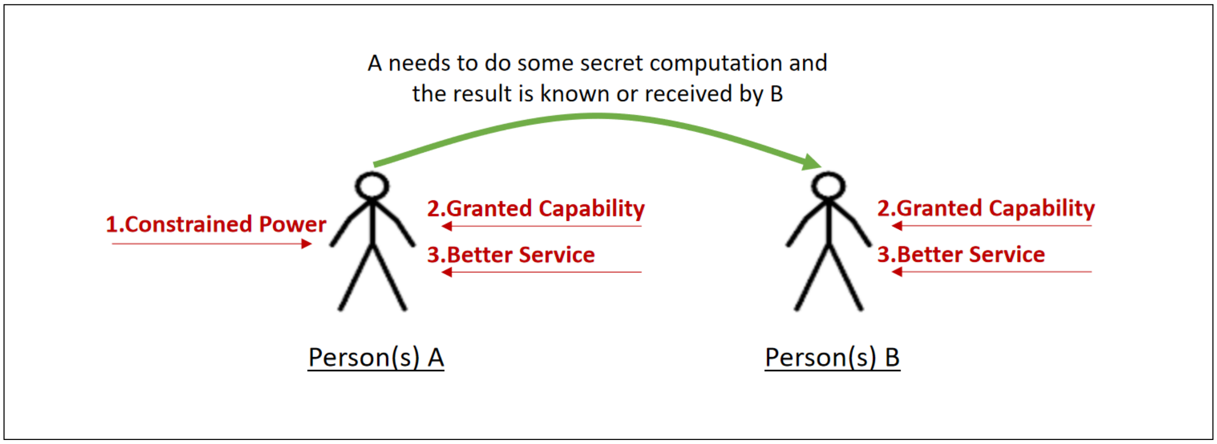


Figure 3: Abstracted Application Scenario for New Definition.

Assuming that the first scheme has met the need for this application scenario. The second scheme aims to redefine the proposed notion or its security definition in the first scheme to (1) apply constrained power on A, (2) create extra capability for A and B, or (3) provide better service for A and B (Figure 3). The details are explained as follows.

The Type 4 benefits (Constrained Power) are to show the importance of the new definition via two areas: powers are distributed and powers are bounded. The benefits arise especially when a single person A cannot be fully trusted in computation in this abstracted application scenario, and the new definition is looking for secret computation by more than one person.

The Type 5 benefits (Granted Capability) are to show the importance of the new definition via four areas where A or B is granted with additional or more powerful capabilities:

- *Preserved Order:* The order is preserved for the person(s) A who launches secret computing. A is able to do something additional to protect his/her secret computing especially when B illegally uses these computing results.
- *Preserved Entitlement:* The entitlement is preserved for the person(s) B who receives computing

results. B is able to do something additional to protect his/her rights when B needs to use A's computing results.

- *Enhanced Privacy:* The privacy is enhanced where A(B) is able to preserve more privacy in this application scenario. Here, privacy refers to information that was not considered or defined in the previous security model and wants to be additionally covered.
- *Enhanced Security:* The security is enhanced because A(B) will be still protected even if adversary B(A) gains more information. Here security refers to enhancing the security that was considered but not enough.

The Type 6 benefits (Better Service) are to show the effectiveness of the new definition via five areas: Delegable, Flexible, Scalable, Integrable, and Relaxable. With this type of benefit, users can further enjoy more practical services that are not provided or cannot be effectively provided in the first scheme. These services are not related to security but practicality of proposed schemes.

4.4 Research Strategies for Benefits

We have classified 6 types of benefits into 17 benefit areas that were categorized according to the characteristic of benefits. In this section, we show how to explore benefits in each benefit area using different research strategies.

In this paper, a research strategy is treated as a plan of action towards making changes (on some objects) from negative results to positive results, such that quantity improvement or desired property appears in the corresponding benefit area. In general, if "*adj*" is an adjective word used to represent a positive result, in this paper, a research strategy is to explore changes:

From $\underline{\textit{less-adj}}$ *(in the first scheme) To* $\underline{\textit{more-adj}}$ *(in the second scheme).*

For example, from slow to fast, from long to short, and from loose to tight which are about the efficiency and security of proposed schemes. In each benefit area, our community has proposed some research strategies to produce corresponding benefits. In the next section, we will introduce 40 research strategies found in the above 17 benefit areas that are original from 3 ways for benefits. That is:

3 Ways for Benefits → *6 Types of Benefits* → *17 Benefit Areas* → *40 Research Strategies.*

In each research strategy, we emphasize that more than one kind of specific benefits can be considered and produced. This is because we can consider multiple and different objects in each research strategy. Here the objects refer to

- input/output elements of algorithms like secret key and ciphertext. By applying the "from long to short" research strategy, we can consider how to reduce the size of secret keys or the size of ciphertexts.
- different algorithms defined inside a cryptography notion like signing algorithm and verification algorithm. By applying the "from slow to fast" research strategy, we can consider how to improve signing efficiency or verification efficiency.
- involved entities in a cryptography notion like signer and verifier. The literature shows that we can do more to benefit entities, such as provide more security protections for signers or verifiers.

We will give concrete examples from the literature when introducing each research strategy.

4.5 Roadmap of 40 Research Strategies

In the following sections, we are going to introduce 40 research strategies proposed in the literature in 17 benefit areas, assuming that Alice has introduced a new notion called *group signatures* and proposed the first scheme.

In a group signature scheme, a group manager can allow different users to join a group and sign messages on behalf of the group. When a group signature is generated, the verification shows that it was generated by one of the group signers but the signer identity is anonymous. In case a dispute of generating a group signature happens, the group manager has a special secret key and can open the group signature to know the signer identity. To help clear what a group signature scheme is, we provide the artificial definition in the first scheme based on [16].

Definition 1 (Group Signatures) *A group signature scheme is composed of the following four probabilistic polynomial time algorithms.*

- **GSetup**(λ, n)$\rightarrow$ $(gpk, gsk_1, gsk_2, \cdots, gsk_n, gmsk)$*: Taking as input a security parameter* λ *and the group size* n*, the algorithm returns the tuple* $(gpk, gsk_1, gsk_2, \cdots, gsk_n, gmsk)$*, where* gpk *is the group public key,* gsk_i *is the group signing key for the group signer* $i \in [1, n]$*, and* $gmsk$ *is the group manager's secret key.*

- **GSign**(gsk_i, m)$\rightarrow$ σ_m*: Taking as input a group signing key* gsk_i *and a message* m*, the algorithm returns a signature* σ_m *under* gsk_i $(i \in [1, n])$*.*

- **GVerify**(m, σ_m, gpk)$\rightarrow$ $0/1$*: Taking as input a message* m*, a signature for* m*, and the group public key gpk, the algorithm returns either 1 (valid) or 0 (invalid).*

- **GOpen**($m, \sigma_m, gmsk$)$\rightarrow$ $i/\perp$*: Taking as input a signature* σ_m *for* m*, and the group manager secret key gmsk, the algorithm returns an identity* $i \in [1, n]$ *or the symbol* $\perp$ *to indicate failure.*

The correctness requires that for all $(gpk, gsk_1, gsk_2, \cdots, gsk_n, gmsk)$*, all messages* m*, and all* $i \in [1, n]$*, if* $\sigma_m \leftarrow \mathsf{GSign}(gsk_i, m)$*, then we have the correctness of* $\mathsf{GVerify}(m, \sigma_m, gpk) \rightarrow 1$ *and* $\mathsf{GOpen}(m, \sigma_m, gmsk) \rightarrow i$*.*

The roadmap for introducing these research strategies is described as follows.

We start with a general description for introducing each research strategy. The description highlights the general gap or problem that exists in the first scheme, and what the scheme aims to achieve. The additional benefits after achieving the aim will fall into the corresponding benefit area.

Next, we use group signatures as the artificial example to further explain each research strategy. Within each research strategy, the gap or problem in the first group signature scheme proposed by Alice will be identified, and what kinds of results the second group signature scheme has achieved will be given. We believe that all artificial examples using group signatures will help readers understand the research strategies in a systemic way. It is worth noting that most artificial examples are based on concrete results in the literature.

Eventually we show how each research strategy was applied to different cryptography notions using three concrete examples in the literature. These three examples include any potential cryptography notion such as signatures, encryption, or protocols. For each concrete example, we will describe the gap or problem that exists in all selected cryptography schemes before the second scheme and how the second scheme fills the gap. We also emphasize that the introduced second scheme could be just part of contributions in that paper we have cited. A proposed new cryptography notion in the literature could be also based on multiple research strategies.

5 Strategies for Benefits from New Construction

Suppose that Bob wants to propose the second scheme with the same definition and foundation as the first scheme for the cryptography notion $\mathcal{N}$ and contribute new construction. In this section, we introduce how Bob can conduct in his research via new construction. We introduce research strategies in the types of benefits via valuable construction (Type 1).

5.1 Type 1: Valuable Construction

5.1.1 Area: More Practical

Strategy 1 (More Practical: From Slow to Fast) *In the first scheme, an algorithm is very slow in computing its output. The increased waiting time has negatively impacted the users' benefits when they run this algorithm. The second scheme aims to have a new construction for improving the computational efficiency of this algorithm.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on cyclic groups and generating each group signature requires 100 exponentiations. The second scheme has significantly improved the signing efficiency with 5 exponentiations only.

The concrete examples in the literature that have applied this research strategy include:

- Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps (Asiacrypt 2005) [14]. In all identity-based signcryption schemes before this work, there have been many proposed schemes with efficient signing/encrypting and verifying/decrypting from bilinear pairing. The scheme proposed in this work turns out to be more efficient than all other proposed schemes so far.

- Revocable Group Signature Schemes with Constant Costs for Signing and Verifying (PKC 2009) [118]. In all revocable group signature schemes before this work, signing and/or verification have linear time complexity in the number of group size or revoked number, or constant time complexity by requiring to update signing keys after each new joining/revocation. The scheme proposed in this work has $O(1)$ time complexity of signing and verification, where no updates of signing keys are required.

- Bootstrapping Fully Homomorphic Encryption over the Integers in Less than One Second (PKC 2021) [125]. In all fully homomorphic encryption (FHE) schemes before this work, bootstrapping FHE over lattices is much more efficient than that over the integers. The FHE scheme over the integers proposed in this work has significantly improved the efficiency where bootstrapping over the integers can be less than one second in a common personal computer.

Strategy 2 (More Practical: From Long to Short) *In the first scheme, the output returned from an algorithm is very long. The size is related to storage cost or communication cost, while a long but unnecessary output has negatively impacted the users' benefits. The second scheme aims to have a new construction for reducing the size of output of this algorithm.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on cyclic groups and each group signature is composed of 20 group elements. The second scheme has significantly reduced the size of the group signature and each is composed of 5 group elements only.

The concrete examples in the literature that have applied this research strategy include:

- Constant Size Ciphertexts in Threshold Attribute-Based Encryption (PKC 2010) [89]. In all attribute-based encryption schemes before this work, schemes that admit reasonably access policies have to produce ciphertexts whose size is at least linear in the number of attributes involved in the policy. The scheme proposed in this work can support threshold policy and the produced ciphertext is constant-size having 3 group elements only.

- Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups (Crypto 2011) [2]. In all structure-preserving signature schemes before this work, each signature is composed of at least 7 group elements. The scheme proposed in this work has reduced the signature size into 3 group elements which are proven to be an optimal size.

- Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors (Eurocrypt 2016) [105]. In all ring signature schemes before this work, there exist lattice-based schemes but their signatures have linear size in the number of ring numbers. The scheme proposed in this work has a logarithmic signature size in the cardinality of the ring, which is based on an efficient lattice-based accumulator that enables short zero-knowledge arguments of membership.

Strategy 3 (More Practical: From Costly to Cheap) *In the first scheme, running an algorithm needs to consume a significant amount of interaction cost among related entities. The cost has negatively impacted the users' benefits. The second scheme aims to have a new construction for reducing the interaction cost when running this algorithm.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the opening algorithm can be modified into an opening protocol, where the signature owner can know the signer's identity with the help of the group manager who knows nothing about the group signature and the identity. However, the opening protocol requires the signature owner and the group manager to interact for 5 moves. The second scheme only requires 3 moves between the owner and the group manager when they run the opening protocol.

