Budapest University of Technology and Economics
Department of Networked Systems and Services

Cryptographic Problems in the Context of Data Markets

Ph.D. Dissertation
of
Máté Horváth

Supervisor: Levente Buttyán, Ph.D.

www.crysys.hu

Budapest, Hungary
2020

Abstract

The Internet of Things (IoT) provides us with a vast amount of new data day by day; however, currently, most of these are only stored without utilizing their full potential. The attractive concept of data markets can change this situation in the near future. Exchanging data, however, is always accompanied by privacy and security concerns. Being a broad concept, data markets can pose various challenges depending on the exact scope, the demands, goals, and opportunities of the actors, most notably on the trust or distrust between them. To better understand the challenges, we provide a general system model for data markets and use it to identify and structure the emerging scenarios and connect them to the relevant areas of cryptography. This problem domain analysis serves as the starting point of the rest of the dissertation. Our analysis highlights open problems, motivating further research on specific cryptographic primitives, and we address three of these. The problems and solutions we propose are not exclusively tied to data market security but can also be applied in secure computation and secure cloud storage.

Private Function Evaluation (PFE) enables two parties to jointly execute a computation such that one of them provides the input while the other chooses the function to compute. According to the standard security requirements, a PFE protocol should leak no more information, neither about the function nor the input, than what is revealed by the computation's output. Existing PFE protocols inherently restrict the scope of computable functions to a particular function class with given output size, thus ruling out the direct evaluation of such problematic functions as the identity map, which would entirely undermine the input privacy requirement. We observe that when not only the input x is confidential but certain partial information $g(x)$ of it as well, standard PFE fails to provide meaningful input privacy if g and the function f to be computed fall into the same function class. We investigate whether it is possible to simultaneously achieve a reasonable level of input and function privacy, even in the above cases. We propose the notion of Controlled PFE (CPFE) with different flavours of security and answer the question affirmatively by showing simple, generic realizations of the new notions. Our main construction, based on functional encryption (FE), also enjoys strong reusability properties enabling, e.g. fast computation of the same function on different inputs. To demonstrate the applicability of our approach, we show a concrete instantiation of the FE-based protocol for inner product computation that enables secure statistical analysis (and more) under the standard Decisional Diffie–Hellman assumption.

The other topics, we cover are connected to techniques that permit both secure storage of sensitive data and flexible data management at the same time.

On the one hand, we aim to make attribute-based encryption (ABE) more suitable for access control to data stored in the cloud. For this purpose, we concentrate on giving the encryptor full control over the access rights, providing feasible key management even in case of multiple independent authorities, and enabling viable user revocation, which is essential in practice. To do so, we extend the decentralized ciphertext-policy attribute-based encryption (CP-ABE) scheme of Lewko and Waters [LW11] with identity-based user revocation. Our revocation system is made feasible by removing the computational burden of a revocation event from the cloud service provider at the expense of some permanent, yet acceptable overhead of the encryption and decryption algorithms executed by the users. Thus, the computation overhead is distributed over a potentially large number of users instead of putting it on a single party (e.g. a proxy server), which would quickly lead to a performance bottleneck. The formal security proof of our scheme is given in the generic bilinear group and random oracle models.

Another line of research that aims to achieve security and flexibility at the same time is the study of searchable symmetric encryption (SSE). Several outstanding constructions have achieved sublinear time keyword search in massive databases by using various data structures to store keywords and document identifiers. When considering SSE, we focus on certain scenarios in which search over the whole database is unnecessary and show that the otherwise inefficient sequential scan (in linear time) can be very practical. The efficiency gain is due to the fact that adding new entries to the database comes for free in this case, while updating a complex data structure without information leakage is rather complicated. To demonstrate the practicality of our approach, we build a simple SSE scheme based on bilinear pairings and prove its security against adaptive chosen-keyword attacks in the standard model under the widely used symmetric external Diffie–Hellman (SXDH) assumption.

Acknowledgement

This thesis would not have seen daylight without the help and support of many people for which I cannot be grateful enough.

I would like to express my gratitude to my supervisor, Levente Buttyán, for the confidence that he has put in me. I feel fortunate for having been his student because I could learn from him much more than the ins and outs of the academic profession. I want to thank all my past and present colleagues in the CrySyS Lab not only for the pleasant atmosphere but also for that I could learn something from all of you. It was a pleasure to work in this team. Special thanks go to our amazing “infra team”,who saved me so much time when I struggled with technical problems. I am thankful to Péter Ligeti, László Csirmaz, Gábor Tardos, Viktória Villányi, and Gyula O.H. Katona, who were the core presenters and audience of the crypto seminars held at Rényi Institute. They introduced me new areas of cryptography and shared my enthusiasm whenever I talked about topics that are closer to me. I am grateful to Péter Antal, Yves Moreau, Fréderik Vercauteren, and István Csonta, who helped my research visit at KU Leuven, which was the most inspiring experience of my Ph.D. I am grateful to all the people with whom I could cooperate on papers or other projects, in particular, István Vajda, Levente Buttyán, Gergely Biczók, Gergely Ács, Gábor Székely, Dóra Neubrandt, Ádám Arany, Jaak Simm, Edward De Brouwer, Yves Moreau, Ilaria Chillotti, Charlotte Bonte, Fre Vercauteren, András Gézsi, András Millinghoffer. Special thanks go to Ilaria Chillotti with whom I have learned that with high probability, failed attempts are the best source of understanding. Many thanks to István András Seres for the many illuminating discussions. Thanks should also go to all the anonymous reviewers of my papers who spent their precious time improving my works.

I am grateful to my parents and family for their continuous support, even in my craziest endeavours, such as the one that led me to write this dissertation. I am also grateful to Örs Rebák for pointing out so many mistakes in my works, including this one. Thanks also go to my friends who did not let me sink into the oceans of research papers and, from time to time, grabbed my hand and forced me to take a fresh breath. The completion of my dissertation would not have been possible without my former teachers, who not only guided me during my studies but also taught me to find my way without a guide. I would like to thank them all.

I am most grateful to my friend and wife, Judit, who tolerated my often time-consuming hobby and did her best to support me.

Finally, I would also like to acknowledge the financial support of the following research agencies, grants, universities for their kind support that allowed me to focus on my work:

- National Research, Development and Innovation Office (NKFIH) of Hungary,[1][2]
- Csibi Sándor Grant
- CELSA Research Fund
- MELLODDY[3]
- Stipends and scholarships of the HEAT Summer School, BIU Winter School on Cryptography, Aarhus University, Summer School on real-world crypto and privacy are kindly appreciated.

[1] The research presented in this work was supported by the National Research, Development and Innovation Office (NKFIH) of Hungary under Grant Contract No. 116675 (K).

[2] The presented work was carried out partly within the SETIT Project (2018-1.2.1-NKP-2018-00004). Project no. 2018-1.2.1-NKP-2018-00004 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme.

[3] This project has received funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement N° 831472. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and EFPIA.

Contents

List of Figures

List of Tables

List of Abbreviations

2PC	secure two-party computation
ABE	attribute-based encryption
AES	advanced encryption standard
CA	central authority
CKA2	adaptive chosen keyword attack
CP-ABE	ciphertext-policy attribute-based encryption
CPFE	controlled private function evaluation
DB	data broker
DDH	decisional Diffie–Hellman problem
Dlog	discrete logarithm problem
DO	data owner
DP	differential privacy
EU-CMA	existential unforgeability under chosen message attack
FE	functional encryption
FHE	fully homomorphic encryption
GGM	generic group model
GID	global identifier
HE	homomorphic encryption
IND-CKA2	indistinguishability under adaptive chosen keyword attack
IND-CPA	indistinguishability under chosen plaintext attack
IoT	internet of things
KP-ABE	key-policy attribute-based encryption
LSSS	linear secret sharing scheme
MAC	message authentication code
ML	machine learning
MPC	secure multi-party computation
OT	oblivious transfer
OTP	one-time pad
PEKS	public key encryption with keyword search
PFE	private function evaluation
PIR	private information retrieval
PPT	probabilistic polynomial time

PRF	pseudo-random function
rCPFE	relaxed controlled private function evaluation
RL	revocation list
ROM	random oracle model
SFE	secure function evaluation
SSE	searchable symmetric-key encryption
SXDH	symmetric external Diffie–Hellman assumption
TEE	trusted execution environment
TTP	trusted third party
VAS	value added service
VASP	value added service provider

Chapter 1

Introduction

The past decades show that our everyday life is going through a continuous paradigm shift: from physical to virtual. The past years demonstrates that the pace of this change is speeding up. One of the biggest challenges of this phenomenon is that in the virtual world, we have to take care of things that we have taken for granted earlier in the physical world. A prominent example of this is trust. When Alice walked into a fashion store, spent there an hour finding her favourite dress, paid with cash, and finally went home cheerfully, for sure she did not have to worry that someone is stealing her last month salary, spying on her to see which dresses she liked and which she did not, which size is fitting on her, or where she was coming from and going to. To do these misdeeds in the physical world requires determination and significant effort, so people usually trust each other not doing them. However, the same activity in the virtual setting (e.g. in an online store) is much more "dangerous" as an entire industry is built around spying on Alice and her fellows to collect data about them [EN16], which turned out to be extremely valuable. The result is that trust cannot be evident anymore.

According to the neat definition of Boaz Barak, "cryptography is about replacing trust with mathematics" [Bar16]. This observation highlights the increasing role of cryptography in our everyday life. Existing cryptographic methods find newer and newer applications whenever a new area develops in the virtual world, or a new cryptographic challenge appears when there are no ready to use solutions or their adoption is non-trivial.

This dissertation studies the emerging concept of data markets and its connections to cryptography. The growing importance of data is beyond question today. According to an EU report [CMM+20] "the value of the data economy, which measures the overall impacts of the data market on the economy as a whole, exceeded the threshold of €400 billion in 2019 for the EU27 plus the United Kingdom, with a growth of 7.6% over the previous year." In spite of the large numbers, data economy today mainly involves companies and the end users remain one of the main sources of data collection instead of having real role in the market. Current technological trends, such as the proliferation of smart devices and the internet of things (IoT), can change this situation as the rapidly increasing amount of data is waiting for utilization. The main barrier of this is that in

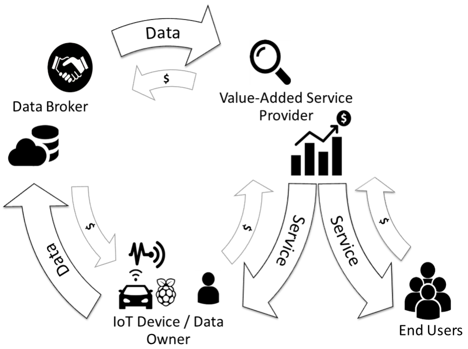


Figure 1.1: Centralized data market model.

most cases the collected data is only available for the user and manufacturer of a sensor or smart device. One possible way of exploiting the full potential of this information is to build an ecosystem around it. This is exactly the idea of data markets [OLJ+19], the basic concept of which is depicted in Fig. 1.1. Here a data broker buys data from the owners and resell the collected data (possibly together with computing resources) to a third party that provides some value added service (VAS) to its users. These services can typically help predictions or support optimization via analysing a wide range of data. While there are several approaches to realize data markets in practice [Oce, Dat17, IOT, Dat], we turn our attention towards the research questions that the security of these markets pose.

Contributions. As a first step, in §3 we provide a general system model for data markets and relying on this, we analyse data markets based on the possible security requirements of the different participants. We provide a problem-domain structuring, in which we systematically identify the possible scenarios that are determined by the possible goals, trust relations, and requirements of the participants. Then these scenarios are connected to different areas of cryptography that also helps us to identify open problems in the area. We than improve upon the state of the art, in three different areas (see §3.4 about how the investigated problems are related to data markets).

We investigate problems where we have to face inherent barriers stemming from interests that essentially contradict with each other but should be satisfied concurrently. As one can not have a cake and eat it too, we are always looking for some trade-off that takes us closer to a satisfactory solution.

The first problem, which we detail in §4 is the conflict of interests of participants in private function evaluation (PFE) protocols. One of them aims to hide an input to a function that is the secret of the other one. Assuming that the function provider has access to the output, the maximum that we can hope for is that the protocol reveals no more information about the output that is already leaked by the output. Even if the entire input is not possible to leak in this way (output length is typically shorter than input length), when launching the protocol, the input provider cannot know what will be revealed about the input. We initiate the study of partial input information leakage in this context and propose the notion of controlled PFE and a relaxation of it. We show generic realizations of these new notions and demonstrate the applicability of the protocol fulfilling the relaxed requirements by implementing it for the inner product functionality (see more details on our contribution in §4.1.1).

Next, in §5 we deal with fine grained access control to encrypted data, more precisely access right revocation in the context of attribute-based encryption (ABE). The challenge of user revocation in ABE schemes comes from their extreme flexibility in user identification. Concretely, it is very easy to identify various user groups when describing individuals with their non-unique attributes. However, when using an attribute-based description of users it becomes very hard a to efficiently identify a single user, who may share his or her attributes with multiple other users. It follows that achieving both flexible access control and efficient user revocation in the same system requires compromise. To this end, we propose a revocable multi-authority ciphertext-policy attribute-based encryption (CP-ABE) scheme and prove its security (for the details of this contribution, we refer to §5.1.1).

Finally, §6 is dedicated to the investigation of the inherent tension that appears during the design of searchable encryption. The problem is that in order to facilitate efficient search (i.e. breaking the $O(n)$ search complexity barrier) one has to introduce some structure in the encrypted database. However, in case of an update, it requires special care to not leak information when the new entry is inserted in the structure. This care, typically incurs significant efficiency loss. In this context, we revisit the so-called "forward index" approach and show its relevance when updates are frequent and searching in parts of the database is enough. For the details of our related contribution, see §6.1.1.

The possible applications of our results, beyond being candidate building blocks (or steps towards such blocks) of secure data markets, are discussed in §4.1.2, §5.1.2, and §6.1.2 respectively. We discuss the necessary background for our results in §2 and conclude our work in §7.

Chapter 2

Background

In this chapter, we provide the necessary background for the rest of the thesis. While we assume some familiarity with the basic concepts of cryptography, we introduce the different primitives and tools that we are going to rely on. While most of the chapter is dedicated to the introduction of details that we are going to use, there is one exception. In §2.1, we provide a non-technical overview of cryptographic primitives and concepts which we will link to various problems that may arise in the context of data markets.

2.1 Cryptographic Toolbox – A High-Level Overview

Before going into details related to our results, we invite the reader to take one step back for an overview of cryptographic tools that enable different forms of secure computation (see Table 2.1) and cryptographic access control to data. Our overview is not exhaustive. Its goal is to provide an intuition of the capabilities of the different tools which can serve as a basis for the problem domain structuring presented in §3. In the subsequent sections of the chapter, we provide further details about those primitives that are related to our results.

	Obfuscation	**FHE**	**FE**	**MPC**	**PFE**
Input	Plaintext	Ciphertext	Ciphertext	Ciphertext	Ciphertext
Output	Plaintext	Ciphertext	Plaintext	Plaintext	Plaintext
No. of possible evaluations	Any	Any	Any	1	1
Is the function public?	No	Yes	Usually yes	Yes	No

Table 2.1: Comparison of different concepts in cryptography that aim to secure computation without relying on trusted hardware.

Trusted execution environment (TEE) refers to an isolated processing environment in which computations can be securely executed irrespective of the rest of the system. At a high level, when considering information leakage, outsourced computation that is executed in TEE (by an untrusted party) is equivalent to local computation and thus in the rest of this work we do not differentiate between these two cases. For more details on TEE, see [SAB15].

Differential privacy (DP) is a mathematical framework for the privacy protection of data used for statistical analysis (see [NSW+17, DP20]). Its informal goal is to make available as much information about a dataset as possible while revealing as little as possible about the individual entries. While it is an important tool for privacy protection in data markets as well [Zhe20], we will not consider this solution because of two reasons. First, we are interested in enforcing access control policies belonging to data, while DP does not restrict access to data. We do not restrict the scope of computable functions to statistics (i.e. individual data values may be of interest instead of only cumulated data).

Homomorphic encryption (HE) enables to execute computations on encrypted data that results in an encrypted output. This capability can be used to make outsourced computation secure, by maintaining the secrecy of inputs and outputs (while the applied function is not concealed). Depending on the supported functions, there exist additively, multiplicatively, somewhat and fully homomorphic encryption schemes. While the latter one supports an arbitrary number of multiplications and additions, in somewhat homomorphic schemes the number of computable multiplications is restricted. See details in [ABC+15].

Functional encryption (FE) is a generalization of traditional (either private- or public-key) encryption schemes that integrates function evaluation into decryption. More precisely, it enables the generation of a so-called functional secret key corresponding to a function f. Feeding the functional key and a ciphertext corresponding to a value x to the decryption algorithm, results in the value $f(x)$ without leaking any more information about x, than exposed by $f(x)$. For details, we refer to §2.5.3.

Attribute-based encryption (ABE) is a subtype of FE, that realizes fine-grained access control. Ciphertexts and secret keys are associated with access control policies and “attributes” (or vice versa) and decryption is possible only if the attributes satisfy the policy. In §5, we elaborate more on this primitive.

Searchable symmetric-key encryption (SSE) and public key encryption with keyword search (PEKS) represent another subtype of FE [BHJP14] in the symmetric- and public-key settings, respectively. In this case, the function that is evaluated on a ciphertext together with decryption is an equality test between the plaintext underlying the ciphertext and the value embedded in the key (a.k.a. trapdoor). This functionality enables keyword search over encrypted data. In §6, we further investigate SSE, focusing on its efficiency.

Oblivious transfer (OT) and Private information retrieval (PIR) are dealing with transmission of data between two parties, such that the receiver does not have to reveal the data, she wants to retrieve from a dataset (PIR). We call this OT when the receiver obtains no other information, besides the desired data, from running the protocol. For more details about OT and PIR, see [OS07].

Secure multi-party computation (MPC) allow n parties to jointly compute an n variate function on their inputs, often through several rounds of interaction, without revealing their inputs to each other. Among those cryptographic primitives that aim to achieve some form of secure computation, realizations of MPC are the closest to practical usability [LP09].

Obfuscation is a program transformation that preserves the functionality but alters the description and operation such that the inner workings of the program remains hidden at the expense of some efficiency loss. In other words, the inputs and outputs of an obfuscated program are plaintexts but the computed function is hidden. In spite of several breakthrough results in the past years, obfuscation is still an exotic area within cryptography with several open questions (see [B1]).

2.2 Security Models

Specifying the security guarantees that a scheme can offer is always a crucial task in cryptography, as the value of a construction is necessarily tied to its security. Consequently, one has to be careful when modelling the environment where the scheme is used.

Standard model. In the so-called standard, or plain, model, we assume that the adversary is limited only by the available amount of time and computational power. These limitations helps us to characterize security through the use of assumptions. In our rsults, we are going to use both *generic* and *concrete* assumptions. The generic ones assume that a certain cryptographic primitive (e.g. MAC, see §2.5.1) exists; this becomes meaningful if the primitive can be instantiated based on a concrete assumption that is a reasonable mathematical conjecture (such as the hardness of factoring integers). This relation also implies that a generic assumption is better in the sense that it might allow various instantiations, but at the same time it is at most "as good as the concrete assumptions it can be based on" [GK16].

Idealized models. Often it turns out to be hard to prove security in the standard model. In this case, common way of facilitating proofs is via idealizing the adversary. In other words, one can separate attacks based on the properties utilized, and assume that an adversary cannot make use of certain properties. In this way, it is enough to prove security against the corresponding restricted adversaries. For instance, the widely used random oracle model (ROM) [BR93] considers ideal adversaries that interact with a "random oracle" instead of evaluating a concrete hash function. This separates attacks on a concrete scheme, by neglecting the ones that utilize the weaknesses of hash functions. This form of idealization both gives a persuasive intuition about security and helps to better understand the nature of the weaknesses that are still possible, because these must be caused by differences between the imagined and the real attacker.

Besides the ROM, we are going to make use of another idealized model, the so-called generic group model (GGM) of [Nec94, Sho97, Mau05], which considers elliptic-curve-group-based cryptosystems. The main assumption of the GGM is that any attack is independent of the specific structure of the group in which a scheme is instantiated. The model captures such generic attacks by substituting the concrete elements and operation with access to a "group oracle" that has two tasks. It can be queried for group elements, and for any i it answers (consistently) with a generic representation $\psi(i)$, called a "handle", that is a random bit string (instead of $g^i \in \mathbb{G}$ in any specific group). In order to execute the group operation, the oracle also has to be queried, and it replies with the handle of the output if the input handles are valid.

Finally, we have to mention the criticism of these models. It has been shown that there exist (rather contrived) schemes that are provably secure in the ROM or in the GGM, but for which any implementation of the oracle leads to insecure schemes in the standard model (see [CGH04] and [Den02], respectively). Note, however, that this does not mean that a security proof in these models entails real-world vulnerabilities [KM07, KM15].

2.3 On Bilinear Pairings and the Used Assumptions

As we are going to make extensive use of groups, where the discrete logarithm problem (Dlog) is believed to be hard. We present further assumptions that we are going to use and also introduce the most important facts related to groups with efficiently computable bilinear maps. We start with the standard decisional Diffie–Hellman problem (DDH).

Assumption 1 (DDH)**.** *Let $\mathbb{G}$ be a multiplicative group of prime order p, $g \in \mathbb{G}$ its generator element and $x, y \in \mathbb{Z}_p^*$ uniformly random values. We say that the Decisional Diffie–Hellman assumption holds in $\mathbb{G}$ if given (g, g^x, g^y, g^r), no probabilistic polynomial time (PPT) algorithm can decide, with higher than $\frac{1}{2}+\mathsf{negl}(p)$ probability, whether $r = xy$ or r is also a uniformly random value from $\mathbb{Z}_p^*$.*

Next, we turn our attention towards bilinear maps. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three multiplicative cyclic groups of prime order p. Let g_1 and g_2 be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear map (pairing), with the following properties:

1. Bilinearity: $\forall u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$
2. Non-degeneracy: $e(g_1, g_2) \neq 1$.

We say that $\mathcal{G} = \{p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e\}$ is a bilinear instance if all the group operations and the bilinear map e are efficiently computable. A pairing is called symmetric (or Type-1) if $\mathbb{G}_1 \neq \mathbb{G}_2$. In case of assymetryc (a.k.a Type-3) pairings, $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphism exists between them. We note that based on both its efficiency and security, Type-3 pairing type is considered to be the ideal choice when instantiating a cryptosystem [CM11]. At the same time, usually symmetric pairing-based cryptosystems can also be transformed to use Type-3 pairings [CHM10]. In this work, we apply both symmetric (in §5) and Type-3 pairings (in §6).

Over asymmetric bilinear instances, we will use the popular symmetric external Diffie–Hellman assumption (SXDH) that we define next. Informally speaking, the SXDH assumption states that no efficient algorithm can solve the DDH problem either in $\mathbb{G}_1$ or in $\mathbb{G}_2$ of a bilinear instance.

Assumption 2 (SXDH)**.** *Let $g_i \in \mathbb{G}_i$ be a generator element of the group and $a_i, b_i \in \mathbb{Z}_p^*$ are uniformly random values for $i = 1, 2$. We say that the symmetric external Diffie–Hellman assumption (SXDH) holds in a bilinear instance $\mathcal{G}$ if given*

$$(g_i, g_i^{a_i}, g_i^{b_i}, g_i^{R_i}),$$

for $i = 1, 2$, no polynomial time algorithm can decide whether $R_i = a_i b_i$ or R_i is also a uniformly random value from $\mathbb{Z}_p^*$.

2.4 Access Structures and Linear Secret Sharing

We are going to make use of access structures (for the formal definition we refer to [Bei96]) in the context of fine-grained access control to encrypted data (see §5). In our case the access policies of the encryptor will determine the set of decryptors through an access structure. Because of efficiency reasons, we restrict our attention to *monotone access structures*, meaning that any superset of an authorized set is authorized as well. We note that (inefficiently) general access structures can also be realized by having the not of each attribute as separate attribute. To enforce an access structure, determined by the encryptor, we are going to make essential use of linear secret sharing schemes (LSSSs). Here we adopt the definitions from those given in [Bei96].

Definition 1 (Linear Secret Sharing Scheme [Bei96])**.** *A secret-sharing scheme* Π *over a set of attributes U is called linear (over $\mathbb{Z}_p$) if*

1. *the shares for each attribute form a vector over $\mathbb{Z}_p$,*

2. *there exists a matrix A with ℓ rows and n columns called the share-generating matrix for* Π*. For all $x = 1, \ldots, \ell$, the x^{th} row of A is labelled by an attribute $\rho(x)$, where ρ is a function from $\{1, \ldots, \ell\}$ to U. When we consider the column vector $v = (s; r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $Av = \lambda$ is the vector of ℓ shares of the secret s according to* Π*. The share $(Av)_x = \lambda_x$ belongs to attribute $\rho(x)$.*

In [Bei96] it is shown that every linear secret sharing-scheme according to the above definition also enjoys the *linear reconstruction property*, defined as follows. Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let

$I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Furthermore, it is also shown in [Bei96] that these constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix A and for unauthorized sets, no such $\{\omega_i\}$ constants exist.

We use the convention that $(1, 0, 0, \ldots, 0)$ is the "target" vector for any linear secret sharing scheme. For any satisfying set of rows I in A, we will have that the target vector is in the span of I, but for any unauthorized set, it is not.

Using standard techniques (see [LW11, Appendix G]) one can convert any monotonic boolean formula into an LSSS representation. An access tree of ℓ nodes will result in an LSSS matrix of ℓ rows.

2.5 The Utilized Cryptographic Primitives

2.5.1 Message Authentication Codes

We are going to make use of deterministic MACs with a specific syntax to construct randomized MACs for our purposes.