The concrete examples in the literature that have applied this research strategy include:

- 3-Move Undeniable Signature Scheme (Eurocrypt 2005) [102]. In all undeniable signature schemes before this work, the best zero-knowledge confirmation protocol requires 4-move interactions. The scheme proposed in this work can complete the confirmation protocol and the disavowal protocol with 3 moves, and the protocols are secure against active and concurrent attacks.

- Identity-Based Aggregate and Multi-Signature Schemes Based on RSA (PKC 2010) [13]. In all identity-based multi-signature schemes before this work, there exist schemes from bilinear pairing and RSA, but the RSA-based schemes require three rounds of interactions. The RSA-based scheme proposed in this work has reduced the round complexity into two with almost the same efficiency.

- Sharing Transformation and Dishonest Majority MPC with Packed Secret Sharing (Crypto 2022) [82]. In all multi-party computation (MPC) schemes before this work, the most efficient scheme in the dishonest majority setting requires $O(n)$ communication complexity per multiplication gate across all n parties. The scheme proposed in this work has reduced the complexity from linear to sublinear using a new technique called sharing transformation.

5.1.2 Area: More Secure

Strategy 4 (More Secure: From Strong-Assumption to Weak-Assumption) *In the first scheme, the security is based on a strong hardness assumption. The second scheme aims to have a new construction and its security is based on a weak hardness assumption (under the same foundation).*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the security is based on a q-type assumption. However, this assumption is stronger than standard assumptions like Computational Diffie-Hellman (CDH) assumption, because breaking the q-type assumption is much easier than those standard assumptions. The second scheme is proven secure under the CDH assumption.

The concrete examples in the literature that have applied this research strategy include:

- Strongly Unforgeable Signatures Based on Computational Diffie-Hellman (PKC 2006) [33]. In all digital signature schemes before this work, there exist strongly unforgeable schemes in the standard model where giving a signature on message m the adversary cannot produce a new signature on this message, but the schemes are based on strong hardness assumptions. The scheme proposed in this work is strongly unforgeable and based on the standard Computational Diffie-Hellman problem in the standard model.

- Realizing Hash-and-Sign Signatures under Standard Assumptions (Eurocrypt 2009) [92]. In all digital signature schemes before this work, there exist schemes secure in the standard model but they depend on strong hardness assumptions or they are not practical enough. The scheme proposed in this work is practical with constant-size public keys and proven secure under standard hardness assumption in the standard model.

- Short Signatures From Weaker Assumptions (Asiacrypt 2011) [90]. In all digital signature schemes before this work, there exist schemes with very short signatures (less than 230 bits for 80-bit security) in the standard model but they depend on the strong RSA assumption or strong q-Diffie-Hellman assumption. The scheme proposed in this work has the same short signature size but can be proven secure under the RSA assumption or q-Diffie-Hellman assumption in the standard model, which is based on a new construction of a programmable hash function.

Strategy 5 (More Secure: From Loose to Tight) *In the first scheme, the attack on the scheme from the adversary can be reduced to solving a hard problem but there is a significant amount of reduction loss. The second scheme aims to have a new construction with no huge reduction loss in its security reduction.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the security proof is based on a q-type assumption with a reduction loss linear in the number of signature queries. That is, if an adversary can break this signature scheme with advantage ϵ, the hardness assumption will be broken with advantage $\frac{\epsilon}{q}$ at most. The second scheme is proven secure under the same hardness assumption but the security is tightly reduced to this assumption with a constant reduction loss equal to 2.

The concrete examples in the literature that have applied this research strategy include:

- A Signature Scheme as Secure as the Diffie-Hellman Problem (Eurocrypt 2003) [74]. In all digital signature schemes before this work, there exist efficient schemes based on RSA or discrete log, but all discrete-log based schemes have security loss linear in the number of hash queries. The discrete-log based scheme proposed in this work has a tight reduction under the Computational Diffie-Hellman assumption (in the random oracle model).

- Fully, (Almost) Tightly Secure IBE and Dual System Groups (Crypto 2013) [49]. In all identity-based encryption schemes before this work, there exist schemes with provable security based on standard hard problems in the standard model, but their security reductions are loose. The scheme proposed in this work is tightly reduced to the DLIN standard assumption in the standard model with almost tight reductions.

- Optimal Security Reductions for Unique Signatures: Bypassing Impossibilities with A Counterexample (Crypto 2017) [84]. In all unique signature schemes before this work, there exist schemes with provable security in the EUF-CMA security model based on standard hardness assumptions, but all reductions are loose and at least linear in the number of signature queries. The scheme proposed in this work has a tight reduction in the standard security model based on the Computational Diffie-Hellman problem.

Strategy 6 (More Secure: From Relaxed to Strict) *In the first scheme, the proof model in the security proof is relaxed from the standard proof model. The second scheme aims to have a new construction with provable security using a more strict proof model.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the security proof is based on a q-type assumption in the random oracle model. However, it has been shown that security proof in this model could be insecure in the real world. The second scheme is proven secure without the use of random oracles.

The concrete examples in the literature that have applied this research strategy include:

- Secure Hash-and-Sign Signatures Without the Random Oracle (Eurocrypt 1999) [70]. In all digital signature schemes before this work, there exist practical schemes provable secure in the standard security model (EUF-CMA) but all their proofs must use random oracles. The practical scheme proposed in this work can be proven secure in the same security model without random oracles.

- Universally Composable Two-Party and Multi-Party Secure Computation (STOC 2002) [43]. In all multi-party computation schemes before this work, these exist secure schemes regardless of the number of corrupted participants, but they are secure in the stand-alone computation model only. The scheme proposed in this work is secure in the universally composable model even if a majority of the participants are corrupted.

- Efficient Identity-Based Encryption Without Random Oracles (Eurocrypt 2005) [140]. In all identity-based encryption schemes before this work, there exist schemes with provable security in the fully secure model (IND-ID-CCA), but all their proofs must use random oracles. The scheme proposed in this work can be proven secure in the same security model without random oracles.

6 Strategies for Benefits from New Foundation

Suppose that Bob wants to propose the second scheme with the same definition as the first scheme for the cryptography notion $\mathcal{N}$ and contribute new foundation. In this section, we introduce how Bob can conduct in his research via new foundation. We introduce research strategies in the types of benefits including valuable foundation (Type 2) and valuable knowledge (Type 3).

6.1 Type 2: Valuable Foundation

6.1.1 Area: More Practical

Strategy 7 (More Practical: From Less to More) *In the first scheme, the algorithms are not efficient enough, and the inefficient construction has negatively impacted the users' benefits. The second scheme aims to use a new foundation to obtain a more efficient construction.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on cyclic groups and each group signature is composed of 20 group elements. The second scheme is constructed from bilinear pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ and each group signature has two group elements in $\mathbb{G}$.

The concrete examples in the literature that have applied this research strategy include:

- Short Signatures from the Weil Pairing (Asiacrypt 2001) [32]. In all digital signature schemes before this work, there exist very practical schemes with signature size as short as 320 bits for 80-bit security. The scheme proposed in this work for the first time using bilinear pairing has reduced signature size to about 160 bits, which is the shortest one among all schemes.

- Short Group Signatures (Crypto 2004) [26]. In all group signature schemes before this work, there exist practical group signature schemes constructed from RSA. The scheme proposed in this work for the first time using bilinear pairing turns out to have shorter group signatures (about 200 bytes).

- Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys (Crypto 2005) [30]. In all broadcast encryption schemes before this work, there exist practical schemes but the ciphertexts or the secret keys cannot be constant-size and depend on the number of receivers. The scheme proposed in this work for the first time using bilinear pairing allows encryption for any subject of receivers and has constant-size ciphertexts and secret keys.

6.1.2 Area: More Secure

Strategy 8 (More Secure: From Weak to Strong) *In the first scheme, the construction is from foundation A, but the foundation A has been shown some weaknesses in security. The second scheme aims to be constructed from a new foundation B with improved security because the new foundation does not have these weaknesses.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on cyclic groups and the Discrete Log problem over cyclic groups serves as the most fundamental hard problem. However, the Discrete Log problem is easy in front of quantum computers. The second scheme is constructed from lattices and it is still unknown how to use quantum computers to solve hard problems defined over lattice.

The concrete examples in the literature that have applied this research strategy include:

- Trapdoors for Hard Lattices and New Cryptographic Constructions (STOC 2008) [72]. In all identity-based encryption schemes before this work, all schemes are constructed from either bilinear pairing or RSA which are not secure against quantum algorithms. The scheme proposed in this work is constructed using lattices that so far can resist quantum algorithms.

- Lattice-Based Group Signature Scheme with Verifier-Local Revocation (PKC 2014) [103]. Verifier-local revocation (VLR) group signatures are group signatures where only verifiers are required to update the revocation information, but not the group signers. In all group signature schemes before this work, there have been many proposed VLR group signature schemes but they all used bilinear pairings which are not secure against quantum computers. The scheme proposed in this work is constructed using lattices that so far can resist quantum attacks.

- Programmable Hash Functions from Lattices: Short Signatures and IBEs with Small Key Sizes (Crypto 2016) [142]. Programmable hash function (PHF) is a powerful tool for constructing cryptography schemes with short outputs in the standard model. In all PHF schemes before this work, all schemes were constructed over groups where the discrete log problem is not secure against quantum computers. The scheme proposed in this work is constructed using lattices that so far can resist quantum attacks.

6.2 Type 3: Valuable Knowledge

6.2.1 Area: Different Foundation

Strategy 9 (Different Foundation: From Plan-A to Plan-B) *In the first scheme, the construction is from foundation A. The second scheme aims to be constructed from a new foundation B, which serves as plan B in case that the foundation A is found to be insecure for scheme construction.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on cyclic groups. It is still unknown how to construct a group signature scheme based on integer ring like the RSA scheme. The second scheme is the first construction from integer ring under the strong RSA hardness assumption. In comparison with the first scheme, the second scheme is still secure even if the Discrete Log problem over cyclic groups is found to be easy.

The concrete examples in the literature that have applied this research strategy include:

- Public-Key Cryptosystems from Lattice Reduction Problems (Crypto 1997) [75]. The security of public key cryptography is based on the existence of computational intractability of problems. In all public-key encryption and signature schemes before this work, they were mostly constructed with hard problems defined over integer rings or cyclic groups. The scheme proposed in this work is based on the difficulty of new lattice-reduction problems, providing a possible alternative to existing candidates.

- Publicly Verifiable Proofs from Blockchains (PKC 2019) [130]. A proof system is publicly verifiable if anyone given the transcript of a proof can be convinced that the corresponding theorem is true. In all schemes before this work, they were constructed and secure based on trust assumptions, heuristic assumptions, specific number-theoretic assumptions, or obfuscation assumptions. The scheme proposed in this work is secure and based on the existence of a very generic blockchain, which is different from existing assumptions.

- Candidate Witness Encryption from Lattice Techniques (Crypto 2022) [137]. Witness encryption (WE) is an encryption scheme where messages are encrypted with respect to an instance of an NP relation, such that decryption needs a valid witness for that instance. In all WE schemes before this work, all schemes from standard assumptions either rely on iO or use more powerful techniques. The scheme proposed in this work relies on a different foundation because this foundation will be trivially broken when one tries to convert it to iO.

6.2.2 Area: Weaker Foundation

Strategy 10 (Weaker Foundation: From Strong to Weak) *In the first scheme, the construction is from foundation A. The second scheme aims to be constructed from a new but weaker foundation B, where B cannot construct A but A implies B. For example, A is a one-way trapoor function and B is a one-way function.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on a very special cyclic group (as the foundation) where the group generator who knows a trapdoor can solve the Discrete Log problem for most problem instances. The second scheme is also constructed from cyclic groups but can be instantiated with any normal group. In comparison with the first scheme, the second scheme has weakened the use of foundation in the scheme construction. It is easier to obtain a secure normal group. If the DL problem over normal groups is easy, then this problem over that special group must be easy too. But if the DL problem over that special group is easy, this problem over normal groups could still be hard.

The concrete examples in the literature that have applied this research strategy include:

- An Efficient Signature Scheme from Bilinear Pairings and Its Applications (PKC 2004) [141]. In all digital signature schemes before this work, there exist schemes with signatures as short as 160 bits for 80-bit security, but the construction must employ a specific and inefficient cryptographic hash function mapping all inputs into elements of a pairing group. The scheme proposed in this work has the same signature size but is constructed using normal hash functions.