Definition 2. *A deterministic (*randomized*) MAC consists of the following algorithms:*

MAC.KeyGen$(\lambda) \rightarrow \mathsf{sk}_{\mathbf{MAC}}$ *This randomized algorithm generates secret key* sk_{MAC} *based on security parameter* λ*.*

MAC$(\mathsf{sk}_{\mathbf{MAC}}, m) \rightarrow \tau$ *Using the secret key* sk_{MAC}*, this deterministic (*randomized*) algorithm produces a tag* τ *for the input message* m*.*

MAC.Verify$(\mathsf{sk}_{\mathbf{MAC}}, m, \tau) \rightarrow \{0, 1\}$ *With the help of the secret key* sk_{MAC} *this deterministic algorithm checks whether* τ *was prepared using* m *or not.*

Note that in case of deterministic MACs, **MAC.Verify**$(\mathsf{sk}_{\mathrm{MAC}}, m, \tau)$ simply computes **MAC**$(\mathsf{sk}_{\mathrm{MAC}}, m) = \tau'$ and checks whether $\tau' = \tau$ or not.

In this work, we are interested in MACs in special form, i.e. we assume that tag $\tau = \alpha^{F(\mathsf{sk}_{\mathrm{MAC}}, m)}$, where α is a generator element of either group $\mathbb{G}_1$ or $\mathbb{G}_2$ of the pairing group $\mathcal{G}$ and F is some function of the secret key and the message to be authenticated. In the literature, several pseudo-random functions (PRFs) were described [NR97, LW09, BMR10, ABP15] in the desired form under various hardness assumptions. However, for our purposes this stronger guarantee of pseudo-randomness is not required, any of these can serve as proper MAC functions satisfying the following requirement of existential unforgeability under chosen message attack (**EU-CMA**).

Definition 3 (**EU-CMA**)**.** *We say that a MAC function (**MAC.KeyGen**, **MAC**, **MAC.Verify**) is secure if it is existentially unforgeable under chosen message attack (**EU-CMA**), i.e. let* $\lambda \in \mathbb{N}$ *be a security parameter, and* $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_{q+1})$ *be an* *non-uniform adversary that has at most negligible advantage in the* **EU-CMA**$_{MAC,\mathcal{A}}(\lambda)$ *experiment (depicted on Fig. 2.1):*

$$\Pr(\textbf{EU-CMA}_{MAC,\mathcal{A}}(\lambda) = 1) \leq \mathsf{negl}(\lambda).$$

```
**EU-CMA**$_{\mathrm{MAC},\mathcal{A}}(\lambda)$ Security Game
$\mathsf{sk}_{\mathrm{MAC}} \leftarrow_{\$} \textbf{MAC.KeyGen}(1^{\lambda})$
$\{m_1, \mathsf{state}_{\mathcal{A}_1}\} \leftarrow \mathcal{A}_1(1^{\lambda})$
$(\tau_1, m_1) \leftarrow \textbf{MAC}(\mathsf{sk}_{\mathrm{MAC}}, m_1)$
for $i = 2, \ldots, q$
   $\{m_i, \mathsf{state}_{\mathcal{A}_i}\} \leftarrow \mathcal{A}_i\left(1^{\lambda}, \mathsf{state}_{\mathcal{A}_{i-1}}, \{\tau_j, m_j\}_{j\in[i-1]}\right)$
   $(\tau_i, m_i) \leftarrow \textbf{MAC}(\mathsf{sk}_{\mathrm{MAC}}, m_i)$
$(\bar{\tau}, \bar{m}) \leftarrow_{\$} \mathcal{A}_{q+1}\left(1^{\lambda}, \mathsf{state}_{\mathcal{A}_q}, \{(\tau_i, m_i)\}_{i\in[q]}\right)$
if $\textbf{MAC.Verify}(\mathsf{sk}_{\mathrm{MAC}}, \bar{m}, \bar{\tau}) = 1$ and $\bar{m} \notin \{m_1, \ldots, m_q\}$
**return** 1
```

Figure 2.1: **EU-CMA** security game for MACs.

For instance, the construction of [NR97] can be used assuming the DDH assumption holds.

2.5.2 Oblivious Transfer

Oblivious transfer (OT) is one of the most fundamental primitives in cryptography and a cornerstone of secure computation. It enables transferring data between two parties, the sender ($\mathcal{S}$) and the receiver ($\mathcal{R}$, a.k.a. chooser), in a way that protects both of them. $\mathcal{S}$ can be sure that $\mathcal{R}$ only obtains a subset of the sent messages, while $\mathcal{R}$ is assured that $\mathcal{S}$ does not know which messages he selected to reveal. In Fig. 2.2 the ideal functionality of k out of n OT [CT05] is represented that we are also going to rely on.

While being a public-key primitive, so-called OT-extension protocols enable rather efficient OT evaluation. To do so, the participants first pre-compute a limited number of "base-OTs" with certain inputs that are independent of their real inputs. Then using the obtained values, they can evaluate a much larger number of OTs by executing more efficient symmetric-key operations only. This kind of efficiency improvement automatically applies to our protocols in §4 after substituting plain OT, with OT-extension with the same functionality [KKRT16, RR17].

2.5.3 Functional Encryption

As we already introduced (see §2.1), FE is a generalized encryption scheme that enables certain computations on hidden data for authorized parties. Both public- and secret-key

The functionality is parametrized by two integers $k < n$, and two parties: a sender $\mathcal{S}$ and a receiver $\mathcal{R}$.

FUNCTIONALITY:
On input $m_1, \ldots, m_n$ messages from $\mathcal{S}$ and an index set $\{i_1, \ldots, i_k\} \subset [n]$ from $\mathcal{R}$

- $\mathcal{S}$ obtains no output,
- $\mathcal{R}$ receives $m_{i_1}, \ldots, m_{i_k}$, but nothing else.

Figure 2.2: Ideal functionality $\mathcal{F}_{OT^n_k}$ of k out of n OT.

variants are known, but here we limit ourselves to the secret-key setting that suffices for our purposes. An secret-key FE scheme consists of the following four algorithms.

$\mathsf{FE.Setup}(\lambda) \to (\mathsf{msk}_{\mathsf{FE}}, \mathsf{pp}_{\mathsf{FE}})$ Upon receiving a security parameter λ it produces the public system parameters $\mathsf{pp}_{\mathsf{FE}}$ and the master secret key $\mathsf{msk}_{\mathsf{FE}}$.

$\mathsf{FE.Enc}(\mathsf{msk}_{\mathsf{FE}}, x) \to \mathsf{ct}$ The encryption algorithm takes the master secret key $\mathsf{msk}_{\mathsf{FE}}$ and a message x and outputs a ciphertext ct.

$\mathsf{FE.KeyGen}(\mathsf{msk}_{\mathsf{FE}}, f) \to \mathsf{fsk}_f$ The key generation algorithm can be used to generate a functional secret key fsk_f for a function f with the help of the $\mathsf{msk}_{\mathsf{FE}}$.

$\mathsf{FE.Dec}(\mathsf{ct}, \mathsf{fsk}_f) \to y$ Having a functional secret key fsk_f (for function f) and a ciphertext ct (corresponding to x), the decryption outputs the value y.

The correctness of FE requires that if fsk_f and ct were indeed generated with the corresponding algorithms using inputs f and x respectively, then $y = f(x)$ must hold. Regarding security, in §4 we are going to use the non-adaptive simulation-based security definition of FE [GVW12]. We note that while the SIM security of FE is impossible to realize in general [BSW11], for several restricted – yet important – cases it is still achievable, e.g. when the number of functional keys are a priori bounded [GVW12], or when the computable function class is restricted [ALS16]. As our applications also use these restrictions, known FE impossibility results do not affect the way we use FE. We recall here the security definition[1] that we are going to rely on.

Definition 4 (q-NA-SIM and q-AD-SIM Security of FE [GVW12])**.** *Let $\mathcal{FE}$ be a functional encryption scheme for a circuit family $\mathcal{C} = \{\mathcal{C}_\nu : \mathcal{X}_\nu \to \mathcal{Y}_\nu\}_{\nu \in \mathbb{N}}$. For every PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ consider the following two experiments:*

[1] Definition 4 is taken verbatim from [GVW12].

$\mathsf{Exp}^{\mathsf{real}}_{\mathcal{FE},\mathcal{A}}(\lambda)$

```
$(\mathsf{pp_{FE}}, \mathsf{msk_{FE}}) \leftarrow\!\!\$\ \mathsf{FE.Setup}(\lambda)$
$(x, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}_1^{\mathsf{FE.KeyGen}(\mathsf{msk_{FE}},\cdot)}(\mathsf{pp_{FE}})$
$\mathsf{ct} \leftarrow\!\!\$\ \mathsf{FE.Enc}(\mathsf{pp_{FE}}, x)$
$\beta \leftarrow\!\!\$\ \mathcal{A}_2^{O(\mathsf{msk_{FE}},\cdot)}(\mathsf{pp_{FE}}, \mathsf{ct}, \mathsf{st})$
$\mathsf{output}(\beta, x)$
```

$\mathsf{Exp}^{\mathsf{ideal}}_{\mathcal{FE},\mathcal{S}}(\lambda)$

```
$(\mathsf{pp_{FE}}, \mathsf{msk_{FE}}) \leftarrow\!\!\$\ \mathsf{FE.Setup}(\lambda)$
$(x, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}_1^{\mathsf{FE.KeyGen}(\mathsf{msk_{FE}},\cdot)}(\mathsf{pp_{FE}})$
    – Let $(C_1, \ldots, C_q)$ be $\mathcal{A}_1$'s oracle queries
    – Let $\mathsf{fsk}_{f_i}$ be the oracle reply to $C_i$
    – Let $\mathcal{V} := \{y_i = C_i(x), C_i, \mathsf{fsk}_{f_i}\}$.
$(\mathsf{ct}, \mathsf{st}') \leftarrow\!\!\$\ \mathcal{S}_1(\mathsf{pp_{FE}}, \mathcal{V}, \lambda)$
$\beta \leftarrow\!\!\$\ \mathcal{A}_2^{O'(\mathsf{msk_{FE}},\mathsf{st}',\cdot)}(\mathsf{pp_{FE}}, \mathsf{ct}, \mathsf{st})$
$\mathsf{output}(\beta, x)$
```

We distinguish between two cases of the above experiment:

1. *The* adaptive *case, where:*

   - *the oracle* $O(\mathsf{msk_{FE}}, \cdot) = \mathsf{FE.KeyGen}(\mathsf{msk_{FE}}, \cdot)$ *and*
   - *the oracle* $O'(\mathsf{msk_{FE}}, \mathsf{st}', \cdot)$ *is the second stage of the simulator, namely* $\mathcal{S}_2^{U_x(\cdot)}(\mathsf{msk_{FE}}, \mathsf{st}', \cdot)$ *where* $U_x(C) = C(x)$ *for any* $C \in \mathcal{C}_\nu$.

   The simulator algorithm $\mathcal{S}_2$ *is stateful in that after each invocation, it updates the state* st' *which is carried over to its next invocation. We call a simulator algorithm* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ admissible *if, on each input* C, $\mathcal{S}_2$ *just makes a single query to its oracle* $U_x(\cdot)$ *on* C *itself.*
   The functional encryption scheme $\mathcal{FE}$ *is then said to be q-query-simulation-secure for one message against adaptive adversaries (q-AD-SIM secure for short) if there is an admissible PPT simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ *such that for every PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *that makes at most q queries, the following two distributions are computationally indistinguishable:*

   $$\left\{\mathsf{Exp}^{\mathsf{real}}_{\mathcal{FE},\mathcal{A}}(\lambda)\right\}_{\nu\in\mathbb{N}} \overset{c}{\approx} \left\{\mathsf{Exp}^{\mathsf{ideal}}_{\mathcal{FE},\mathcal{S}}(\lambda)\right\}_{\nu\in\mathbb{N}}$$

2. *The* non-adaptive *case, where the oracles* $O(\mathsf{msk_{FE}}, \cdot)$ *and* $O'(\mathsf{msk_{FE}}, \mathsf{st}, \cdot)$ *are both the "empty oracles" that return nothing: the functional encryption scheme* $\mathcal{FE}$ *is then said to be q-query-simulation-secure for one message against non-adaptive adversaries (q-NA-SIM secure, for short) if there is a PPT simulator* $\mathcal{S} = (\mathcal{S}_1, \perp)$ *such that for every PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *that makes at most q queries, the two distributions above are computationally indistinguishable.*

As shown by [GVW12, Theorem A.1.], in the non-adaptive setting (that we also use), q-NA-SIM security for one message is equivalent to q-NA-SIM security for many messages.