- Identity-Based Encryption from the Diffie-Hellman Assumption (Crypto 2017) [57]. In all identity-based encryption schemes before this work, there exist schemes based on the Diffie-Hellman hard problems defined over cyclic groups, but all groups must be equipped with a bilinear map. The scheme proposed in this work is fully secure based on Computational Diffie-Hellman problems but can be constructed from any group even if there is no bilinear map.

- Chosen Ciphertext Security from Injective Trapdoor Functions (Crypto 2020) [91]. In all public-key encryption schemes before this work, there exist generic constructions of IND-CCA secure schemes from IND-CPA secure schemes, lossy trapdoor functions, or doubly enhanced trapdoor permutation. The scheme proposed in this work is a generic construction for IND-CCA security from injective trapdoor functions, which are weaker than all existing foundations.

7 Strategies for Benefits from New Definition

Suppose that Bob wants to propose the second scheme with the same application scenario as the first scheme and contribute new definition. In this section, we introduce how Bob can conduct in his research via new definition. We introduce research strategies in the types of benefits including constrained power (Type 4), granted capability (Type 5), and better service (Type 6).

7.1 Type 4: Constrained Power

7.1.1 Area: Distributed Power

Strategy 11 (Distributed Power: From Centralized to Decentralized) *In the first scheme, the secret computation is done by a single user A. If A is untrustworthy, he or she could pose a threat to B's benefits. The second scheme aims to restrict this secret computation to only be available by a* *number of group users from A.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, there is only one group manager who can use $gmsk$ to open all group signatures to know the identities of signers. The second scheme has introduced a threshold mechanism for group management. There is a group of managers, and each group manager will receive its share of the group manager's secret key $gmsk_i$. Any subset of group managers cannot open a group signature together unless the capacity is not less than a threshold number.

The concrete examples in the literature that have applied this research strategy include:

- Threshold Cryptosystems (Crypto 1989) [54]. In all public key encryption schemes before this work, ciphertexts generated for pk will be decrypted using sk that is owned by one single person. The scheme proposed in this work has introduced a mechanism where sk is shared by a group of n persons and any t out of n persons can work together in a non-interactive way to decrypt ciphertexts.

- Efficient and Provably Secure Trapdoor-free Group Signature Schemes from Bilinear Pairings (Asiacrypt 2004) [120]. In all group signature schemes before this work, the group manager will perform the management of letting users join the group and opening groups signatures using the same group master secret key. The scheme proposed in this work has introduced a mechanism where these two managements are split into two managers with different group master secret keys.

- Decentralized Attribute-Based Signatures (PKC 2013) [122]. Attribute-based signatures (ABS) are specific digital signatures where the private key of an attribute set S can sign message m along with a policy P if and only if S fulfils the policy, while verifiers learn nothing except that signers have attributes fulfilling the policy. In all ABS schemes before this work, all private keys are generated by a central authority. The scheme proposed in this work has introduced a decentralized mechanism where any person can perform as the authority and issue private keys of attributes for signers.

7.1.2 Area: Bounded Power

Strategy 12 (Bounded Power: From Unbounded to Boundable) *In the first scheme, the user A has a secret key and can do any secret computations without restrictions. If A is untrustworthy, he or she could pose a threat to B's benefits because A can compute whatever he/she likes. The second scheme aims to bind A in computations with a secure mechanism.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, each group signing key gsk_i ($i \in [1, n]$) is used to generate group signatures and the group manager secret key $gmsk$ is used to open group signatures. However, $gmsk$ can also be used to generate group signatures on behalf of any gsk_i and frame the group signers. The second scheme has introduced a protected mechanism such that the group manager will not like to do this. More precisely, if the group manager uses $gmsk$ to generate a group signature σ_m for message m on behalf of gsk_i, then the group signer can use gsk_i to extract the group manager secret key $gmsk$ from the signature σ_m for m.

The concrete examples in the literature that have applied this research strategy include:

- k-Times Anonymous Authentication (Extended Abstract) (Asiacrypt 2004) [135]. In all authentication schemes before this work, there exist schemes that can keep the identities of users anonymous in all authentications. The scheme proposed in this work introduces a mechanism where authentication is anonymous but the identity will be traceable after being authenticated beyond a pre-fixed number k.

- Traceable and Retrievable Identity-Based Encryption (ACNS 2008) [10]. In all identity-based encryption schemes before this work, there exist specific schemes with an accountable mechanism which can judge whether user or the private key generator (PKG) releases a private key to the public. The scheme proposed in this work has introduced a retrievable mechanism to bind the PKG, where the master secret key owned by the PKG will be extractable if it is the PKG who leaks the private keys of users to the public.

- Double-Authentication-Preventing Signatures (ESORICS 2014) [126]. In all digital signature schemes before this work, the owner of a secret key can sign any messages without limitation. The scheme proposed in this work has introduced a mechanism where signers cannot use their keys to sign on two messages with the same subject; otherwise, their secret keys can be extracted using the corresponding two signatures.

7.2 Type 5: Granted Capability

7.2.1 Area: Preserved Order

Strategy 13 (Preserved Order: From Unknown to Aware) *In the first scheme, A has done some secret computations for B to continue some applications. But A does not know which computation from A will be used by B. The second scheme aims to propose a mechanism to allow A to know which computation is being used by B.*

The artificial example applying this research strategy is as follows. In the first group signature, once a group signature for m is generated by a group signer, any verifier who receives the group signature can verify it but the group signers do not know who is verifying this signature. The second scheme has introduced a verification-restricted mechanism. It works as follows: when a verifier wants to verify a signature σ_m, the verifier must send his/her identity ID and message m to any group signer, and the group signer can generate a token $T_{ID,m}$ to this verifier. The group signature cannot be verified unless the verifier has $T_{ID,m}$ and a private key related to his/her identity ID generated by a trusted third party who serves as the private key generator.

The concrete examples in the literature that have applied this research strategy include:

- Undeniable Signatures (Crypto 1989) [48]. In all digital signature schemes before this work, once a signature is generated, any verifier can verify it using the signer's signing key and the signer therefore does not know who has verified this signature. The second scheme proposed in this work has introduced a mechanism where signature verification needs the help from signers and signers cannot deny if signatures were indeed generated by them.

- Provably Secure Partially Blind Signatures (Crypto 2000) [4]. In all blind signature schemes before this work, the signer does not know what messages he/she is signing, and only the signature receiver knows. The scheme proposed in this work has introduced a mechanism where the signer knows part of the messages to be signed which cannot be cheated by the signature receiver.

- Break-glass Encryption (PKC 2019) [129]. In all proposed encryption schemes before this work, all ciphertexts can only be decrypted with the help of corresponding secret keys. The scheme proposed in this work has introduced a mechanism where ciphertexts stored in a third party can be decrypted by the third party without keys but exactly once, and this decryption is detectable and noticeable by the key owner.

Strategy 14 (Preserved Order: From Unknown to Traceable) *In the first scheme, the secret computation done by A will be passed to B (a group of users) to continue some secret computation. But one of the users in B has broken the rule and done something not allowed by A. The second scheme aims to propose a mechanism to allow A to trace which user in B should be blamed.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the verification of a group signature can be modified into an interactive proof where the owner of the group signature can prove to a verifier that he/she has a group signature for m without leaking the group signature. Unfortunately, the opening algorithm cannot work on this interactive proof to open the real identity of the signer. That is, the real signer is unknown and cannot be traced from the interactive proof. The second scheme has proposed a mechanism where the group manager can not only trace the identity from any given group signature but also from any well-designed interactive proofs without knowing the group signature.

The concrete examples in the literature that have applied this research strategy include:

- Tracing Traitors (Crypto 1994) [50]. In all public key encryption schemes before this work, a pair of keys denoted by (pk, sk) is generated where pk is used for encryption and sk can be used by a group of persons for decryption, but it is possible that someone releases sk to the public and we do not know whom. The scheme proposed in this work has introduced a mechanism where a tuple of keys composed of pk and more than one different secret keys sk_i are generated for different receivers, such that all secret keys sk_i can decrypt ciphertexts computed using pk and we can also trace which pirate secret key has been released.

- Group Encryption (Asiacrypt 2007) [100]. In all public key encryption schemes before this work, there exist schemes where the receiver of a ciphertext is anonymous from the view of the ciphertext. The scheme proposed in this work has introduced a mechanism where the sender of ciphertext can hide the identity of the receiver within a group and also prove that this receiver belongs to the group, while the group manager can open and trace who the real receiver is.

- Traceable Secret Sharing and Applications (Crypto 2021) [83]. In all (t, n)-secret sharing schemes before this work, a user can split and share a secret to n parties where any t of shares can recover the secret, but it is unknown to the user which t shares are used to recover the secret. The scheme proposed in this work has introduced a mechanism where the secret is shared in the way that once a party reveals its share, he/she will be traced and caught with valid proof generated by the user.

Strategy 15 (Preserved Order: From Binding to Revocable) *In the first scheme, the secret computation done by A will be passed to B to continue some secret computation. But B has broken the rule and done something not allowed by A. The second scheme aims to propose a mechanism to allow A to revoke the ability of secret computation by B.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, once a group signer is corrupted and has generated group signatures, his/her identity can be traced by the group manager with $gmsk$. A consequence process is to kick this group signer out of the group. However, the only secure solution available for the first scheme is to re-generate the group keys including the group public key which is impractical because it has impacted all verifiers. The second scheme has proposed a secure revocation mechanism where any group signers can be revoked and disabled without the change of other group signers and the group public key.

The concrete examples in the literature that have applied this research strategy include:

- A Public-Key Traitor Tracing Scheme with Revocation Using Dynamic Shares (PKC 2001) [139]. In all public-key based encryption schemes with traitor tracing before this work, there are many proposed schemes that can efficiently trace traitors from black-box decoders, but they did not consider how to revoke traitors' private keys. The proposed scheme in this work has introduced a mechanism where traitor tracing can further revoke traitors without updating any private key of the remaining valid users.

- Efficient Revocation in Group Signatures (PKC 2001) [35]. In all group signature schemes before this work, there have been many proposed schemes with different focus on either efficiency or security but they did not consider how to revoke group members if misusing by some members happened. The scheme proposed in this work has introduced a mechanism where group members can be efficiently revoked and cannot sign in the future without impacting the security of past group signatures.

- Identity-Based Encryption with Efficient Revocation (ACMCCS 2008) [23]. In all identity-based encryption schemes before this work, there exist many schemes but they can only trivially revoke a user's private key by using a time period and have to update n private keys for all remaining n users. The scheme proposed in this work has introduced a mechanism where users can be revoked at any time and the update efficiency is a logarithmic size of n.

Strategy 16 (Preserved Order: From Inequitable to Equitable) *In the first scheme, the secret computation done by A will be passed to B who should do some secret computation for A. But B is able to break rules and create inequitable results. The second scheme aims to propose a secure mechanism to provide equitable computations between A and B.*

The artificial example applying this research strategy is as follows. There are two companies denoted by A and B using group signatures to digitally sign contracts. One day, they made a deal and decided to sign on the same contract denoted by m remotely. Intuitively, either A or B should send its group signature to the other party to complete the exchange of the signed contract. However, neither of them would like to disclose the signed contract first for security reasons. The first scheme cannot be applied to solve the concerns from A and B. The second scheme has introduced a mechanism that allows one party (denoted by A) to sign and send the group signature σ_m to the other party (denoted by B), but the group signature σ_m cannot be verified correctly without the input of group signature for m generated by B. With the help of the second scheme, any verifier must know either nothing or that A and B have both signed the contract.

The concrete examples in the literature that have applied this research strategy include:

- Optimistic Fair Exchange of Digital Signatures (Extended Abstract) (Eurocrypt 1998) [8]. In all digital signature schemes before this work, when two parties would like to sign on messages and then exchange their signatures, one party A has to send his/her signature to another party first but another party B could just run away after receiving A's signature. The scheme proposed in this work has introduced a mechanism where A and B can securely exchange signatures and it needs the help of a trusted third party to complete the exchange if and only if B attempts to cheat.

- Sequential Aggregate Signatures from Trapdoor Permutations (Eurocrypt 2004) [111]. In all aggregate signature schemes before this work, there have been many schemes that can combine n

signatures from n different signers on n different messages into one single signature, but the verification cannot show who signed messages first. The scheme proposed in this work has introduced a mechanism where the set of n signers is ordered when the aggregate signature is generated and the order can be verified by verifiers.