Chapter 3

Problem Domain Analysis

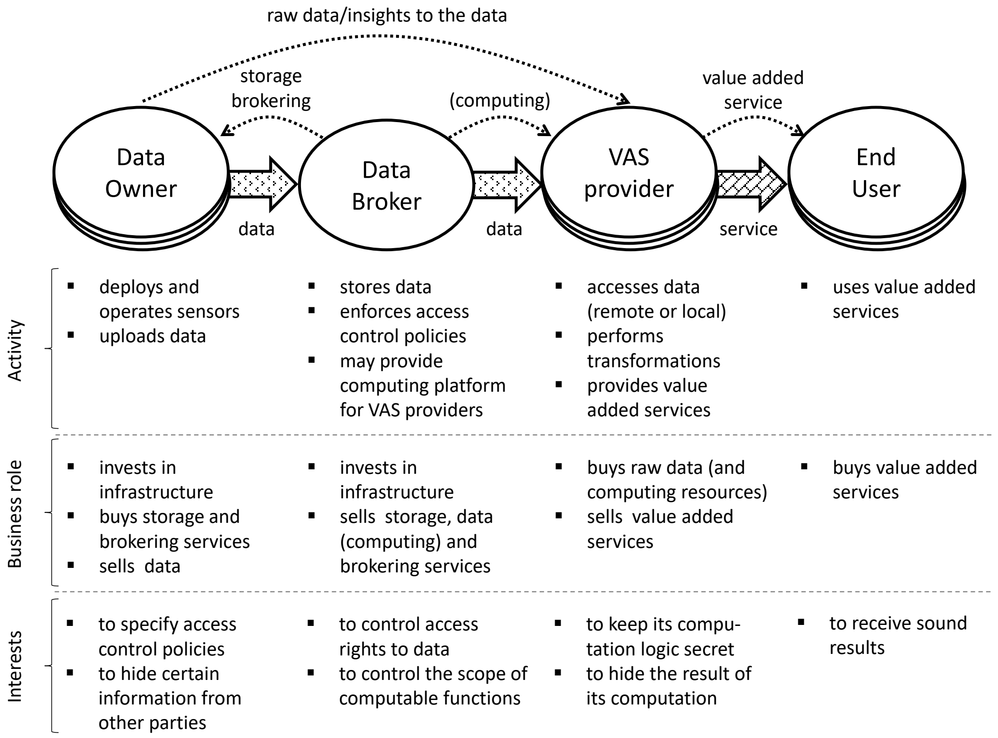
3.1 Introduction

The issue of security and privacy arises naturally whenever data get out of the control of its owner. In this chapter, we investigate the possible security issues related to data markets. For this, we propose a general system model for data markets. While there are several important aspects of trading in this model, we concentrate our attention on its main product, namely data. More precisely, we investigate the trust relations between the different participants of the market and explore the effect of these relations on how data has to be handled in order to satisfy the requirements of each party. Consequently, we neglect every other – even security – issue that is not directly concerned with data. Practically, we assume that every challenge regarding privacy-preserving trading is already solved (e.g. pricing [LLMS14], prevention of dishonest behaviour, enforcement of market regulations, incentivisation, etc.) except for how sold and bought information is passed from the owner to the customer under different circumstances.

In this context, we would like to answer the following questions:

Depending on the trust relations of the participants of a data market, what kind of security issues has to be addressed?
Which areas of cryptography can contribute to solve the challenges and what are the open problems that the different scenarios highlight?

Our system model in §3.2 shows that centralised data markets involve parties of three type: data owner (DO), data broker (DB), and value added service provider (VASP) which can all trust or distrust the others. Based on the level of trust between the parties, we identify 24 different types of interaction, which we call *scenarios* in the sequel. Some of these scenarios can be handled in a straightforward way, some are closely related to different areas of cryptography, whereas others motivate further research on specific problems. Our work also motivates the adjustment of existing cryptographic primitives to the use-cases provided by data markets.



	Data Owner	Data Broker	VAS provider	End User
Activity	▪ deploys and operates sensors ▪ uploads data	▪ stores data ▪ enforces access control policies ▪ may provide computing platform for VAS providers	▪ accesses data (remote or local) ▪ performs transformations ▪ provides value added services	▪ uses value added services
Business role	▪ invests in infrastructure ▪ buys storage and brokering services ▪ sells data	▪ invests in infrastructure ▪ sells storage, data (computing) and brokering services	▪ buys raw data (and computing resources) ▪ sells value added services	▪ buys value added services
Interests	▪ to specify access control policies ▪ to hide certain information from other parties	▪ to control access rights to data ▪ to control the scope of computable functions	▪ to keep its computation logic secret ▪ to hide the result of its computation	▪ to receive sound results

Figure 3.1: System model of the envisioned Data Market.

3.2 System Model for Data Markets

Without loss of generality, we assume that the resources of the data market are provided by owners of sensors and IoT devices (DO), who provide (sell) data directly to a DB and indirectly to VASPs. In our model, the DB is not decentralized but rather a single entity, investing in infrastructure for data storage and executing computations. Naturally, different DBs might compete with each other in the future and most probably also with different distributed systems but our focus is the inner working of a single, centralized marketplace with one DB. At the same time, we do not make any restrictions on communication, thus our model encompasses scenarios where data is only transferred between DOs and VASPs in a peer-to-peer fashion. On the contrary, storing data on a centralised server (as a DB) in encrypted form and transmitting only the keys between the other parties is equivalent to a decentralised system when regarding information leakage about the data.

The DBs offer their services to different VASPs that can utilize data in order to fulfil the various needs of end users (either individuals or companies). Our model does not restrict the scope of offered services (i.e. the functions computed) but we envision statistical analysis or the training of machine learning (ML) models as the foundation of typical value added services.

Even if the final goal of such ecosystems is to serve the end users, they are less important in our study as we are interested in the security aspects of data markets. More precisely, all the available information for an end user is a subset of the information handled by the corresponding VASP and vice versa, any information about the end user is exposed in the system through the VASP so for our purposes it is enough to consider VASPs.

The imagined model of the ecosystem is depicted in Fig. 3.1, containing also the rough business models of the different parties. At the same time, the economic aspects of such system are out of the scope of this work.

3.3 Problem Domain Analysis

We start our analysis with the investigation of the relations between the participants of the system introduced in §3.2. After identifying their possible goals regarding security (see also Fig. 3.1), we organize the emerging use-cases and identify the relevant branches of cryptography and also point out research directions.

3.3.1 Trust Relations of the Participants

The main goal of our study in [C3] was to identify the different scenarios that emerge when some of the actors in a data market does not trust all the other parties. In order to go through all possible cases and identify already solved and still open problems, we investigate the possible forms of trust between data owners, data broker and value-added service providers.

DO → DB If the DB is trusted by the DO, then it is straightforward to store the data as a plaintext. In this case, the enforcement of access control policy to the data can be outsourced to the DB. On the other hand, an untrusted DB should not store its clients' data in the clear but in *encrypted* form, together with the corresponding *metadata as cleartext*. Note that the latter one is necessary for providing the data brokering service. Access control of data becomes more challenging that can be solved, e.g. by providing a key exchange mechanism between DOs and VASPs.

DO → VASP While a trusted VASP may access plaintext data, access control might still be important as most probably only smaller sets of data are sold to the VASP and thus even in this case the VASP should have only *conditional access* to data. When the VASP is not even trusted by the DO, it is natural to expect that it has no direct access to any data and it is only allowed to get results of non-trivial computations on some specific data.

DB → DO Trusting DOs from the viewpoint of a DB is equivalent to *trusting the data source* i.e. the DB assumes that the received data is not fake, its source is the claimed source and the result of any measurement was not modified. Using algorithmic measures (e.g. by checking the consistency of the data) the lack of this confidence can be remedied only partially and thus this type of distrust is out of the scope of this work. At the same time, this problem can be addressed, e.g. by using pricing based on game theoretic considerations or by contracts.

DB → VASP As the DB is selling information to the VASP, pricing should scale with the amount of provided information that can be measured using two different metrics as well. The first is the available amount of data for the VASP that can be controlled enforcing some access control policy for the data. Note that in this regard, the interests of the DO and the DB coincides and thus we assume that either of them enforces a common access policy. The second possible way of measuring the sold information is through the scope of computable functions on the data that is available for a VASP. One natural way of restricting the computing capabilities of the VASP is via providing it with a restricted interface that naturally determines restrictions. However, we are interested in a more general setting, where such limitation is not present, especially because it leaves no room for the interests of the VASP (see the VASP → DB relation). Accordingly, we assume that arbitrary function descriptions are forwarded to the DB that are evaluated if they are in a class of "allowed functions" (for which the VASP has paid). Alternatively, if the computation of the functions is not outsourced to the DB, it should be solved that the data, sent to the VASP, is only useful for the computation of "allowed functions" and cannot be used as input for any other functions.

VASP → DO When purchasing data from DB, the VASPs naturally rely on the honesty of the DOs, however, the enforcement of honest behaviour is the duty of the DB according to the system model of the data market. Accordingly, this trust relationship is indirect and thus not investigated.

VASP → DB The business model of a VASP is built around the functions, that are evaluated on the data, bought from the DB. The function might also reflect private information of the end user of the service (e.g. in a location-based service, the query of the VASP can leak the location of its user). These highlights the importance of this asset and shows that VASPs are motivated to hide the computation logic they use, even if the computation is executed by the DB as a part of its services.

In case of so-called learnable functions, which can be reconstructed merely from input-output pairs, hiding these values is an important pillar of keeping the function private. Moreover, the output alone can also have business value as the end user pays for this to the VASP. When talking about the input data, it is important to differentiate the data value from the metadata. If the DB stores plaintext data, VASPs can only obscure the accessed data if both the accessed values and metadata remain hidden from the DB. When only encrypted data is available to the DB, typically metadata can help to feed the proper input to the function of the VASP but hiding both information can be the goal of a privacy-aware VASP who would prefer to hide its access pattern to data.

Depending on which of these four assets are intended to be hidden from the DB, 2^4 different scenarios can occur. For the ease of exposition, we denote a VASP's confidentiality goals with 4-tuple (F, I, I', O). Each variable correspond to a binary value, 0 meaning public and 1 denoting confidential. The variables represent the function to be computed (F), its input value(s) (I), metadata for the input(s) (I') and the output (O) of the computation. For example, $(1, 0, 0, 0)$ represents that the VASP only wants to hide the computational logic form the DB but not the input, metadata, and the output). Some of the resulting scenarios are contradictory so we ignore them in the rest of the work. These are the following.

- We assume that it does not make sense to hide the accessed input metadata from the DB whenever the input itself is not hidden. Or from a different perspective, hiding the accessed metadata from the DB implies that the used data values are hidden as well. Accordingly $(0, 0, 1, 0), (0, 0, 1, 1), (1, 0, 1, 0), (1, 0, 1, 1)$ are all meaningless.
- $(0, 0, 0, 1)$ is also contradictory as given a function and its input, the output can be obtained so it cannot be hidden.

In case of outsourced computation, the VASP might also want to verify the soundness of the received output. Variants of publicly verifiable computation can help to enable this (e.g. in case of MPC [ZNP15]), however, we assume the DB is only honest-but-curious and not malicious, especially as the DB is motivated to provide reliable service in order to retain its customers.

3.3.2 On the Used Notations

Having identified the possible requirements of the different parties in the system, we introduce a notation to refer to the possible scenarios that depend on the requirements of the participants.

From the viewpoint of the DOs, four different worlds might exist depending on whether the DB and the VASP are trusted or not. Each world can be further subdivided based on the trust between the DB and the VASPs:

- the DB either allows any function evaluation to the VASP (0) or restrict the information that a function can leak about the data (1);
- as described in §3.3.1, the confidentiality preferences of a VASP can be described with a 4-tuple.

Accordingly, we denote the cumulated confidentiality preferences of the parties in square brackets, where letters indicate the DO's trust (T) or distrust (U) towards the DB (on the top) and the VASP (bottom). The letters are followed by number(s) indicating the requirements of the DB and the VASP towards each other (using the above described notation). For example, $\left[{}^{\mathsf{T0}}_{\mathsf{T0000}}\right]$ denotes the scenario, where parties fully trust each other.

To denote more general scenarios, we use $*$ as a wildcard, that can substitute both values of a number. Somewhat misusing the notation, we even leave out $*$ from the notation, when it does not cause misunderstanding. In this way, we simplify the notation of general scenarios, which include all the sub-scenarios that the missing numbers would determine. For example, $\left[\begin{smallmatrix}\mathsf{U}*\\ \mathsf{T}****\end{smallmatrix}\right]$ will be denoted by $\left[\begin{smallmatrix}\mathsf{U}\\ \mathsf{T}\end{smallmatrix}\right]$ referring to every scenario where the DB is untrusted but the VASP is trusted by the DO.

In the sequel, we will use the following notations. The database, provided by a DO is denoted by $\mathcal{X}$ and the corresponding set of metadata by $\mathcal{M}$. A data entry $x_m \in \mathcal{X}$ has metadata $m \in \mathcal{M}$ while a DB determines the function class $\mathcal{F}$ that it allows to evaluate to VASPs, whose function we denote with f.

One might miss from the description of the scenarios the specification of the enforcer of an access policy, which can be done either by a DO or by the DB. We find that this question is rather technology related and the effect of the options are the same, so we do not define this in the use-cases.

3.3.3 Resulting Scenarios from the Data Owner's Perspective

At this point, we are ready to identify the various scenarios that arise from requirements of the actors in a data market. As the data owner has a central role from the privacy perspective, we choose to classify the scenarios based on the data owners trust preferences. In the next level, subclasses are formed based on the trust relation between the data broker and the value-added service provider. The summary of the problem domain structuring is depicted in Table 3.1.

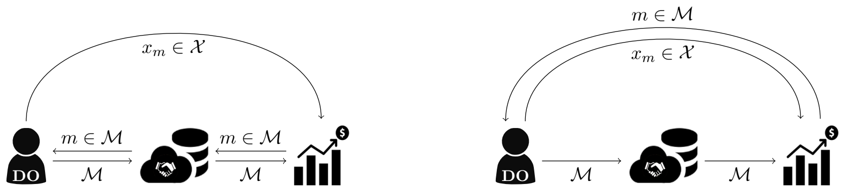
Next, we are going to go through the scenarios, discussing their relevance, the related cryptographic solutions, and the open problems.

Trusted Service Provider and Broker

We start with the case of a naive data owner who trusts the other parties who handle his or her data ($\left[\begin{smallmatrix}\mathsf{T}\\ \mathsf{T}\end{smallmatrix}\right]$). The reason of this trust can be diverse. Just to name some of them:

- The data in question is not necessarily sensitive. This is typically the case when the added value of the data owner does not lie in sharing the data but rather in measuring it. In this sense, it would be better to call such entities device owner, as they make measurements of publicly accessible phenomenons (e.g. weather).
- Trust can be the result of contractual clauses.
- Technological barriers can also enforce trust (when there is no other available option or it would be too expensive).

Since the DB is trusted, we assume that the access control to the data is realised by the DB. Access control policies can represent both the preferences of the user (e.g. who might prefer not to share data with certain VASPs) and the restrictions coming from the business model (e.g. which data chunks are sold together).



(a) Scenario $\left[{}^{\mathsf{U0}}_{\mathsf{T1101}}\right]$.

(b) Scenario $\left[{}^{\mathsf{U0}}_{\mathsf{T1111}}\right]$.

Figure 3.2: Ideal solutions for type $\left[{}^{\mathsf{U}}_{\mathsf{T}}\right]$ scenarios.

Limitations of the scenarios. In the $\left[{}^{\mathsf{T}}_{\mathsf{T}}\right]$ setting, only a very few constraint can be applied in the relation of the DB and a VASP, because both of them can access plaintext data. Accordingly, the DB is unable to restrict the scope of computable functions ($\left[{}^{\mathsf{T1}}_{\mathsf{T}}\right]$), as a VASP can freely use the downloaded data. Consequently in type $\left[{}^{\mathsf{T0}}_{\mathsf{T0***}}\right]$ scenarios the computed functions cannot be monitored, even if the VASP would accept this (except the case when the DB and the VASP are realised by the same entity). Regarding the privacy of a VASP, the input data to its function cannot be hidden from the DB without hiding the accessed metadata as well. This is because the DB has access to the plaintext values corresponding to every metadata, thus ruling out the type $\left[{}^{\mathsf{T}}_{\mathsf{T*10*}}\right]$ scenarios.

Relevant scenarios.

Scenario $\left[{}^{\mathsf{T0}}_{\mathsf{T0000}}\right]$. In this case, the DB and the VASP can be considered to be equivalent, as they have access to the same information.

Scenarios $\left[{}^{\mathsf{T0}}_{\mathsf{T1000}}\right]$ and $\left[{}^{\mathsf{T0}}_{\mathsf{T1001}}\right]$. These are the traditional not privacy-preserving data market scenarios, where the VASP simply receives the bought plaintext data without any further restrictions. When the value-added service reveals the output of the computation of the service provider, we obtain scenario $\left[{}^{\mathsf{T0}}_{\mathsf{T1000}}\right]$.

Scenarios $\left[{}^{\mathsf{T0}}_{\mathsf{T1110}}\right]$ and $\left[{}^{\mathsf{T0}}_{\mathsf{T1111}}\right]$. In these cases the VASP intends to protect its intellectual property or the privacy of its users. The naive solution is to download the entire database (i.e. buy the relevant database in the data market) and execute the necessary operations locally. At the same time, scenario $\left[{}^{\mathsf{T0}}_{\mathsf{T1111}}\right]$ is a typical use-case for single-server PIR, that can help to hide the accessed data entries while decreasing the communication cost. From the pricing perspective, the usage of a 1 out of n OT protocol is even more beneficial, because it ensures that the VASP obtains only a single data entry, while it still remains hidden which was the accessed entry. In this way, the privacy of the input does not require extra expenses, as unnecessary data does not have to be bought to ensure privacy.

Trusted Service Provider and Untrusted Broker

Type $\left[{}^{\mathsf{U}}_{\mathsf{T}}\right]$ scenarios are probably the most common among the privacy-preserving data market concepts today. In this setting the data owner trusts the entity that pays for his or her data in the sense that raw data is sold without restrictions on its usage. At the same time, the natural goal of the DO is to avoid the data leakage during the marketing and selling process, therefore does not trust the DB. As the DB cannot handle plaintext data, in practice often peer-to-peer communication is used to transfer data between seller and customer (see e.g. the federated approach of International Data Spaces [OLJ+19, §6]). In these scenarios, the DB only handles metadata as cleartext, so from the data security point of view, its role is equivalent (does not leak more information about the data) to blockchain-based decentralised data markets [IOT, Oce, Dat17].

Limitations of the scenarios. Similarly to $\left[\substack{\mathsf{U}\\ \mathsf{T}}\right]$, in case of $\left[\substack{\mathsf{U}\\ \mathsf{T}}\right]$ it is impossible to limit or even observe the usage of the data, as plaintext data is available for the VASP (after buying it), ruling out type $\left[\substack{\mathsf{U1}\\ \mathsf{T}}\right]$ and $\left[\substack{\mathsf{U}\\ \mathsf{T0{*}{*}{*}}}\right]$ scenarios. The type $\left[\substack{\mathsf{U}\\ \mathsf{T{*}0{*}{*}}}\right]$ scenarios are also contradictory because the DB cannot know the input of VASPs, which they bought for their computation, without contradicting the DO's will.

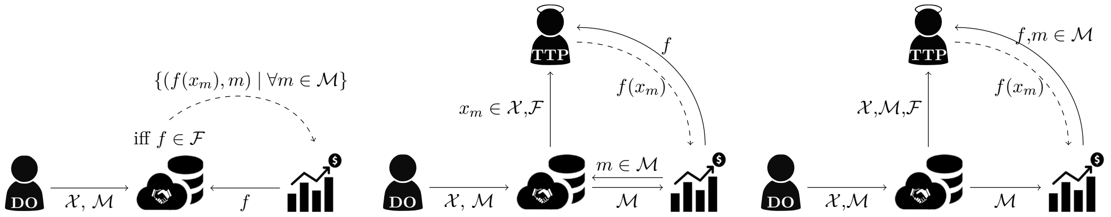
Relevant scenarios.

Scenarios $\left[\substack{\mathsf{U0}\\ \mathsf{T1101}}\right]$ and $\left[\substack{\mathsf{U0}\\ \mathsf{T1100}}\right]$. The ideal realisation of the former scenario is depicted in Fig. 3.2a. Undisputedly, peer-to-peer data transmission is not always possible, e.g. when a DO does not intend to store data; has availability constraints; he or she is unable to handle possibly large number of data requests.

In these cases, secure data storage can be an important service of the DB. If it stores and supplies the VASPs with encrypted data, the task of the DO reduces to provide the decryption key to the VASP. This method reduce both the cost and the frequency of the necessary communication between a DO and a VASP. Access control of the data can be realised simultaneously by the DO (through key sharing) and DB (through ciphertext sharing). In §5 we are going to consider fine-grained access control to the data in more details.

Scenarios $\left[\substack{\mathsf{U0}\\ \mathsf{T1111}}\right]$ and $\left[\substack{\mathsf{U0}\\ \mathsf{T1110}}\right]$. As depicted in Fig. 3.2b the only difference between these and the previously discussed scenarios is that in these the DB cannot even see the accessed metadata. In practice it reduces the task of the DB to the level of a public bulletin board on which the available metadata is advertised. However, because of the same reasons as above, secure data storage can be an important supplementary service again. The previously described methods are applicable here again but now these have to be extended with a privacy-preserving method for the VASP to obtain the bought ciphertext without disclosing which information it needed. For this, PIR and OT can be used as in $\left[\substack{\mathsf{T0}\\ \mathsf{T1111}}\right]$. Furthermore if the DO chooses to forward also the metadata[1] in encrypted form to the DB, then searchable encryption can be applied to find the relevant data entry. We further

[1] Clearly, some porperties of the data has to be shared with the DB, otherwise it cannot provide the brokering service. However this information is enough to have about bigger data chunks, in which the VASP can search for the necessary information without revealing what exactly it wants to know.


$\{(f(x_m), m) \mid \forall m \in \mathcal{M}\}$
iff $f \in \mathcal{F}$
DO
$\mathcal{X}, \mathcal{M}$
f
TTP
$x_m \in \mathcal{X},\mathcal{F}$
f
$f(x_m)$
DO
$\mathcal{X}, \mathcal{M}$
$m \in \mathcal{M}$
$\mathcal{M}$
TTP
$f,m \in \mathcal{M}$
$\mathcal{X},\mathcal{M},\mathcal{F}$
$f(x_m)$
DO
$\mathcal{X},\mathcal{M}$
$\mathcal{M}$

(a) Scenario $\left[{}^{\mathsf{T1}}_{\mathsf{U0111}}\right]$ (b) Scenario $\left[{}^{\mathsf{T1}}_{\mathsf{U1001}}\right]$. (c) Scenario $\left[{}^{\mathsf{T1}}_{\mathsf{U1111}}\right]$.

Figure 3.3: Ideal solutions for $\left[{}^{\mathsf{T}}_{\mathsf{U}}\right]$ type scenarios (DO trusts the DB but not the VASP). Dashed arrow indicates that output is only sent if $f \in \mathcal{F}$.

explore this pathway in §6. We note, that in these scenarios, pricing must depend on the access frequency or on which specific data chunk is accessed, because these are the only available information to the DB.

Untrusted Service Provider and Trusted Broker

In the following scenarios, we assume that the DO trusts the DB but not the VASP, that might be surprising for the first sight. Distrust towards a VASP means that it is not allowed to use the data for whatever it wants. Instead of buying the raw data, in these scenarios the VASP pays for information or some insights that can be deduced from the data. This approach allows more complex pricing compared to selling data chunks that can be beneficial for both the VASPs (who can pay for exactly what they need) and the DOs (who can reduce the amount of disclosed information thus preserving the value of their data). When the data broker is trusted like now, that suggests that the value of the involved data is not related to its sensitivity but rather to its existence and availability. For example, measurements of a machine tool in a factory is only valuable in a very specific context (for specific companies using the same type of machine tool) but not outside of it (e.g. not for a DB).

Limitations of the scenarios. The restriction that VASPs can buy insights but not raw data has the corollary that type $\left[{}^{\mathsf{T0}}_{\mathsf{U}}\right]$ scenarios cannot exist. The reason is the following. If a VASP could execute arbitrary computation without restriction (as should happen in these scenarios), then with the evaluation of the identity function it could obtain raw data, which the DO wants to avoid. Furthermore type $\left[{}^{\mathsf{T}}_{\mathsf{U*10*}}\right]$ scenarios are also contradictory since the public metadata of the accessed input reveals to the DB the intended input which it stores in the clear.

Relevant scenarios.

Scenarios $\left[{}^{\mathsf{T1}}_{\mathsf{U0111}}\right]$ and $\left[{}^{\mathsf{T1}}_{\mathsf{U0110}}\right]$. Fig. 3.3a depicts a secure but inefficient solution for the problem related to $\left[{}^{\mathsf{T1}}_{\mathsf{U0111}}\right]$. The challenge here is to reduce the computational (evaluation of f, $|\mathcal{M}|$ times) and communication (return the $|\mathcal{M}|$ number of outputs) costs of the naive solution.

Scenarios $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1001}\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1000}\end{smallmatrix}\right]$. The ideal solution for the problem of $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1001}\end{smallmatrix}\right]$ is represented in Fig. 3.3b. The main challenge here is to enable the DB to verify the function of a VASP without learning anything more, than that the function satisfies a given requirement or not. To the best of our knowledge, this task is only solvable for very simple requirements today.