- Generalized Channels from Limited Blockchain Scripts and Adaptor Signatures (Asiacrypt 2021) [12]. In all fair exchange schemes before this work, when party A wants to use his/her signature to exchange a secret value with any party who knows the value, they have to use the help of a trusted third party. The scheme proposed in this work has introduced a mechanism that can be applied in this scenario without any third party with a formal definition and provably secure construction.

Strategy 17 (Preserved Order: From Equivalent to Hierarchical) *In the first scheme, B can do same-power secret computations as A. The second scheme aims to propose a mechanism to allow A to restrict the computing ability of B to a level lower than what A can compute.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, there is only one group manager who can use the $gmsk$ to open all group signatures. The workload is heavy for this group manager in a large scale application scenario. The second scheme has introduced a hierarchical mechanism to manage the use of $gmsk$. There are three properties in this mechanism. First, $gmsk$ can be used to generate a level-1 group manager secret key $gmsk_{[T_1]}$ equipped with a tag name T_1. Second, the key $gmsk_{[T_1]}$ can only be used to open those group signatures for messages including the tag name T_1. Third, generally speaking, the level-i group manager secret key for $[T_1, T_2, \cdots, T_i]$ can generate level-$(i+1)$ group manager secret key for $[T_1, T_2, \cdots, T_i, T_{i+1}]$ that can only open group signatures for messages including the tags $[T_1, T_2, \cdots, T_i, T_{i+1}]$.

The concrete examples in the literature that have applied this research strategy include:

- Toward Hierarchical Identity-Based Encryption (Eurocrypt 2002) [94]. In all identity-based encryption schemes before this work, only the private key generator (PKG) can issue private keys unless sharing the master secret keys with other parties. The scheme proposed in this work has introduced a mechanism that allows a hierarchy of key escrow at several levels, where a level-i private key can issue private keys for level-$(i+1)$.

- Hierarchical Group Signatures (ICALP 2005) [136]. In all group signature schemes before this work, the use of the group master secret key $gmsk$ by the group manager can manage all group members and open all group signatures. The scheme proposed in this work has introduced a mechanism that allows the management of $gmsk$ in a hierarchical way, where a manager can manage a group of sub-managers or group signers, and each sub-manager can only open those signatures generated by group signers under his/her management.

- Hierarchical Predicate Encryption for Inner-Products (Asiacrypt 2009) [121]. In all inner-product encryption schemes before this work, a private key of a vector $\hat{v}$ generated by the PKG can decrypt ciphertexts computed using another vector $\hat{u}$ if and only if $\hat{u} \cdot \hat{v} = 0$. The scheme proposed in this work has introduced a mechanism where the hierarchical structure is applied for inner-product encryption. More precisely, a level-i private key of vectors $(\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_i)$ can generate level-$(i+1)$ private keys for vectors $(\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_i, \hat{u}_{i+1})$, which can be used to decrypt ciphertexts generated using $(\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_i, \hat{v}_{i+1})$ if and only if $\hat{u}_j \cdot \hat{v}_j = 0$ for all $j \in [1, i+1]$.

7.2.2 Area: Preserved Entitlement

Strategy 18 (Preserved Entitlement: From Suspected to Self-Provable) *In the first scheme, a secret computation done by A is received by B and other third parties who believe that it could be done by B. But this computing result has negatively impacted B' benefits. The second scheme aims to propose a mechanism to allow B to prove that this computation was not done by him/her.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, who can open the signer identity for generating a group signature must be the group manager using $gmsk$. In the case that the group manager is offline, the group signer David cannot prove that a group signature σ_m for m was not issued by him. This could decrease David's reputation if he is highly suspected. The second scheme has proposed a mechanism that allows a group signer to prove that he/she is not the generator of a group signature without the help of the group manager.

The concrete examples in the literature that have applied this research strategy include:

- Toward the Fair Anonymous Signatures: Deniable Ring Signatures (CT-RSA 2006) [101]. In all ring signature schemes before this work, given a ring signature on a message m, no one including signers in the ring can know who signed the message and this could let signers in the ring become victims and blamed. The scheme proposed in this work has introduced a mechanism where any signer in the ring can prove that the ring signature on m was or was not generated by him/her.

- Reducing Trust in the PKG in Identity Based Cryptosystems (Crypto 2007) [80]. In all identity-based encryption schemes before this work, the private key generator (PKG) can generate private keys for all users and therefore a user could be blamed because of releasing his/her private key to the public even it was done by the PKG. The scheme proposed in this work has introduced a mechanism where a special key generation protocol is applied and a user can prove that a pirate private key was generated by the PKG.

- Disavowable Public Key Encryption with Non-Interactive Opening (AsiaCCS 2015) [95]. In all public-key encryption schemes before this work, when a sensitive message m is encrypted in a ciphertext CT, the receiver cannot prove that the plaintext m inside CT is different from m^* (being traced) unless releasing the secret key. The scheme proposed in this work has introduced a mechanism where the ciphertext receiver can prove that the plaintext in a received ciphertext is not m^* without disclosing the secret key or the plaintext.

Strategy 19 (Preserved Entitlement: From Faulty to Fault-Tolerant) *In the first scheme, the computation done by A is important for B. But the first scheme did not consider the case that the received computation results have something faulty. The second scheme aims to propose a feasible solution to allow B to use these computing results without being impacted by faults.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the signing algorithm can be modified into a special signing algorithm. In this special algorithm, a subset of the group signers with capacity k can generate a special group signature showing that k signers have generated this group signature. However, if one of group signers in this subset tries to mislead the signing, this special property will be destroyed and a verifier cannot verify the number of signers in this special group signature. The second scheme has introduced a mechanism that a verifier can verify the number of honest signers in this kind of group signature even when malicious participants exist.

The concrete examples in the literature that have applied this research strategy include:

- Identification of Bad Signatures in Batches (PKC 2000) [124]. In all batch verification of signature schemes before this work, they can verify a collection of valid signatures in an efficient way with cost less than the total cost of verifying all of them one by one, but the verification algorithm will not work if some signatures are invalid. The scheme proposed in this work has introduced a mechanism that can efficiently identify invalid signatures from a collection of signatures.

- A Practical and Secure Fault-Tolerant Conference-Key Agreement Protocol (PKC 2000) [138]. In all conference-key agreement schemes before this work, they allow a group of people to generate a secret and common conference key, but they did not consider the issue that a malicious participant might try to mislead other participants and disrupt the establishment of a common conference key. The scheme proposed in this work has introduced a mechanism where a common conference key can be still established by honest participants when malicious participants exist.

- Fault-Tolerant Aggregate Signatures (PKC 2016) [88]. In all aggregate signature schemes before this work, they can aggregate n signatures by different signers on n different messages into a short one, but adding an invalid signature into the aggregate one will destroy the validity of all signatures in the aggregate signature. The scheme proposed in this work has introduced a mechanism where given an aggregate signature, verifiers are able to determine the subset of all messages belonging to the aggregate signature that was signed correctly.

Strategy 20 (Preserved Entitlement: From Unknown to Knowable) *In the first scheme, the computation done by A is important for B. But it is also possible for dishonest A to return a wrong computation result to B. The second scheme aims to propose a mechanism to allow B to verify the computing result from A.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the group manager can open any group signature σ_m for m and know the identity of the group signer. However, for any third party without having the $gmsk$, they cannot verify the opened result by the group manager. The second scheme has introduced a mechanism that the opening algorithm on input $(m, \sigma_m, gmsk)$ will not only return an identity i but also a proof. With the help of this proof, any verifier can verify that this group signature was indeed generated by the identity i.

The concrete examples in the literature that have applied this research strategy include:

- Verifiable Random Functions (STOC 1999) [116]. In all pseudorandom function schemes before this work, one can compute pseudorandom output $y = f_s(x)$ for the input x using a seed s and a pseudorandom function f, but it is unknown for the receiver whether y is valid or not. The scheme proposed in this work has introduced a mechanism where the output y can be publicly verified with a public key.

- Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (ACISP 2004) [110]. In all ring signature schemes before this work, when ring signatures are generated, a verifier cannot know which signers in the ring generated these signatures. The scheme proposed in this work has introduced a mechanism where a verifier can verify and identify whether two ring signatures are from the same signer or not if they are generated with respect to the same ring.

- Verifiable Delay Functions (Crypto 2018) [24]. In all cryptography schemes before this work, one can use a cryptography scheme to force a user to do a specific number of sequential computations even if parallel processing is not available, but a third party cannot efficiently verify the computed result by the user. The scheme proposed in this work has introduced a mechanism where computing results can be efficiently and publicly verified at a cost less than re-doing computation.

Strategy 21 (Preserved Entitlement: From Uncomputable to Conditionally-Computable) *In the first scheme, the secret computations done by A will be used by B. But all parameters in computing results chosen by A cannot be changed and they could negatively impact B's benefits when B uses them directly. The second scheme aims to propose a mechanism to allow B to do some well-defined computations over computing results by A.*

The artificial example applying this research strategy is as follows. A digital signature on message m guarantees that the integrity of message m is protected and the signed message m cannot be modified without having the secret key. In the first group signature scheme, suppose that David has received two group signatures for messages m_1 and m_2, he can only disclose three signed results, namely (m_1), or (m_2), or (m_1, m_2), to a verifier. The second scheme has introduced a homomorphic mechanism to compute over signed messages. On input of two group signatures for m_1 and m_2, David can compute a new group signature for a well-defined homomorphic computing result, denoted by $m_1 \bigotimes m_2$. Most interestingly, given the group signature for the homomorphic message $m_1 \bigotimes m_2$, the group manager can still open the identity of group signers who generated the signatures for m_1 and m_2.

The concrete examples in the literature that have applied this research strategy include:

- Content Extraction Signatures (ICISC 2001) [134]. In all digital signature schemes before this work, after a document denoted by $m = (m_1, m_2, \cdots, m_n)$ has been signed, the signature receiver must display the whole document in order to be verified that this document has been signed by the signer; otherwise, any deletion on the document message (for example, m_1 is deleted) will destroy the authentication validity. The scheme proposed in this work has introduced a non-trivial mechanism where the signature receiver can hide part of document message (for example, m_1 is hidden and the revealed document message is $m' = (*, m_2, m_3, \cdots, m_n)$ only); while the receiver can still keep the signature on the modified message m' valid.

- Fully Homomorphic Encryption Using Ideal Lattices (STOC 2009) [71]. In all public key encryption schemes before this work, when receiving a collection of ciphertexts on plaintexts $(m_1, m_2, \cdots, m_n)$, the receiver cannot compute encryption on $C(m_1, m_2, \cdots, m_n)$ for any circuit C if the secret key is unknown. The scheme proposed in this work has introduced a non-trivial mechanism that allows the ciphertext receiver to do this kind of homomorphic encryption.

- Signatures on Randomizable Ciphertexts (PKC 2011) [22]. In all digital signature schemes before this work, after a ciphertext is treated as a message and signed, the signed ciphertext cannot be modified; otherwise, its signature will become invalid. The scheme proposed in this work has introduced a non-trivial mechanism where the signature receiver can re-randomize the random numbers in the ciphertext while the signature on the ciphertext after re-randomization is still valid.

Strategy 22 (Preserved Entitlement: From Others-Enable to Others-Disable) *In the first scheme, A can do some kinds of computations for B. But one kind of computations by A could have negative impact on B's benefits. The second scheme aims to propose a mechanism such that this kind of computation is only feasible by B.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, each group signing key gsk_i ($i \in [1, n]$) is used to generate group signatures and the group manager secret key $gmsk$ is used to open group signatures. However, $gmsk$ can also be used to generate group signatures on behalf of any gsk_i and frame the group signers. The second scheme has introduced a protected mechanism where $gmsk$ can only be used to open group signatures without the signing ability.

The concrete examples in the literature that have applied this research strategy include:

- Certificateless Public Key Cryptography (Asiacrypt 2003) [6]. In all identity-based encryption schemes before this work, the private key generator (PKG) knows the private keys of all users and can decrypt their ciphertexts. The scheme proposed in this work has introduced a mechanism where the private keys generated by the PKG are just partial keys and cannot decrypt ciphertexts because they are computed using identities and additional public keys chosen by users.