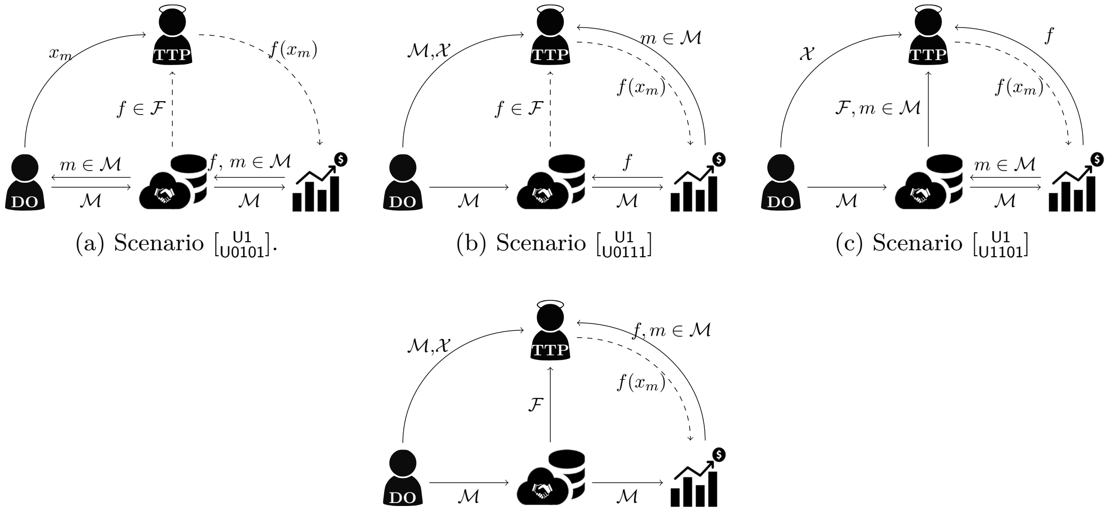
- When the function provider receives output in PFE, the output length is typically shorter than the input length. This is necessary for achieving meaningful security. In 3.3b, this restriction rules out the evaluation of the identity function, thus allowing the DB to ensure that this simple condition is fulfilled.
- Using a TEE on the DB side, could be an interesting direction. For this, the VASP should supply the TEE with an encrypted description of its function, that is decrypted and executed in the TEE (possibly if certain conditions hold). The output should be encrypted with the key of the VASP in $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1001}\end{smallmatrix}\right]$, but can be sent in the clear in $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1000}\end{smallmatrix}\right]$.
- In $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1000}\end{smallmatrix}\right]$, a VASP could use obfuscation to hide its function (if it is sufficiently simple) before sending it to the DO for evaluation. If the output is short (so it cannot reveal the input) the DB can be forward it to the VASP. Unfortunately the state of the art obfuscation methods only enable the obfuscation of extremely simple functions, e.g. pattern matching with wildcards [BKM+18] that has a one bit output (showing whether the input matched the pattern or not).

These restricted solutions highlight the need for exploring the opportunities of restricted private computation. Finding different trade-off between function privacy and control is an open problem. In §4, we contribute this research direction.

Scenarios $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1111}\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}\mathsf{T1}\\ \mathsf{U1110}\end{smallmatrix}\right]$. The privacy goal of Fig. 3.3c is an extension of Fig. 3.3b with the privacy of the accessed metadata. Obviously, one way to avoid the trusted third party (TTP) here is direct communication between a VASP and a DO but one naturally would like to avoid this solution because of it lacks the benefits of a data market. We are not aware of any protocols that would be directly applicable here, and the solution seems rather challenging because of the mentioned function hiding vs. verification problem that is extended with the private information retrieval problem.

Untrusted Service Provider and Broker

Finally, we turn our attention towards the case of the most distrustful data owners, who neither trust the DB, nor the VASP. The interpretation of type $\left[\begin{smallmatrix}\mathsf{U}\\ \mathsf{U}\end{smallmatrix}\right]$ scenarios is very similar to that of $\left[\begin{smallmatrix}\mathsf{T}\\ \mathsf{U}\end{smallmatrix}\right]$. The only difference is that now the data to be sold is private, consequently the broker also should not have direct access to it.



(a) Scenario $\begin{bmatrix}\text{U1}\\\text{U0101}\end{bmatrix}$.

(b) Scenario $\begin{bmatrix}\text{U1}\\\text{U0111}\end{bmatrix}$

(c) Scenario $\begin{bmatrix}\text{U1}\\\text{U1101}\end{bmatrix}$

(d) Scenario $\begin{bmatrix}\text{U1}\\\text{U1111}\end{bmatrix}$

Figure 3.4: Ideal solutions for $\begin{bmatrix}\text{U}\\\text{U}\end{bmatrix}$ type scenarios, where the data owner does not trust the other parties.

Limitations of the scenarios. In $\begin{bmatrix}\text{U}\\\text{U}\end{bmatrix}$ scenarios neither the DB, nor the VASP can access to plaintext data. Consequently, the DB will never know the plaintext form of the data that was used as input for the computations of VASPs. This shows that the sub-scenarios $\begin{bmatrix}\text{U}\\\text{U}{*}\text{0}{*}{*}\end{bmatrix}$ will never occur. The impossibility of $\begin{bmatrix}\text{U0}\\\text{U}\end{bmatrix}$ has the same reason why $\begin{bmatrix}\text{T0}\\\text{U}\end{bmatrix}$ is ruled out: namely unrestricted function evaluation would enable VASPs to obtain plaintext data through computing the identity function.

Relevant scenarios.

Scenarios $\begin{bmatrix}\text{U1}\\\text{U0101}\end{bmatrix}$ and $\begin{bmatrix}\text{U1}\\\text{U0100}\end{bmatrix}$. Assuming shared keys between DOs and VASPs, these scenarios (see ideal solution in Fig. 3.4a) correspond to computation outsourcing.

- The standard solution is to use a HE scheme to encrypt data that is stored by the DO. Upon the function and (meta)data request of the VASP, the function can be verified and computation carried out on the corresponding ciphertext. The resulting ciphertext can be decrypted by the VASP using the shared key. When the computation involves data, coming from different DOs, multi-key HE [CDKS19] can be beneficial. We note that access to "insights" is controlled through key sharing by the DO, and function verification by the DB.
- Another approach is to use a TEE on the DB side, in which data can be decrypted before computation (and if the DB is not supposed to see, the output can be encrypted with the key of the VASP).

- When the sold insight is not confidential in $[^{\mathtt{U1}}_{\mathtt{U0100}}]$, FE can be an alternative tool. For this, the DB has to store FE ciphertext and after the verification of a function request from a VASP, it requests functional key either form the DO or from a key distribution authority, that enables the evaluation of the function during decryption.

Scenarios $[^{\mathtt{U1}}_{\mathtt{U0111}}]$ and $[^{\mathtt{U1}}_{\mathtt{U0110}}]$. These are variants of the previous two scenarios, in which the accessed data's metadata has to be hidden as well from the DB (see Fig. 3.4b). Consequently, the above described approaches need to be integrated with PIR or OT, which seems to be challenging regarding efficiency.

Scenarios $[^{\mathtt{U1}}_{\mathtt{U1101}}]$ and $[^{\mathtt{U1}}_{\mathtt{U1100}}]$. The goal, represented in Fig. 3.4c, extends that of Fig. 3.4a by keeping the computation logic of the VASP private. When handling function verification on the DO side is acceptable, the following approaches can apply:

- In $[^{\mathtt{U1}}_{\mathtt{U1100}}]$, FE can be used to encrypt the data, stored by the DB. For the computation of f, a VASP needs to request ciphertext from the DB and functional key for f from either the DO or a key distribution authority, enabling the evaluation of f (but nothing more) during decryption.
- The scenario, considered in [GLL+19] is almost exactly matches to $[^{\mathtt{U1}}_{\mathtt{U1101}}]$. Their MPC-based solution requires the VASP to agree on the function to be evaluated with the DOs whose data it intends to use.

When the DB has to solve access control entirely alone, the problem of private function verification comes up again.

Scenarios $[^{\mathtt{U1}}_{\mathtt{U1111}}]$ and $[^{\mathtt{U1}}_{\mathtt{U1110}}]$. Extending the privacy goal of VASPs, with hiding the access pattern as well, seems rather challenging and we are not aware of solutions that fulfil all the requirements (see Fig. 3.4d).

3.4 The Scope of this Dissertation

In this chapter, we have observed the diversity of data markets that leads to various use-cases that can be organized into two main categories. The first is the traditional one, where the VASPs can buy raw data. In this case, we have to face similar problems than in the context of secure cloud storage, where the DB plays the role of the cloud service provider while DOs and VASPs are the clients. Depending on the exact scenario, one of these problems has to be solved: the stored data has to be hidden from the DB ($[^{\mathtt{U0}}_{\mathtt{T110*}}]$); the way (when and what) data is accessed has to be hidden ($[^{\mathtt{T0}}_{\mathtt{T111*}}]$), or both of these are required to be hidden ($[^{\mathtt{U0}}_{\mathtt{T111*}}]$). However, these are well known problems with rich literature, there are still room for improvement when considering more concrete problems. In §5–6, we do exactly this, focusing on the scenarios $[^{\mathtt{U0}}_{\mathtt{T11*1}}]$ (and the solutions naturally extends to scenarios $[^{\mathtt{U0}}_{\mathtt{T11**}}]$). Our goal is to minimize the usability and efficiency gap between secure and insecure solutions. Namely, in §5 we

					Trusted Data Broker (stores plaintext data)		Untrusted Data Broker (without access to plaintext data)	
	function	input value	input metadata	output	No restriction on the computed function (0)	Limited function queries (1)	No restriction on the computed function (0)	Limited function queries (1)
Trusted VASP (can access to plaintext data)	0	0	0	0	DB = VASP	As VASP is trusted by the DO, it can access to plaintext data that can be used for any computation without the consent of the DB	Revealing the input to the DB contradicts with the data owner's will to hide data from DB	As VASP is trusted by the DO, it can access to plaintext data that can be used for any computation without the consent of the DB
	0	0	1	0	public input⇔secret metadata			
	0	0	0	1	Function and input determine the output			
	0	0	1	1				
	0	1	0	0	Metadata reveals input		As VASP can access to plaintext data the evaluated function is not revealed to DB	
	0	1	0	1				
	0	1	1	0	As VASP can access to plaintext data the evaluated function is not revealed to DB			
	0	1	1	1				
	1	0	0	0	VASP computes locally and publishes the output		Revealing the input to the DB contradicts with the data owner's will to hide data from DB	
	1	0	0	1	VASP computes locally			
	1	0	1	0	Public input⇔secret metadata			
	1	0	1	1				
	1	1	0	0	Metadata reveals input		Fig. 3.2a with published output	
	1	1	0	1			Fig. 3.2a (see §5)	
	1	1	1	0	VASP downloads all data, computes locally, publishes output		Fig. 3.2b with published output	
	1	1	1	1	VASP downloads all data and computes locally		Fig. 3.2b (see §6)	
Untrusted VASP (without access to plaintext data)	0	0	0	0	Without restriction on the computable functions VASP could access the output of the identity function contradicting with the requirement that it should not get plaintext data	Function verification with outsourced computation	Without restriction on the computable functions VASP could access the output of the identity function contradicting with the requirement that it should not get plaintext data	Revealing the input to the DB contradicts with the data owner's will to hide data from DB
	0	0	1	0		Public input⇔ secret metadata		
	0	0	0	1		Function and input determine the output		
	0	0	1	1				
	0	1	0	0		Metadata reveals input		Fig. 3.4a with published output
	0	1	0	1				Fig. 3.4a
	0	1	1	0		As Fig. 3.3a with published output		Fig. 3.4b with published output
	0	1	1	1		Fig. 3.3a		Fig. 3.4b
	1	0	0	0		Fig. 3.3b with published output		Revealing the input to the DB contradicts with the data owner's will to hide data from DB
	1	0	0	1		Fig. 3.3b (see §4)		
	1	0	1	0		Public input ⇔ secret metadata		
	1	0	1	1				
	1	1	0	0		Metadata reveals input		Fig. 3.4c with published output
	1	1	0	1				Fig. 3.4c
	1	1	1	0		Fig. 3.3c with published output		Fig. 3.4d with published output
	1	1	1	1		Fig. 3.3c		Fig. 3.4d

Table 3.1: Summary of the identified scenarios and their ideal solutions, including the contradictory cases (denoted with grey), the trivial ones (green), the ones that are interesting from the viewpoint of cryptography (orange and darker orange for those that are considered in this dissertation).

explore how can a DO provide fine-grained access control to his or her data even when exceptions occur, i.e. when the access rights of someone has to be revoked unexpectedly. In 6 we further extend our solution to efficiently handle the problem, depicted in Fig. 3.2b as long as part of the accessed metadata is allowed to leak (assuming that the DB can recover parts of the metadata anyway, e.g. using some side-information). Combining our results in §5–6, gives a possible solution for scenario $\left[{}^{\mathsf{U0}}_{\mathsf{T1111}}\right]$. We note that scenarios $\left[{}^{\mathsf{U0}}_{\mathsf{T11**}}\right]$ are especially important because solutions for the challenges they pose can be adapted to the context of decentralised data markets where data transmission is solved with peer to peer communication [OLJ+19].

In the second category of the identified use-cases, not raw data is sold but it is processed first as a part of the service, and only the result is shared with the customer. This approach allows for more sophisticated pricing and better privacy for the data owner. At the same time, it requires secure computation techniques and realisation poses challenges. Even if a concrete scenario coincides with the use case of a primitive or a protocol, it is very uncertain that a concrete computation is feasible to realize with the available techniques or not. Moreover, we have to face with the problem of verifying a private function in scenarios $\left[{}^{\mathsf{*1}}_{\mathsf{U1***}}\right]$. In §4, we study this problem and propose a relaxation to obtain a viable solution (at least for simple statistical functions) in the context of $\left[{}^{\mathsf{T1}}_{\mathsf{U1001}}\right]$.

Chapter 4

Partial Input Information in Private Function Evaluation

4.1 Introduction

Secure two-party computation (2PC) a.k.a. secure function evaluation (SFE) protocols enable two parties, Alice and Bob, to compute a function of their choice on their private inputs without disclosing their secrets to each other or anyone else (see Fig. 4.1a). In real life, however, the participants not necessarily have interchangeable roles. We call private function evaluation (PFE) a protocol if one party can alone choose the function to evaluate, while the other provides the input to it (see Fig. 4.1b) while both of them intends to hide their contribution. PFE can be realized by invoking 2PC after the function was turned into data. A universal function [Val76] is a "programmable function" that can implement *any* computation up to a given complexity. It takes two inputs, the description of the function to be computed and the input to it. By evaluating a public universal function using 2PC, all feasibility results extend from 2PC to PFE. Improving efficiency turns out to be more challenging. Indeed, universal functions cause significant – for complex computations even prohibitive – overhead, and the elimination of this limitation was the primary focus of PFE research [KS08, KS16].

In [C2], we initiated the study of a security issue that – to the best of our knowledge – received no attention earlier. This is the problem, we identified in the context of data markets, and which is present in scenario $\left[\substack{\mathsf{T1}\\ \mathsf{U1001}}\right]$, according to the notations of §3 (see Fig. 3.3b). In this chapter, we elaborate on our findings. More concretely, we focus on the opportunities of the input provider to control the information leakage of her input. As PFE guarantees Bob that his function is hidden from Alice, he can learn some information about the input of Alice such that it remains hidden what was exactly revealed. Disclosing the entire input by evaluating the identity function is typically ruled out by the restriction that the computable function class has shorter output length than input length. At the same time, the following question arises:

Is it really possible to determine the computable function class so that no *function is included which could reveal sensitive information about the input?*

We argue that most often exceptions occur in every function class, so measures are required to also protect such partial information besides the protection of the input as a whole. As intentional partial information recovery goes entirely unnoticed when only the function provider, Bob receives the function's output, later on we consider this scenario.

For a simple and illustrative example, let us recall one of the most popular motivating applications for PFE. In privacy-preserving credit checking [PSS09, §7], Alice feeds her private data to a Boolean function of her bank (or another service provider) that decides whether she is eligible for credit or not. Using PFE for such computation allows Alice to keep her data secret and the bank to hide its crediting policy. Notice that the function provider can extract *any binary information* about the input and use it, e.g. to discriminate clients. The leaked partial information can be, e.g. gender or the actual value of any indicator variable about the data that should not be necessary to reveal for credit checking. This problem motivates us to answer the following questions:

Is it possible to enable the input provider to rule out the leakage of specific sensitive information in PFE without exposing what partial information she wants to hide? What kind of trade-offs between input and function privacy can lead to efficient protocols with meaningful security?

4.1.1 Contributions

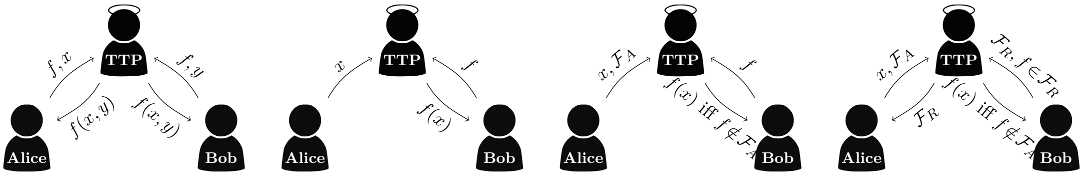
Our contributions can be summarized as follows.

- We initiate the study of partial information protection in the context of private function evaluation.

- To take the first step, we put forward the notion of controlled private function evaluation (CPFE) and formally define its security (see Fig. 4.1c for its ideal functionality). We also devise a relaxed definition, called rCPFE (see Fig. 4.1d) that guarantees weaker (but still reasonable) k-anonymity style function privacy leading to a trade-off between security and efficiency.

- Then we show conceptually simple, generic realizations of both CPFE and rCPFE. In the latter case, we utilize the modified function privacy guarantee (through using functional encryption) to enable the reusability of the protocol messages in case of multiple function evaluations. As a result, in our rCPFE when evaluating the same function(s) on multiple, say d inputs, the communication and online computation overhead only increases with an additive factor proportional to d instead of a multiplicative factor as in ordinary PFE.

- To demonstrate the practicality of the rCPFE approach, we instantiate our generic protocol for the inner product functionality enabling secure statistical analysis in a controlled manner under the standard DDH assumption. Our proof of concept implementation shows that the reusability property indeed results in a significant



(a) SFE or 2PC (b) PFE (c) CPFE (d) rCPFE

Figure 4.1: Comparison of the *ideal functionality* of different concepts for secure function evaluation, realized with the help of a trusted third party (TTP). The key difference lies in which information Alice and Bob can or cannot have access to.

performance improvement over the state of the art secure inner product evaluation method [DSZ15].

4.1.2 Applications

We believe that in most PFE applications, the evaluated function class also permits the leakage of potentially sensitive partial information about the input. The above example demonstrates that this is true even for very restricted Boolean functions. To motivate our inner product rCPFEprotocol, we mention two of its possible application in a data market (both in scenario $\left[{}^{\mathsf{T1}}_{\mathsf{U1001}}\right]$).

Location Privacy. Let us assume that a data broker periodically collects location-based information from DOs in vector form, where vector elements correspond to information related to specific positions. Such data can be important for VASPs, offering location-based services, without the proper infrastructure to collect the necessary data. During their interaction that can be an inner product computation[1], the VASP should hide the location of its users, while the DB may want to protect the exact information in specific locations or to adjust higher price if specific measurements are used. These can be achieved by having control over the possible queries of a VASP.

Logistic Regression Evaluation. The linear part of logistic regression computation is an inner product of the input and weight vectors. Our inner product rCPFE can help to rule out weight vectors that are unlikely to belong to a model but are base vectors that could reveal a sensitive input vector element.

[1] E.g. multiplying the data vector with a position vector (that is non-zero in all positions representing locations close to the user – possibly containing weights depending on the distance – and zero otherwise) can give useful information.

4.2 Related Work

Some PFE variants share ideas with our concepts. Semi-private function evaluation (semi-PFE) [PSS09, KKW17] for instance, also relaxes the function privacy requirement of PFE by revealing the topology of the function being evaluated. While this relaxation also leads to a useful trade-off between function privacy and efficiency, unfortunately, the available extra information about the function does not necessarily allow Alice to rule out the evaluation of functions that are against her interest.

Selective private function evaluation (SPFE) [CIK+01] deals with a problem that is orthogonal to the one considered in this paper. Namely, SPFE also aims to conceal information that is leaked in PFE. However, instead of protecting Alice (the data owner), it intends to increase the security of Bob by hiding from Alice the location of the function's input in her database via using private information retrieval (PIR).

Leaving the field of PFE and comparing our work to related problems in secure computation, we find that hiding the computed function raises similar issues in other contexts. [BGJS16] put forth the notion of verifiable obfuscation that is motivated by the natural fear for executing unknown programs. The goal here is similar than in our setting: some assurance is required that the hidden functionality cannot be arbitrary. However, the fundamental difference between our CPFE and the verifiable obfuscation and verifiable FE of [BGJS16] is that while the latter ones enforce correctness when an obfuscator or authority may be dishonest, CPFE tries to disable semi-honest parties to evaluate specific functions (i.e. to handle exceptions in PFE).

Our rCPFE is built upon functional encryption (FE) in a black-box manner. This generalization of traditional encryption was first formalized by [BSW11]. While general-purpose FE candidates [GGH+13, GGHZ16] currently rely on untested assumptions like the existence of indistinguishability obfuscation or multilinear maps, our application does not require such heavy hammers of cryptography (see details in §2.5.3). In the context of FE, [NAP+14] raised the question of controllability of function evaluation. The essential difference, compared to our goals, is that they want to limit repeated evaluations of the *same* function[2] that they solve with the involvement of a third party.

Finally, we sum up the state of the art of private inner product evaluation. The provably secure solutions are built either on partially homomorphic encryption schemes [GLLM04, DC14] or 2PC protocols [DSZ15] but public-key inner product FE [ABCP15] is also capable of the same task. At the same time, several ad-hoc protocols achieve better performance in exchange for some information leakage (see, e.g. [ZWH+15] and the references therein), but these constructions lack any formal security argument.

[2] In FE schemes, the control over the computable functions is in the hand of the master secret key holder, so this is not an issue unlike in PFE.

PARAMETERS: participants P_1, P_2, a class $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$ of deterministic functions *[and an integer $\kappa > k$]*

FUNCTIONALITY:
On inputs $x_1, \dots, x_d \in \mathcal{X}$ and $\mathcal{F}_A \subset \mathcal{F}$ from P_1; and $\mathcal{F}_B = \{f_1, \dots, f_k\} \subset \mathcal{F}$ from P_2

- P_1 receives no output, *[or P_1 receives $\mathcal{F}_R$ s.t. $\mathcal{F}_B \subset \mathcal{F}_R \subset \mathcal{F}$ and $|\mathcal{F}_R| = \kappa$]*
- P_2 obtains $\{y'_{i,j} = f'_j(x_i)\}_{i \in [d], j \in [k]} \subset \mathcal{Y} \cup \{\perp\}$ for

$$f'_j(x_i) = \begin{cases} f_j(x_i) & \text{if } f_j \notin \mathcal{F}_A \\ \perp & \text{otherwise.} \end{cases}$$

Figure 4.2: Ideal functionalities for $\mathcal{F}_{\text{CPFE}}$ and $\mathcal{F}_{\textbf{rCPFE}}$ (see the extensions in brackets) formulated generally for multiple inputs and multiple functions.

4.3 General Approaches for Securing Partial Input Information in PFE

In this part, we introduce the notion of controlled PFE and in §4.3.1 formally define its security in different flavours. Next, in §4.3.2–4.3.3, we propose two general protocols satisfying these security requirements.

4.3.1 Definitional Framework

Our first security definition for controlled PFE captures the intuitive goal of extending the PFE functionality with a blind function verification step by P_1 to prevent unwanted information leakage. See the corresponding ideal functionality $\mathcal{F}_{\text{CPFE}}$ in Fig. 4.2 that we call controlled PFE, and the security definition below. For the ease of exposition, later on we denote the inputs of the participants as $\mathsf{inp} = (\{x_i\}_{i \in [d]}, \mathcal{F}_A, \{f_j\}_{j \in [k]})$ with the corresponding parameters.

Definition 5 (SIM security of CPFE wrt. semi-honest adversaries)**.** *Let* Π *denote a Controlled PFE (CPFE) protocol for a function class* $\mathcal{F}$ *with functionality* $\mathcal{F}_{\text{CPFE}}$ *(according to Fig. 4.2). We say that* Π *achieves SIM security against semi-honest adversaries, if the following criteria hold.*

- Correctness: *the output computed by* Π *is the required output, i.e.*

$$\Pr[\mathsf{output}^{\Pi}(1^\lambda, \mathsf{inp}) \neq \mathcal{F}_{\text{CPFE}}(\mathsf{inp})] \leq \mathsf{negl}(\lambda).$$

- Function Privacy: *there exists a probabilistic polynomial time (PPT) simulator* $\mathcal{S}_{P_1}$*, s.t.*

$$\{\mathcal{S}_{P_1}(1^\lambda, \{x_i\}_{i \in [d]}, \mathcal{F}_A)\}_{\lambda, x_i, \mathcal{F}_A} \overset{c}{\approx} \{\mathsf{view}^{\Pi}_{P_1}(1^\lambda, \mathsf{inp})\}_{\lambda, x_i, f_j, \mathcal{F}_A}.$$

- Data Privacy: *there exists a PPT simulator* $\mathcal{S}_{P_2}$*, s.t.*

$$\{\mathcal{S}_{P_2}(1^\lambda, \{f_j\}_{j\in[k]}, \{y'_{i,j}\}_{i\in[d],j\in[k]}\}_{\lambda,f_j} \overset{c}{\approx} \{\mathsf{view}^{\Pi}_{P_2}(1^\lambda, \mathsf{inp})\}_{\lambda,x_i,f_j,\mathcal{F}_A}$$

where $\mathsf{inp} = (\{x_i\}_{i\in[d]}, \mathcal{F}_A, \{f_j\}_{j\in[k]}), f_j \in \mathcal{F}, \mathcal{F}_A \subset \mathcal{F}, x_i \in \mathcal{X}, y'_{i,j} \in \mathcal{Y} \cup \{\bot\}$*, and* $\lambda \in \mathbb{N}$*.*

We also propose a relaxation of Definition 5, which on the one hand gives up perfect function privacy but on the other, allows us to construct efficient protocols while still maintaining a k-anonymity style guarantee [**?**] for function privacy. As SIM security alone cannot measure how much information is leaked by a set of functions, we formulate an additional requirement to precisely characterise function privacy.