- Registration-Based Encryption: Removing Private-Key Generator from IBE (TCC 2018) [69]. In all identity-based encryption schemes before this work, the private key generator (PKG) knows the private keys of all users and can decrypt their ciphertexts. The scheme proposed in this work has introduced a mechanism where the PKG no longer generates private keys for users but just collects users' registered public keys and identities into the master public key, while encryption is still as convenient as normal IBE.

- Group Signatures with User-Controlled and Sequential Linkability (PKC 2021) [55]. In all group signature schemes before this work, the group manager can open group signatures to know the signers' identities and some schemes even allow the group manager to publish a token to allow the public to verify that two group signatures were generated by the same signer. The scheme proposed in this work has introduced a mechanism where only the group signers can control the linkage of his/her generated signatures.

7.2.3 Area: Enhanced Privacy

Strategy 23 (Enhanced Privacy: From Public to Private) *In the first scheme, A has done some secret computations for B. But A(B) is able to gain some parameters related to B(A)'s privacy or concern. The second scheme aims to propose a mechanism to allow B(A) to hide these parameters.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, given a group signature σ_m for message m, any verifier cannot know who is the identity of the real signer except the group manager, but the verifier can obtain the group capacity n from the group public key and a group signature. The second scheme has introduced a new construction where the identity of the real signer and the group capacity are both anonymous to the verifiers.

The concrete examples in the literature that have applied this research strategy include:

- Short Redactable Signatures Using Random Trees (CT-RSA 2009) [46]. In all redactable signature schemes before this work, a signature receiver can remove any substrings from a signed message without impact on signature verification, but verifiers can see the length of removed substrings. The scheme proposed in this work has introduced a mechanism where the length of the removed substring in a redactable signature can be hidden.

- Unlinkability of Sanitizable Signatures (PKC 2010) [37]. In all sanitizable signature schemes before this work, a signer can delegate a third party to modify a signed message under his/her modification instruction while keeping the signature valid, but it is possible for verifiers to identify that two sanitized signatures on two messages are actually from the same original message and its signature. The scheme proposed in this work has introduced a mechanism where a verifier does not know whether two sanitizable signatures are from the same original signature or not.

- Threshold Signatures with Private Accountability (Crypto 2022) [31]. In all threshold signature schemes before this work, there exist some schemes that can account for the set of signers who generated a threshold signature, but this accountability is public and everyone can know the set of signers. The scheme proposed in this work has introduced a mechanism where only an entity having a secret key can know the set of signers of a threshold signature.

Strategy 24 (Enhanced Privacy: From Clear to Fuzzy) *In the first scheme, A has done some secret computations for B. But one party is able to gain some parameters related to another party's privacy or concern. The second scheme aims to propose a mechanism to let those gained parameters become fuzzy when the other entity tries to know them.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, any group signer under the group public key gpk can generate any group signature for any message without leaking his/her identity, but verifiers know that the signer must be from the group gpk. In some complicated scenarios, this kind of privacy protection could be not enough for a group signer. The second scheme has introduced a ring based group signature scheme. In this scheme, a group signer under gpk^t taking as input the group signer key gsk_i and other group public keys $(gpk^1, \cdots, gpk^{t-1}, gpk^{t+1}, \cdots, gpk^N)$ can generate a ring-group signature σ_m, where the verification result shows that the real signer is a group signer from one of the groups in $(gpk^1, \cdots, gpk^N)$. Further, adding this ring mechanism will not impact the opening ability of the group manager using $gmsk^t$.

The concrete examples in the literature that have applied this research strategy include:

- Designated Verifier Proofs and Their Applications (Eurocrypt 1996) [96]. In all non-interactive zero-knowledge proof schemes before this work, upon receiving proof, any party can be convinced that the prover indeed knows the witness of the proven statement. The scheme proposed in this work has introduced a mechanism where only designed confirmer Jake can be convinced that the prover indeed knows the witness, while others can only know that either the prover knows the witness or it is cheating by Bob.

- How to Leak a Secret (Asiacrypt 2001) [128]. In all digital signature schemes before this work, once a signature is generated, the signer's identity will be publicly verifiable or knowable by a group manager using a group signature scheme. The scheme proposed in this work has introduced a mechanism where the real signer of a signature is hidden in one ring of signers that were totally decided by the real signer but no one can trace the real signer.

- Multimodal Private Signatures (Crypto 2022) [119]. In all private signature schemes before this work, the signer can set the identity of signing a message to be traceable or not using different signature notions, but there are not too many choices for signers except full anonymity or full tractability by an authority. The scheme proposed in this work has introduced a mechanism that allows signers to be partially traced by an authority who knows a fuzzy identity of the signer.

7.2.4 Area: Enhanced Security

Strategy 25 (Enhanced Security: From Model-Strong to Model-Stronger) *In the first scheme, A will do multiple secret computations. But one party's security will be compromised if another party can obtain some additional computing results. The second scheme aims to propose a scheme that it is still secure for one party even if another party gains these additional computing results.*

The artificial example applying this research strategy is as follows. The security of group signatures requires that if an adversary only knows the group signing keys of identities in the set S, then all signatures computed by the adversary must be opened with an identity in S. In the first group signature scheme, the traceability is secure if and only if the adversary can only query and ask the group manager to open those valid group signatures. If the adversary is able to query invalid group signatures and obtain those opened results, the adversary is able to break the traceability. The second scheme is more secure because the traceability cannot be broken even if the adversary can query invalid group signatures and receive their opened results.

The concrete examples in the literature that have applied this research strategy include:

- Short Signatures Without Random Oracles (Eucorypt 2004) [25]. In all digital signature schemes before this work, they are proved unforgeable where the adversary cannot forge a valid signature on a new message without being queried. The scheme proposed in this work is proven secure in a stronger security model where the adversary is allowed to forge a valid signature on a queried message as long as the forged signature is different from the queried signature.
- Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model (Crypto 2009) [7]. In all cryptography schemes before this work, they are proven secure on the condition that the adversary knows nothing about the secret key being used for secret operations. The scheme proposed in this work is proven secure in a stronger security model where the adversary is allowed to obtain part of the secret key denoted by $f(sk)$ and f is an arbitrary leaking function.
- Deniable Authentication when Signing Keys Leak (Eurocrypt 2023) [45]. A deniable authentication scheme allows a sender to authentically send messages to a receiver and only the receiver can be convinced that they were indeed sent from the sender. In all deniable authentication schemes before this work, they are proven secure against the adversary who is not allowed to query the secret key of the sender. The scheme proposed in this work is proven secure in a stronger security model where the adversary can query and obtain the sender's secret key.

Strategy 26 (Enhanced Security: From Universal to Partial) *In the first scheme, A can do multiple types of secret computations that will be used by B. But each result of type-1 computation is very powerful and will impact all related results of type-2 computation. The second scheme aims to propose a mechanism to allow A to let type-1 computation have a partial impact on related type-2 computation.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the group manager secret key $gmsk$ is very powerful. Once it is published, anyone can use it to know the identity of the group signer in every group signature. Therefore, if David has been corrupted and all his generated signatures should be revealed for auditing, the group manager cannot simply publish $gmsk$ for everyone to trace and audit those signatures generated by David. The second scheme has introduced a mechanism where the group manager can use $gmsk$ to generate a special secret key, denoted by $gmsk_D$, that can only open those group signatures generated by David's group signing key.

The concrete examples in the literature that have applied this research strategy include:

- Traceable Signatures (Eurocrypt 2004) [99]. In all group signature schemes before this work, the group master secret key is very powerful because it can open all group signatures once it is published. The scheme proposed in this work has introduced a mechanism where the group manager can issue a token for the public to trace all signatures generated by a single group signer. This has protected the security of group signatures generated by other group signers, when compared to releasing the master secret key.

- Sanitizable Signatures (ESORICS 2005) [9]. In all proxy signature schemes, the proxy can use the proxy key to sign any messages on behalf of the original signer. The scheme proposed in this work has introduced a mechanism where the proxy can only modify part of a signed message generated by the original signer and the modification is under the original signer's control. This has protected the security of signed messages when signatures are generated by a proxy.

- Traceable Group Encryption (PKC 2014) [107]. In all group encryption schemes before this work, the group master secret key is powerful because it can open and know the real ciphertext receiver once it is published. The scheme proposed in this work has introduced a mechanism where the group manager can issue a token for the public to trace all ciphertexts generated for a specific receiver while keeping other ciphertext receivers anonymous. This has protected the anonymity of other receivers in the group encryption, when compared to releasing the master secret key.

7.3 Type 6: Better Service

7.3.1 Area: Delegable

Strategy 27 (Delegable: From Personal to Aided) *In the first scheme, B can only complete computations by himself/herself. But the computations are heavy for B. The second scheme aims to propose a mechanism to allow B to complete the computation with the help of an untrusted third party.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, verifying a group signature needs to conduct hundreds of exponentiations which is heavy for verifiers equipped with lightweight computing devices. The second scheme has introduced an aided mechanism where a verifier can use a rather small computation cost to verify a group signature with the aid of an untrusted third party including the group signer.

The concrete examples in the literature that have applied this research strategy include:

- Speeding Up Secret Computations with Insecure Auxiliary Devices (Crypto 1988) [114]. In all digital signature schemes before this work, the signing operation is expensive especially when it happens in lightweight devices. The scheme proposed in this work has introduced a mechanism where signing operations can be completed with the help of an untrusted party.

- Server(Prover/Signer)-Aided Verification of Identity Proofs and Signatures (Eurocrypt 1995) [108]. In all identification and digital signature schemes before this work, the verification operation is expensive especially when it happens in lightweight devices. The scheme proposed in this work has introduced a mechanism where verification operations can be completed with the help of an untrusted party.

- Identity-Based Server-Aided Decryption (ACISP 2011) [109]. In all identity-based encryption schemes before this work, the decryption operation is expensive especially for lightweight devices. The scheme proposed in this work has introduced a mechanism where decryption operations can be completed with the help of an untrusted party.

Strategy 28 (Delegable: From Personal to Proxy) *In the first scheme, the secret computations done by A cannot be done by B. The second scheme aims to propose a mechanism to allow A to securely delegate the ability of secret computation to B.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, generating a group signature can only be completed by group signers who have group signing keys. However, in the real world, a group signer David might be on holiday and cannot do the signing job. The second scheme has introduced a proxy mechanism where David can delegate the group signing ability to a proxy Frank without giving his group signing key directly. More precisely, with the help of time stamping, after the end of the proxy date, the proxy group signing key will automatically become invalid. Further, group signatures generated by David using the group signing key and generated by proxy using the proxy group signing key are indistinguishable from the view of all verifiers.

The concrete examples in the literature that have applied this research strategy include:

- Proxy Signatures for Delegating Signing Operation (ACMCCS 1996) [113]. In all digital signature schemes before this work, a signature cannot be issued if the signer is not available. The scheme proposed in this work has introduced a non-trivial mechanism where the signer can securely delegate the signing right to a proxy without directly giving the signing key to the proxy.

- Designated Confirmer Signatures (Eurocrypt 1994) [47]. In all undeniable signature schemes before this work, a signature cannot be confirmed or disavowed if the signer is not available. The scheme proposed in this work has introduced a non-trivial mechanism where the signer can securely delegate a third party to help verifiers verify signatures.

- Public Key Encryption with Keyword Search (Eurocrypt 2004) [27]. In all public key encryption schemes, once a keyword is encrypted and attached to plaintext, no one can see the keyword unless having the corresponding secret key. The scheme proposed in this work has introduced a non-trivial mechanism where the key owner can securely delegate the keyword search to a third party who knows nothing about keywords or secret key.

Strategy 29 (Delegable: From Personal to Convertible) *In the first scheme, A can do multiple types of secret computations (type-1 and type-2) for B. The second scheme aims to propose a mechanism for A to allow someone to transform the result of type-1 computation to type-2 when authorized by A.*

The artificial example applying this research strategy is as follows. A group signature scheme is a special digital signature where there is more than one signer on behalf of a group public key and there is a group manager who can open any group signature to know the identity of the group signer. In the first group signature scheme, the generated group signatures are different from normal digital signatures. In some complex scenarios, we might need the support of both group signatures and digital signatures. The second scheme has introduced a convertible mechanism. By generating a special secret key for the group manager, a group signer can generate a group signature with gsk_i and the group signature can be converted into a normal digital signature under the public key pk by the group manager.

The concrete examples in the literature that have applied this research strategy include:

- Convertible Undeniable Signatures (Crypto 1990) [34]. In all undeniable signature schemes before this work, an undeniable signature on m cannot be verified if the signer is not available, and the signer needs to re-generate a normal signature if the signer wants to make the verification become public. The scheme proposed in this work has introduced a non-trivial mechanism where undeniable signatures can be converted to normal signatures by a third party with the help of secret tokens generated by the signer.

- Divertible Protocols and Atomic Proxy Cryptography (Eurocrypt 1998) [21]. In all public-key encryption schemes before this work, a ciphertext computed for pk_A cannot be decrypted by the key owner of pk_B unless the key owner of sk_A decrypts it first and then encrypts the message under pk_B. The scheme proposed in this work has introduced a non-trivial mechanism where a proxy after authorization can convert any ciphertext under pk_A to ciphertext under pk_B.

- Universal Designated-Verifier Signatures (Asiacrypt 2003) [133]. In all designated-verifiable signature schemes before this work, only the signer or the receiver can generate such a signature using a secret key. The scheme proposed in this work has introduced a non-trivial mechanism where normal signatures can be converted into designated verified signatures without having secret keys.

7.3.2 Area: Flexible

Strategy 30 (Flexible: From Online to Offline) *In the first scheme, A(B) has to do the computations in the online phase after receiving some parameters. The second scheme aims to allow A(B) to flexibly do part of computations in the offline phase.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the construction is based on cyclic groups and generating each group signature requires computing 100 exponentiations. The second scheme has introduced an online/offline mechanism where all exponentiations can be completed in the offline phase without knowing the message to be signed and only one modular multiplication is needed in the online phase after receiving the message.

The concrete examples in the literature that have applied this research strategy include:

- On-line/Off-line Digital Signatures (Crypto 1989) [59]. In all digital signature schemes before this work, all signing operations are inefficient. The scheme proposed in this work has introduced an online/offline signing mechanism where the offline phase can complete heavy precomputations without knowing the message to be signed, and the online phase after knowing the message is much faster than the offline phase.

- Identity-Based Online/Offline Encryption (FC 2008) [85]. In all identity-based encryption schemes before this work, all encryption operations require to take several exponentiations. The scheme proposed in this work has introduced an online/offline encryption mechanism where the offline phase can complete heavy precomputations without knowing the receiver identity and message to be encrypted, and the online phase after knowing the identity and message is very fast with little time cost.

- Online/Offline Attribute-Based Encryption (PKC 2014) [93]. In all attribute-based encryption schemes before this work, all encryption operations are inefficient especially when the access policy is complicated. The scheme proposed in this work has introduced an online/offline encryption mechanism in the key encapsulation mechanism setting, where the offline phase can complete heavy precomputations without knowing the access policy, and the online phase is very fast in encapsulating a session key into a ciphertext under the given access policy.

Strategy 31 (Flexible: From Static to Dynamic) *In the first scheme, A needs to do some secret computations for B. But what A can do has been fixed after the setup. The second scheme aims to allow A to do secret computations in a dynamic way where some secret computations are not defined during the setup phase.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, all group signing keys are generated in the setup phase for all group signers meaning that the group signers must join the group at the beginning. The second scheme has introduced a dynamic mechanism where users can also join as group signers after the generation of the group public key. Further, the group manager does not know each group signing key in the second scheme.

The concrete examples in the literature that have applied this research strategy include:

- Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials (Crypto 2002) [39]. An accumulator scheme allows one to put a large set of inputs $(x_1, x_2, \cdots, x_n)$ into one short value X, called accumulator, such that there is a witness for proving that a value x_i ($i \in [1, n]$) has been accumulated in X. In all accumulator schemes before this work, there exist efficient schemes with short witness for proving $x_i \in X$, but they do not allow updating the set of inputs. The scheme proposed in this work has introduced a dynamic mechanism where one can add or delete inputs in X.

- Dynamic Threshold Public-Key Encryption (Crypto 2008) [53]. Threshold public key encryption (TPKE) is a specific encryption where the decryption key corresponding to a public key is shared among a set of n users and decryption needs at least t users to cooperate. In all TPKE schemes before this work, the set of n users must be fixed during the setup. The scheme proposed in this work has introduced a dynamic mechanism where users can dynamically join into the set and cooperate with the decryption.

- Dynamic Provable Data Possession (ACMCCS 2009) [58]. Proof of storage (PoS) is a cryptography scheme where a client can efficiently verify the integrity of data remotely stored in the cloud. In all PoS schemes before this work, they only consider the case that all stored data are static and cannot be updated. The scheme proposed in this work has introduced a dynamic mechanism where the PoS still works even if the client wants to update the outsourced data by inserting, modifying, or deleting part of the data.

Strategy 32 (Flexible: From Necessary to Unnecessary) *In the first scheme, A will do secret computations while B can verify the computing results. But these computations require the holding of some assumptions first. The second scheme aims to allow A(B) to complete the computations without the need for these assumptions to improve the practicality.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, any group signer can be revoked by the group manager who simply updates a revocation list set as part of the group public key. This revocation list must be known by all verifiers. However, the first scheme also requires all other group signers to be informed of the updated revocation list. The second scheme has introduced a simplified mechanism where there is no need to inform other group signers when some group signers are revoked and the revocation list has been updated.

The concrete examples in the literature that have applied this research strategy include:

- Sequential Aggregate Signatures with Lazy Verification from Trapdoor Permutations (Asiacrypt 2012) [36]. A sequential aggregate signature scheme allows n signers to sign a message each in order and finally produce a short signature. In all sequential aggregate signature schemes before this work, they require a signer to do verification on the aggregate-so-far signature before adding its own signature for security purposes. The scheme proposed in this work has introduced a mechanism where this kind of requirement is not needed and can be removed.

- Let a Non-barking Watchdog Bite: Cliptographic Signatures with an Offline Watchdog (PKC 2019) [51]. In all digital signature schemes before this work, all schemes secure in the presence of kleptographic attacks require an online watchdog to collect all communicating transcripts. The scheme proposed in this work has introduced a mechanism where an online watchdog is not necessary and an offline watchdog can clip the power of subversions via only one-time black-box testing of the implementation.

- Non-interactive Blind Signatures for Random Messages (Eurocrypt 2023) [87]. In all blind signature schemes before this work, all schemes require online interactions between the signer and the receiver to protect the anonymity of the message to be signed. The scheme proposed in this work has introduced a mechanism where online interaction is not necessary and the signer can still generate a blind signature for a specific receiver in mind.

Strategy 33 (Flexible: From Coarse-Grained to Fine-Grained) *In the first scheme, A will do secret computations for B to continue some secret computations. An entity knows that the secret computation was done by an entity who is the key owner of X. The second scheme aims to propose a fine-grained mechanism such that X is extended to Y, and Y shows fine-grained information about who the entity is. For example, X is just a random public key and Y is the identity of the entity.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, given a group signature for m, any verifier can run the verification to check whether it is a valid group signature from the group under the public key gpk. The group public key gpk itself has no sense and we need a certificate to indicate the owner identity of this gpk. The second scheme has proposed an identity-based group signature scheme, where gpk can be any identity of the group, while $gmsk$ is computed from gpk and a master secret key held by a trusted third party.

The concrete examples in the literature that have applied this research strategy include:

- Identity-Based Cryptosystems and Signature Schemes (Crypto 1984) [131]. In all public key encryption schemes before this work, when a message is encrypted with a public key pk, the sender does not know who is the real receiver unless there is a certificate showing who owns this public key. The scheme proposed in this work has introduced a new notion where the sender can directly encrypt messages using the receiver's identity while the receiver can decrypt the ciphertext with a private key generated by a private key generator (PKG).

- Identity-Based Undeniable Signatures (CT-RSA 2004) [106]. In all undeniable signature schemes before this work, when an undeniable signature is successfully verified, the verifier only knows that he/she is interacting with someone who owns the corresponding public key pk but it reveals nothing about the owner's identity. The scheme proposed in this work has introduced a new notion where the verifier knows that he/she is interacting with a signer corresponding to an identity known by the verifier.

- Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data (ACMCCS 2006) [81]. In all identity-based encryption schemes before this work, a message is encrypted for coarse-grained receivers who are determined by the sender before the encryption. The scheme proposed in this work has introduced a new notion where the sender enables fine-grained access control of encrypted data using flexible access policies.

7.3.3 Area: Scalable

Strategy 34 (Scalable: From Single to Multiple) *In the first scheme, A can do secret computations while B can verify the computing results. But if A(B) needs to do multiple times of computations, he/she can only repeat them one by one. The second scheme aims to propose a mechanism to allow A(B) to do multiple computations at the same time.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, verifying a group signature needs to conduct hundreds of exponentiations. The verification cost increases linearly in the number of input group signatures because the verification algorithm can verify only one signature each time and a verifier has to verify them one by one. The second scheme has introduced a batch verification algorithm where the verification algorithm can verify multiple signatures at the same time. Without decreasing any security, the second scheme has significantly improved the verification efficiency and the verification cost is about 10 exponentiations for each group signature on average.

The concrete examples in the literature that have applied this research strategy include:

- Broadcast Encryption (Crypto 1993) [62]. In all public-key encryption schemes before this work, if a secret message needs to be sent to a set of receivers, the sender has to generate n independent ciphertexts such that the communication cost increases linearly in the number of receivers. The scheme proposed in this work has introduced a mechanism where the complexity length of ciphertext sent to a set of receivers has been reduced and smaller than $O(n)$.

- Aggregate and Verifiably Encrypted Signatures from Bilinear Maps (Eurocrypt 2003) [29]. In all digital signature schemes before this work, if n signatures on distinct messages by different signers need to be stored or transferred, the cost must be linear in the number of n. The scheme proposed in this work has introduced a mechanism where these signatures can be aggregated into a constant one without impact on verification.

- Reusable Garbled Circuits and Succinct Functional Encryption (STOC 2013) [76]. Garbled circuits (GC) allow computing a function f on an input x without leaking anything about f or x besides $f(x)$. In all GC schemes before this work, they offer no security if one GC is used on multiple inputs x. The scheme proposed in this work has introduced a mechanism where one GC can be reusable for multiple inputs multiple times.

Strategy 35 (Scalable: From Bounded to Unbounded) *In the first scheme, A needs to do some secret computations for B. But what A(B) can do are bounded with some parameters and this has negatively impacted A(B)'s benefits. The second scheme aims to allow A(B) to do secret computations without being bounded by these parameters.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the setup algorithm needs to input a number n to decide the group capacity. That is, the number of group signers for a group public key cannot be more than n. The second scheme has proposed an unbounded mechanism where the setup algorithm can be run without having any bound number n and the group manager can add an unbounded number (polynomial size) of users as group signers.

The concrete examples in the literature that have applied this research strategy include:

- Efficient Generic Forward-Secure Signatures with an Unbounded Number of Time Periods (Eurocrypt 2002) [112]. In all forward-secure signature schemes before this work, to generate a key pair, the user must specify the total number of time periods for updating the signing key. The scheme proposed in this work has introduced a mechanism where the total number is unbounded and does not need to be fixed in advance when the key pair is generated.

- Constrained PRFs for Unbounded Inputs (CT-RSA 2016) [5]. Constrained pseudorandom function (CPRF) allows one to evaluate $F(k_S, \cdot)$ on all input x from a predefined set S only with a constrained key k_S. In all CPRF schemes before this work, the input length must be fixed beforehand during key generation. The scheme proposed in this work has introduced a mechanism where the input length is unbounded.

- Unbounded HIBE and Attribute-Based Encryption (Eurocrypt 2011) [104]. In all HIBE schemes before this work, there exist schemes in the standard model but the maximum hierarchy depth had to be fixed at the setup phase. The scheme proposed in this work has introduced a mechanism where the hierarchy depth is unbounded and the scheme is proven secure in the standard model.

Strategy 36 (Scalable: From Narrow to Wide) *In the first scheme, A needs to do some secret computations for B. But the computation is limited in a small space. The second scheme aims to allow A to do secret computations in a larger space to benefit applications.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, the opening algorithm on input of a valid group signature will return an index $i \in [1, n]$ showing the identity of the group signer. The second scheme has introduced a more powerful mechanism that will return any arbitrary string as long as it is linked to the identity of that group signer. There are two benefits of the second scheme. First, even if i is leaked, it will not leak the capacity of the group. Second, there is no need for the group manager to record the relations between identities and indexes to trace identities from indexes, as long as the arbitrary string is set as the identity.

The concrete examples in the literature that have applied this research strategy include:

- 1-out-of-n Signatures from a Variety of Keys (Asiacrypt 2002) [3]. In all ring signature schemes before this work, all public keys in a ring must have the same flavor of keys such as RSA-keys or DL-type keys. The scheme proposed in this work has introduced a mechanism that allows a mixture use of different flavors of keys at the same time.

- Attribute-Based Encryption for Circuits (STOC 2013) [78]. In all attribute-based encryption schemes before this work, the most flexible access policy is the boolean-formula based predicate. The scheme proposed in this work has introduced a mechanism that allows supporting larger classes of predicates for circuits of any arbitrary polynomial size.

- Functional Commitments for All Functions, with Transparent Setup and from SIS (Eurocrypt 2023) [44]. A functional commitment (FC) allows a user to commit to a function from a specified family, then later reveal values of evaluating desired inputs which are verifiable. All practical FC schemes before this work only support linear functions. The scheme proposed in this work has introduced a mechanism that can practically support nonlinear functions or all functions with any bounded complexity.

7.3.4 Area: Integrable

Strategy 37 (Integrable: From Single-Service to Bunch-Service) *In the first scheme, A needs to do multiple secret computations for B who will use all computing results later. But these secret computations are separated and independent. The second scheme aims to combine all computations together to benefit A(B) in computing (use).*

The artificial example applying this research strategy is as follows. We consider the scenario where group signatures are applied in certificate and signatures. An authority uses the group public key gpk^* to issue certificates for users, namely generating group signatures on messages $(name, gpk)$, where $name$ refers to the name of a group and gpk refers to the group public key of a group. The group signers in gpk will use their group signing keys to sign on digital documents for e-business. To convince a verifier that m was published by the organization $name$, the verifier needs to verify that (1) the group signature σ_m for m is valid under gpk, and (2) gpk has a valid certificate for $name$ under gpk^*. We can use the first scheme to meet the above application scenario, but we need to send both the certificate and the group signature σ_m to the verifier. The second scheme has introduced a certificate-based mechanism where the certificate for gpk and the group signature for m can be bunched together in computation and transmission. The comparison shows that the second scheme has saved 50% percent of computation cost and communication costs.

The concrete examples in the literature that have applied this research strategy include:

- Digital Signcryption or How to Achieve Cost(Signature $\&$ Encryption) $<<$ Cost(Signature) + Cost(Encryption) (Crypto 1997) [143]. In all cryptography schemes before this work, there exist public key encryption schemes for data confidentiality and signature schemes for data integrity, but they did not consider how to efficiently address the applications that need both confidentiality and integrity. The scheme proposed in this work has introduced a new notion that can efficiently combine the application of public key encryption and digital signatures.

- Securely Combining Public-Key Cryptosystems (ACMCCS 2001) [86]. In all cryptography schemes before this work, to be able to sign messages, the user needs to generate a key pair for signature purposes; to be able to decrypt ciphertexts of public key encryption, the user also needs to generate a key pair for encryption purposes. The scheme studied in this work has introduced how to combine and use one key pair for both signing and decryption.

- A Certificate-Based Signature Scheme (CT-RSA 2004) [98]. In all cryptography schemes before this work, to convince verifiers that a message is published by David, David should run a digital signature scheme to sign on the message using a key pair (pk, sk) and also run a certificate scheme to obtain a certificate showing that pk belongs to David. The scheme proposed in this work has introduced a new notion that can efficiently combine the need of publishing both digital signatures and certificates on their public keys.

Strategy 38 (Integrable: From Space-Wide to Space-Narrow) *In the first scheme, A needs to do some secret computations for B. But what A can do are rather general over large defined spaces and these computing results cannot be well applied. The second scheme aims to restrict A's secret computations in a smaller space to enjoy some nice features.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, any arbitrary string can be set as messages to be signed because messages are hashed into integers before the signing operation. The final group signature is composed of some group elements and some modular integers. The second scheme has redefined the spaces of all these objects, where the group public key, group signatures, and messages to be signed are all group elements. With the restriction of these spaces, although the second scheme is less efficient than the first scheme, it can be applied as a building block and solve some problems in applications using zero-knowledge proofs.

The concrete examples in the literature that have applied this research strategy include:

- Invariant Signatures and Non-Interactive Zero-Knowledge Proofs are Equivalent (Crypto 1992) [77]. In all digital signature schemes before this work, the signing algorithm is defined as a probabilistic algorithm. The scheme proposed in this work has introduced a new definition where the signing algorithm is deterministic and the signature on a message is unique, which shows some new exciting applications.
- Deterministic and Efficiently Searchable Encryption (Crypto 2007) [15] In all public key encryption schemes before this work, the encryption algorithm is defined as a probabilistic algorithm. The scheme proposed in this work has introduced a new definition where the encryption algorithm is deterministic, which shows some new exciting applications.
- Structure-Preserving Signatures and Commitments to Group Elements (Crypto 2010) [1]. In all digital signature schemes before this work, the space of messages, public keys, and signatures can be arbitrary. The scheme proposed in this work has introduced a new definition where all those spaces are elements of a pairing group, which shows some new exciting applications.

7.3.5 Area: Relaxable

Strategy 39 (Relaxable: From Object-Wide to Object-Narrow) *In the first scheme, what A and B can do are defined in a strong way where the defined algorithm captures multiple cases. It was found that what A and B will do in a specific scenario has fewer cases to be considered. The second scheme aims to be efficiently reconstructed for this scenario by relaxing the algorithm definition.*

The artificial example applying this research strategy is as follows. In the first group signature scheme, any arbitrary string can be set as messages to be signed because messages are hashed into integers before the signing operation. The final group signature is composed of hundreds of group elements. We found that in many application scenarios, the message to be signed is very short. The second scheme has introduced a new construction where the signature size is linear in the bit length of the message if the message to be signed is less than the order of cyclic groups.

The concrete examples in the literature that have applied this research strategy include:

- Identity-Based Aggregate Signatures (PKC 2006) [73]. In all aggregate signature schemes before this work, there exist schemes that can aggregate any signatures into a short one, but it is still unknown how to efficiently aggregate identity-based signatures. The scheme proposed in this work has introduced how to aggregate those identity-based signatures which are generated with the same tag.
- Batch Verification of Short Signatures (Eurocrypt 2007) [38]. In all batch verification of signature schemes before this work, there exist efficient batch verification schemes on pairing-based signatures, but the batching cost still requires linear pairing operations in the number of signatures. The scheme proposed in this work has introduced how to do batch verification with a constant number of pairing operations on those pairing-based short signatures which are generated under the same time period.

- Locally Verifiable Signature and Key Aggregation (Crypto 2022) [79]. In all aggregate signature schemes before this work, each verification will allow a verifier to know that n messages have been signed but the computation cost is linear in the number of n. The scheme proposed in this work has introduced how to just let one verify one signed message in the aggregate signature with computation cost independent of n.

Strategy 40 (Relaxable: From Security-Strong to Security-Weak) *In the first scheme, A will do multiple secret computations for B. It was found that B will receive fewer computing results from A or B has received more restrictions in computing than what A expects in a specific scenario. The second scheme aims to be efficiently reconstructed for this scenario by relaxing the security definition.*

The artificial example applying this research strategy is as follows. The security of group signatures requires that if an adversary only knows the group signing keys of identities in the set S, then all computed signatures by the adversary must be opened with an identity in S. In the first group signature scheme, the traceability is secure for any corruption as long as the set S satisfying $|S| \leq n-1$. We found some very special application scenarios where the adversary can corrupt at most one group signer. The second scheme has a special construction where the traceability is secure if and only if the adversary can only corrupt one group signer. The comparison shows that this weakened security requirement has significantly improved efficiency. Each group signature in the second scheme is composed of three group elements only, while it is composed of hundreds of group elements in the first scheme.

The concrete examples in the literature that have applied this research strategy include:

- Relaxing Chosen-Ciphertext Security (Crypto 2003) [42]. In all public key encryption schemes before this work, the IND-CCA security model is the standard and widely-accepted security model. The scheme proposed in this work has relaxed this standard security model to a replayable CCA security model, which is shown to be sufficient for many applications.

- Relaxed Security Notions for Signatures of Knowledge (ACNS 2011) [65]. Signatures of knowledge (SoK) allows one who knows the witness of any NP statement to sign messages. In all SoK schemes before this work, simulatability was defined to ensure anonymity in applications. The scheme proposed in this work has relaxed the definition of the simulatability, which is shown to be sufficient for many applications.

- A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels (PKC 2022) [20]. In all cryptography schemes for secure message transfer before this work, IND-CCA security has been found not necessary for this application and there have been some proposed weaker security models and their constructions. The scheme proposed in this work has further relaxed the security definition from IND-CCA which is shown to be sufficient for this application, and the corresponding construction is simpler and more efficient.

8 Conclusion

We have introduced the research philosophy of cryptography research behind more than 800 academic papers that we have surveyed. In short, the primary motivation of proposing new schemes in these papers is to advance novel knowledge for humanity. While the novelty of proposed knowledge is primarily reflected by additional benefits, although some may be little, blurred, or come at a cost due to tradeoffs. In this paper, 40 research strategies from classified 17 benefit areas were categorized to help researchers know what benefits we can explore in research. It is worth noting that the introduced 40 research strategies are not all of the research strategies that our community has found. Further research strategies can be explored based on the features and properties of each benefit area.

To conclude this work, we firmly believe that cryptography research is an inexhaustible and never-ending pursuit in the history of human activities, no matter how excellent research outcomes our community has achieved. While individual contributions to cryptography research may appear insignificant in the grand scheme of human civilization, the collective impact of all researchers is bringing about significant qualitative changes for humanity.

References

[1] Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer (2010)

[2] Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal structure-preserving signatures in asymmetric bilinear groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer (2011)

[3] Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 415–432. Springer (2002)

[4] Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 271–286. Springer (2000)

[5] Abusalah, H., Fuchsbauer, G., Pietrzak, K.: Constrained prfs for unbounded inputs. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 413–428. Springer (2016)

[6] Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer (2003)

[7] Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer (2009)

[8] Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures (extended abstract). In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 591–606. Springer (1998)

[9] Ateniese, G., Chou, D.H., de Medeiros, B., Tsudik, G.: Sanitizable signatures. In: di Vimercati, S.D.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 159–177. Springer (2005)

[10] Au, M.H., Huang, Q., Liu, J.K., Susilo, W., Wong, D.S., Yang, G.: Traceable and retrievable identity-based encryption. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 94–110 (2008)

[11] Auerbach, B., Cash, D., Fersch, M., Kiltz, E.: Memory-tight reductions. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 101–132. Springer (2017)

[12] Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostáková, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized channels from limited blockchain scripts and adaptor signatures. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13091, pp. 635–664. Springer (2021)

[13] Bagherzandi, A., Jarecki, S.: Identity-based aggregate and multi-signature schemes based on RSA. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 480–498. Springer (2010)

[14] Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.: Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Roy, B.K. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 515–532. Springer (2005)

[15] Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer (2007)

[16] Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer (2003)

[17] Bellare, M., Rogaway, P.: The exact security of digital signatures - how to sign with RSA and rabin. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer

[18] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) CCS 1993. pp. 62–73. ACM (1993)

[19] Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 368–397. Springer (2015)

[20] Beskorovajnov, W., Gröll, R., Müller-Quade, J., Ottenhues, A., Schwerdt, R.: A new security notion for PKC in the standard model: Weaker, simpler, and still realizing secure channels. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022. LNCS, vol. 13178, pp. 316–344. Springer (2022)

[21] Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer (1998)

[22] Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Signatures on randomizable ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 403–422. Springer (2011)

[23] Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: Ning, P., Syverson, P.F., Jha, S. (eds.) CCS 2008. pp. 417–426. ACM (2008)

[24] Boneh, D., Bonneau, J., Bünz, B., Fisch, B.: Verifiable delay functions. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 757–788. Springer (2018)

[25] Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer (2004)

[26] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer (2004)

[27] Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer (2004)

[28] Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer (2001)

[29] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer (2003)

[30] Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer (2005)

[31] Boneh, D., Komlo, C.: Threshold signatures with private accountability. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13510, pp. 551–581. Springer (2022)

[32] Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer (2001)

[33] Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational diffie-hellman. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 229–240. Springer (2006)

[34] Boyar, J., Chaum, D., Damgård, I., Pedersen, T.P.: Convertible undeniable signatures. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 189–205. Springer (1990)

[35] Bresson, E., Stern, J.: Efficient revocation in group signatures. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 190–206. Springer (2001)

[36] Brogle, K., Goldberg, S., Reyzin, L.: Sequential aggregate signatures with lazy verification from trapdoor permutations - (extended abstract). In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 644–662. Springer (2012)

[37] Brzuska, C., Fischlin, M., Lehmann, A., Schröder, D.: Unlinkability of sanitizable signatures. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 444–461. Springer (2010)

[38] Camenisch, J., Hohenberger, S., Pedersen, M.Ø.: Batch verification of short signatures. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 246–263. Springer (2007)

[39] Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer (2002)

[40] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS 2001. pp. 136–145. IEEE Computer Society (2001)

[41] Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: Vitter, J.S. (ed.) STOC, 1998. pp. 209–218. ACM (1998)

[42] Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer (2003)

[43] Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: Reif, J.H. (ed.) STOC, 2002. pp. 494–503. ACM (2002)

[44] de Castro, L., Peikert, C.: Functional commitments for all functions, with transparent setup. IACR Cryptol. ePrint Arch. p. 1368 (2022), `https://eprint.iacr.org/2022/1368`

[45] Chakraborty, S., Hofheinz, D., Maurer, U., Rito, G.: Deniable authentication when signing keys leak. IACR Cryptol. ePrint Arch. p. 213 (2023)

[46] Chang, E., Lim, C.L., Xu, J.: Short redactable signatures using random trees. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 133–147. Springer (2009)

[47] Chaum, D.: Designated confirmer signatures. In: Santis, A.D. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 86–91. Springer (1994)

[48] Chaum, D., Antwerpen, H.V.: Undeniable signatures. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 212–216. Springer (1989)

[49] Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 435–460. Springer (2013)

[50] Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer (1994)

[51] Chow, S.S.M., Russell, A., Tang, Q., Yung, M., Zhao, Y., Zhou, H.: Let a non-barking watchdog bite: Cliptographic signatures with an offline watchdog. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11442, pp. 221–251. Springer (2019)

[52] Coron, J.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer (2002)

[53] Delerablée, C., Pointcheval, D.: Dynamic threshold public-key encryption. In: Wagner, D.A. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 317–334. Springer (2008)

[54] Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer (1989)

[55] Diaz, J., Lehmann, A.: Group signatures with user-controlled and sequential linkability. In: Garay, J.A. (ed.) PKC 2021. LNCS, vol. 12710, pp. 360–388. Springer (2021)

[56] Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)

[57] Döttling, N., Garg, S.: Identity-based encryption from the diffie-hellman assumption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 537–569. Springer (2017)

[58] Erway, C.C., Küpçü, A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) CCS 2009. pp. 213–222. ACM (2009)

[59] Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. J. Cryptol. **9**(1), 35–67 (1996)

[60] Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Ortiz, H. (ed.) STOC 1990. pp. 416–426. ACM (1990)

[61] Fiat, A.: Batch RSA. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 175–185. Springer (1989)

[62] Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer (1993)

[63] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer (1986)

[64] Fischlin, M., Fischlin, R.: Efficient non-malleable commitment schemes. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 413–431. Springer (2000)

[65] Fischlin, M., Onete, C.: Relaxed security notions for signatures of knowledge. In: López, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 309–326 (2011)

[66] Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10992, pp. 33–62. Springer (2018)

[67] Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer (1984)

[68] Garg, S., Gupta, D.: Efficient round optimal blind signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 477–495. Springer (2014)

[69] Garg, S., Hajiabadi, M., Mahmoody, M., Rahimi, A.: Registration-based encryption: Removing private-key generator from IBE. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018. LNCS, vol. 11239, pp. 689–718. Springer (2018)

[70] Gennaro, R., Halevi, S., Rabin, T.: Secure hash-and-sign signatures without the random oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer (1999)

[71] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) STOC 2009. pp. 169–178. ACM (2009)

[72] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) STOC, 2008. pp. 197–206. ACM (2008)

[73] Gentry, C., Ramzan, Z.: Identity-based aggregate signatures. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 257–273. Springer (2006)

[74] Goh, E., Jarecki, S.: A signature scheme as secure as the diffie-hellman problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 401–415. Springer (2003)

[75] Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Jr., B.S.K. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer (1997)

[76] Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) STOC 2013. pp. 555–564. ACM (2013)

[77] Goldwasser, S., Ostrovsky, R.: Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 228–245. Springer (1992)

[78] Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) STOC 2013. pp. 545–554. ACM (2013)

[79] Goyal, R., Vaikuntanathan, V.: Locally verifiable signature and key aggregation. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13508, pp. 761–791. Springer (2022)

[80] Goyal, V.: Reducing trust in the PKG in identity based cryptosystems. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 430–447. Springer (2007)

[81] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) CCS 2006. pp. 89–98. ACM (2006)

[82] Goyal, V., Polychroniadou, A., Song, Y.: Sharing transformation and dishonest majority MPC with packed secret sharing. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13510, pp. 3–32. Springer (2022)

[83] Goyal, V., Song, Y., Srinivasan, A.: Traceable secret sharing and applications. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12827, pp. 718–747. Springer (2021)

[84] Guo, F., Chen, R., Susilo, W., Lai, J., Yang, G., Mu, Y.: Optimal security reductions for unique signatures: Bypassing impossibilities with a counterexample. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 517–547. Springer (2017)

[85] Guo, F., Mu, Y., Chen, Z.: Identity-based online/offline encryption. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 247–261. Springer (2008)

[86] Haber, S., Pinkas, B.: Securely combining public-key cryptosystems. In: Reiter, M.K., Samarati, P. (eds.) CCS 2001. pp. 215–224. ACM (2001)

[87] Hanzlik, L.: Non-interactive blind signatures for random messages. IACR Cryptol. ePrint Arch. p. 388 (2023)

[88] Hartung, G., Kaidel, B., Koch, A., Koch, J., Rupp, A.: Fault-tolerant aggregate signatures. In: Cheng, C., Chung, K., Persiano, G., Yang, B. (eds.) PKC 2016. LNCS, vol. 9614, pp. 331–356. Springer (2016)

[89] Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer (2010)

[90] Hofheinz, D., Jager, T., Kiltz, E.: Short signatures from weaker assumptions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 647–666. Springer (2011)

[91] Hohenberger, S., Koppula, V., Waters, B.: Chosen ciphertext security from injective trapdoor functions. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12170, pp. 836–866. Springer (2020)

[92] Hohenberger, S., Waters, B.: Realizing hash-and-sign signatures under standard assumptions. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 333–350. Springer (2009)

[93] Hohenberger, S., Waters, B.: Online/offline attribute-based encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 293–310. Springer (2014)

[94] Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer (2002)

[95] Ishida, A., Emura, K., Hanaoka, G., Sakai, Y., Tanaka, K.: Disavowable public key encryption with non-interactive opening. In: Bao, F., Miller, S., Zhou, J., Ahn, G. (eds.) ASIA CCS 2015. p. 667. ACM (2015)

[96] Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer (1996)

[97] Johnson, D., Menezes, A., Vanstone, S.A.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Sec. **1**(1), 36–63 (2001)

[98] Kang, B.G., Park, J.H., Hahn, S.G.: A certificate-based signature scheme. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 99–111. Springer (2004)

[99] Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer (2004)

[100] Kiayias, A., Tsiounis, Y., Yung, M.: Group encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 181–199. Springer (2007)

[101] Komano, Y., Ohta, K., Shimbo, A., Kawamura, S.: Toward the fair anonymous signatures: Deniable ring signatures. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 174–191. Springer (2006)

[102] Kurosawa, K., Heng, S.: 3-move undeniable signature scheme. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 181–197. Springer (2005)

[103] Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-based group signature scheme with verifier-local revocation. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 345–361. Springer (2014)

[104] Lewko, A.B., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer (2011)

[105] Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 1–31. Springer (2016)

[106] Libert, B., Quisquater, J.: Identity based undeniable signatures. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 112–125. Springer (2004)

[107] Libert, B., Yung, M., Joye, M., Peters, T.: Traceable group encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 592–610. Springer (2014)

[108] Lim, C.H., Lee, P.J.: Server (prover/signer)-aided verification of identity proofs and signatures. In: Guillou, L.C., Quisquater, J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 64–78. Springer (1995)

[109] Liu, J.K., Chu, C., Zhou, J.: Identity-based server-aided decryption. In: Parampalli, U., Hawkes, P. (eds.) ACISP 2011. LNCS, vol. 6812, pp. 337–352. Springer (2011)

[110] Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 325–335. Springer (2004)

[111] Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 74–90. Springer (2004)

[112] Malkin, T., Micciancio, D., Miner, S.K.: Efficient generic forward-secure signatures with an unbounded number of time periods. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 400–417. Springer (2002)

[113] Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: Gong, L., Stearn, J. (eds.) CCS 1996. pp. 48–57. ACM (1996)

[114] Matsumoto, T., Kato, K., Imai, H.: Speeding up secret computations with insecure auxiliary devices. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 497–506. Springer (1988)

[115] Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer (1989)

[116] Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: FOCS 1999. pp. 120–130. IEEE Computer Society (1999)

[117] Micali, S., Shamir, A.: An improvement of the fiat-shamir identification and signature scheme. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 244–247. Springer (1988)

[118] Nakanishi, T., Fujii, H., Hira, Y., Funabiki, N.: Revocable group signature schemes with constant costs for signing and verifying. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 463–480. Springer (2009)

[119] Nguyen, K., Guo, F., Susilo, W., Yang, G.: Multimodal private signatures. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13508, pp. 792–822. Springer (2022)

[120] Nguyen, L., Safavi-Naini, R.: Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 372–386. Springer (2004)

[121] Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer (2009)

[122] Okamoto, T., Takashima, K.: Decentralized attribute-based signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 125–142. Springer (2013)

[123] Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In: Roy, B.K. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 1–20. Springer (2005)

[124] Pastuszak, J., Michatek, D., Pieprzyk, J., Seberry, J.: Identification of bad signatures in batches. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 28–45. Springer (2000)

[125] Pereira, H.V.L.: Bootstrapping fully homomorphic encryption over the integers in less than one second. In: Garay, J.A. (ed.) PKC 2021. LNCS, vol. 12710, pp. 331–359. Springer (2021)

[126] Poettering, B., Stebila, D.: Double-authentication-preventing signatures. In: Kutylowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 436–453. Springer (2014)

[127] Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)

[128] Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer (2001)

[129] Scafuro, A.: Break-glass encryption. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11443, pp. 34–62. Springer (2019)

[130] Scafuro, A., Siniscalchi, L., Visconti, I.: Publicly verifiable proofs from blockchains. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11442, pp. 374–401. Springer (2019)

[131] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer (1984)

[132] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer (1997)

[133] Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J.: Universal designated-verifier signatures. In: Laih, C. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 523–542. Springer (2003)

[134] Steinfeld, R., Bull, L., Zheng, Y.: Content extraction signatures. In: Kim, K. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 285–304. Springer (2001)

[135] Teranishi, I., Furukawa, J., Sako, K.: k-times anonymous authentication (extended abstract). In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 308–322. Springer (2004)

[136] Trolin, M., Wikström, D.: Hierarchical group signatures. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 446–458. Springer (2005)

[137] Tsabary, R.: Candidate witness encryption from lattice techniques. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13507, pp. 535–559. Springer (2022)

[138] Tzeng, W.: A practical and secure-fault-tolerant conferenc-key agreement protocol. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 1–13. Springer (2000)

[139] Tzeng, W., Tzeng, Z.: A public-key traitor tracing scheme with revocation using dynamic shares. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 207–224. Springer (2001)

[140] Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer (2005)

[141] Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In: Bao, F., Deng, R.H., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer (2004)

[142] Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: Short signatures and ibes with small key sizes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 303–332. Springer (2016)

[143] Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) $<<$ cost(signature) + cost(encryption). In: Jr., B.S.K. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer (1997)