Definition 6 (SIM security of relaxed CPFE wrt. semi-honest adversaries)**.** *Let* Π *denote a relaxed CPFE (rCPFE) protocol for a function class* $\mathcal{F}$ *with functionality* $\mathcal{F}_{\boldsymbol{rCPFE}}$ *(according to Fig. 4.2). We say that* Π *achieves SIM security against semi-honest adversaries, if the following criteria hold.*

- Correctness: *the output computed by* Π *is the required output, i.e.*

$$\Pr[\mathsf{output}^{\Pi}(\lambda, \kappa, \mathsf{inp}) \neq \mathcal{F}_{rCPFE}(\kappa, \mathsf{inp})] \leq \mathsf{negl}(\lambda).$$

- Function Privacy: *is defined in two flavours:*
  - κ*-relaxed function privacy holds, if* $\exists$ $\mathcal{S}_{P_1}$*, a PPT simulator, s.t.*

$$\{\mathcal{S}_{P_1}(1^\lambda, \kappa, \{x_i\}_{i\in[d]}, \mathcal{F}_A)\}_{\lambda,\kappa,x_i,\mathcal{F}_A} \overset{c}{\approx} \{\mathsf{view}^{\Pi}_{P_1}(1^\lambda, \kappa, \mathsf{inp})\}_{\lambda,\kappa,x_i,f_j,\mathcal{F}_A}.$$

  - *Strong* κ*-relaxed function privacy holds if besides the existence of the above* $\mathcal{S}_{P_1}$*, it also holds that for any PPT* $\mathcal{A}$*:*

$$\left|\Pr[\mathcal{A}(\mathsf{aux}, \mathcal{F}_R) \in \mathcal{F}_B] - \frac{k}{\kappa}\right| \leq \mathsf{negl}(\lambda)$$

where $\mathsf{aux} \in \{0,1\}^*$ *denotes some a priori known auxiliary information about* $\mathcal{F}_B$*.*

- Data Privacy: *there exists a PPT simulator* $\mathcal{S}_{P_2}$*, s.t.*

$$\{\mathcal{S}_{P_2}(\lambda, \kappa, \{f_j\}_{j\in[k]}, \{y'_{i,j}\}_{i\in[d],j\in[k]}\}_{\lambda,\kappa,f_j} \overset{c}{\approx} \{\mathsf{view}^{\Pi}_{P_2}(\lambda, \kappa, \mathsf{inp})\}_{\lambda,\kappa,x_i,f_j,\mathcal{F}_A}$$

where $\mathsf{inp} = (\{x_i\}_{i\in[d]}, \mathcal{F}_A, \{f_j\}_{j\in[k]}), f_j \in \mathcal{F}, \mathcal{F}_A \subset \mathcal{F}, x_i \in \mathcal{X}, y'_{i,j} \in \mathcal{Y} \cup \{\bot\}$*, and* $\lambda, \kappa \in \mathbb{N}$*.*

Protocol $\Pi^{\mathrm{CPFE}}_{\mathcal{F}}$

PARAMETERS: λ parametrizing security, a function class $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$, and a universal circuit UC for the function class $\mathcal{F}$

INPUTS:

- P_1: $x, \mathcal{F}_A \subset \mathcal{F}$
- P_2: $f \in \mathcal{F}$

PROTOCOL:

Using a secure two-party computation protocol, P_1 and P_2 executes the following computation on their inputs:

- If $f \in \mathcal{F}_A$, return $\perp$ to both P_1 and P_2.
- Otherwise compute the universal circuit $UC(f, x) = f(x) \in \mathcal{Y}$ outputting $\perp$ to P_1 and $f(x)$ to P_2.

Figure 4.3: General 2PC-based CPFE

4.3.2 Universal Circuit-based CPFE

The natural approach for realizing CPFE comes from the traditional way of combining universal circuits and SFE to obtain PFE. Fig. 4.3 shows how the same idea with conditional evaluation leads to CPFE in the single input, single function setting. The following theorem is a straightforward consequence of the security of SFE.

Theorem 4.3.1. *The CPFE protocol of Fig. 4.3 is secure according to Definition 5, if the used SFE protocol is SIM secure in the semi-honest model.*

The main drawback of this approach is that when extending the protocol to handle multiple inputs or functions, its complexity will multiplicatively depend on the number of inputs or functions because of the single-use nature of 2PC.

4.3.3 Reusable Relaxed CPFE from FE

We observe that the notion of rCPFE not only allows the input provider to verify the functions to be evaluated but also opens the door for making parts of the protocol messages reusable multiple times, thus leading to significant efficiency improvements.

A naive first attempt to realize rCPFE is to execute the computation on the side of P_1. Upon receiving a κ function descriptions (including both the intended and dummy functions) P_1 can easily verify the request and evaluate the allowed ones on her input. The results then can be shared with P_2, using an OT scheme achieving both the required data and function privacy level. Unfortunately, the κ function evaluations lead to scalability issues. The subsequent natural idea is to shift the task of function evaluation to P_2, to eliminate the unnecessary computations and to hide the output from

P_1 entirely. Since at this point P_1 has both the inputs and the functions to evaluate, the task resembles secure outsourcing of computation where function evaluation must be under the strict control of P_1. These observations lead us to the usage of FE and the protocol in Fig. 4.4 in which both ciphertext and functional keys can be reused in multiple computations. When instantiated with the FE scheme of [GVW12], $\Pi_{\mathcal{F}}^{\text{rCPFE}}$ can be used for all polynomial sized functions in theory (in practice verifying the circuits would be a bottleneck).

Theorem 4.3.2. *The protocol of Fig. 4.4 is SIM secure according to Definition 6 achieving κ-relaxed function privacy for k function queries by P_2, if the underlying FE scheme is k-query non-adaptive SIM secure (k-NA-SIM) for a single message and the used OT protocol is SIM secure against semi-honest adversaries.*

We prove Theorem 4.3.2, by showing that the protocol of Fig. 4.4 fulfils the requirements of Definition 6 with the assumption that the underlying FE and OT are SIM secure against semi-honest adversaries.

Proof. As correctness directly follows from the correctness of the underlying FE and OT, we turn our attention towards the security requirements. We argue input and weak relaxed function privacy by showing that the view of both parties can be simulated (without having access to the inputs of the other party) using the simulators guaranteed by the SIM security of FE and OT.

Corrupted P_1: Weak Relaxed Function Privacy. Besides its input and output, the view of P_1 consists of the received OT messages and the function query $\mathcal{F}_R$. Simulation becomes trivial because of the fact that the output of P_1 also contains $\mathcal{F}_R$. Thus $\mathcal{S}_{P_1}((x_1, \ldots, x_d), \mathcal{F}_R)$ can return $\mathcal{F}_R$ and the output of the sender's simulator $\mathcal{S}_{OT}^{\mathcal{S}}$ guaranteed by the SIM security of OT. The simulated view is clearly indistinguishable from the real one.

Corrupted P_2: Input privacy. The following simulator $\mathcal{S}_{P_2}$ simulates the view of a corrupt P_2, that consists of its input $(f_1, \ldots, f_k)$, output $\{y_{i,j}'^{*} = f_i'(x_j)\}_{i \in [k], j \in [d]}$, the used randomness and the incoming messages. $\mathcal{S}_{P_2}$ first determines the index set $I^* = \{i \mid \exists j : y_{i,j}' \neq \perp\} \subseteq [k]$. Next, it sets up the parameters of the ideal experiment according to Definition 4. To do so, it samples $(\mathsf{msk_{FE}}^*, \mathsf{pp_{FE}}^*) \leftarrow\!\!\$\, \mathsf{FE.Setup}(\lambda)$ and then for all $i \in I^*$ generates functional secret keys $\mathsf{fsk}_{f_i}^* \leftarrow\!\!\$\, \mathsf{FE.KeyGen}(\mathsf{pp_{FE}}^*, \mathsf{msk_{FE}}^*, f_i)$. For the simulation of the FE ciphertexts (corresponding to unknown messages), we can use the FE simulator $\mathcal{S}_{FE}$ for many messages (implied by one-message q-NA-SIM security [GVW12]). Thus $\mathcal{S}_{FE}(\mathsf{pp_{FE}}^*, \{y_{i,j} = f_i(x_j), f_i, \mathsf{fsk}_{f_i}^*\}_{i \in I^*, j \in [d]}, \lambda) = (\mathsf{ct}_1^*, \ldots, \mathsf{ct}_d^*)$ can be appended to the simulated view together with $\mathsf{pp_{FE}}^*$. The incoming messages of Step III. are simulated using the OT simulator $\mathcal{S}_{OT}^{\mathcal{R}}$ for the receiver. Finally the output of $\mathcal{S}_{OT}^{\mathcal{R}}(\lambda, \{\mathsf{fsk}_{f_i}^*\}_{i \in I^*} \cup \{\perp_i\}_{i \in [k] \setminus I^*})$ is appended to the simulated view.

Now we show the indistinguishability of the real and simulated views. As the inputs and outputs are the same in both cases, we have to compare the randomness and

Protocol $\Pi_{\mathcal{F}}^{\text{rCPFE}}$

Parameters: κ, λ parametrizing security and function class $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$
Inputs:

- P_1: $x_1, \ldots, x_d \in \mathcal{X}, \mathcal{F}_A \subset \mathcal{F}$
- P_2: $\mathcal{F}_B = \{f_1, \ldots, f_k\} \subset \mathcal{F}$

Protocol:

Online Phase

Step I. To initiate the evaluation of functions in $\mathcal{F}_B$, P_2

(1) samples $\kappa - k$ functions randomly: $\{f_i \leftarrow\!\!\$\; \mathcal{F}\}_{k<i\leq\kappa}$,
(2) takes a random permutation on κ elements to set $\mathcal{F}_R := (\hat{f}_1, \ldots, \hat{f}_\kappa)$, where $\hat{f}_i = f_{\sigma^{-1}(i)}$ so that each f_i ends up at position $\sigma(i)$ in the sequence,
(3) finally, sends[3] $\mathcal{F}_R$ to P_1.

Step II. Upon receiving a function request $\mathcal{F}_R$, P_1

(1) samples $(\mathsf{msk}_{\mathsf{FE}}, \mathsf{pp}_{\mathsf{FE}}) \leftarrow\!\!\$\; \mathsf{FE.Setup}(\lambda)$,
(2) encrypts the input data: $\mathsf{ct}_j \leftarrow\!\!\$\; \mathsf{FE.Enc}(\mathsf{pp}_{\mathsf{FE}}, \mathsf{msk}_{\mathsf{FE}}, x_j)$ for all $j \in [d]$,
(3) determines the index set of allowed functions $I := \{i \mid \hat{f}_i \notin \mathcal{F}_A\}$,
(4) generate functional keys $\mathsf{fsk}_{\hat{f}_i} \leftarrow\!\!\$\; \mathsf{FE.KeyGen}(\mathsf{pp}_{\mathsf{FE}}, \mathsf{msk}_{\mathsf{FE}}, \hat{f}_i)$ for all $i \in I$.
(5) finally, sends $\mathsf{pp}_{\mathsf{FE}}$ and $\{\mathsf{ct}_j\}_{j\in[d]}$ to P_2.

Step III. P_1 and P_2 invoke the $\mathcal{F}_{OT^n_k}$-functionality:

(1) P_1 act as *sender* with κ messages as input: $m_i = \mathsf{fsk}_{\hat{f}_i}$ for $i \in I$ and $m_i = \bot$ for $i \in [\kappa] \setminus I$.
(2) P_2 act as *receiver* with input $(\sigma(1), \ldots, \sigma(k))$
(3) P_2 receives $m_{\sigma(1)}, \ldots, m_{\sigma(k)}$ where $m_{\sigma(i)} = \mathsf{fsk}_{f_i}$ or $m_{\sigma(i)} = \bot$ if it was not an allowed function (thus implicitly also obtaining the index set $I \cap [k]$).

Offline Phase

P_2 can evaluate the allowed functions from $\mathcal{F}_B$ on all input of P_1 by running $\mathsf{FE.Dec}(\mathsf{fsk}_{f_i}, \mathsf{ct}_j) = f_i(x_j)$ for all $i \in I \cap [k]$.

Figure 4.4: General rCPFE construction.

the incoming messages. First notice that $\mathsf{pp}_{\mathsf{FE}}$ and $\mathsf{pp}_{\mathsf{FE}}{}^*$ are generated with different random choices. At the same time, these cannot be told apart as otherwise the

[3] Depending on $\mathcal{F}$ and the sampling of the dummy functions, communication cost of transferring the function descriptions can be reduced. In §4.4.1, we describe such optimizations for the inner product function class.

choices were not random. The rest of the incoming messages depend on these parameters. Observe that $I^* = I \cap [k]$. The security of the used FE scheme guarantees that $(\mathsf{ct}_1^*, \ldots, \mathsf{ct}_d^*)$ even together with functional keys $\{\mathsf{fsk}^*_{f_i}\}_{i \in I^*}$ are indistinguishable from $(\mathsf{ct}_1, \ldots, \mathsf{ct}_d)$ with $\{\mathsf{fsk}_{f_i}\}_{i \in I \cap [k]}$. Finally, the security of the OT simulation guarantees that $(\mathsf{msg}_1^{\mathsf{OT}}, \ldots, \mathsf{msg}_\kappa^{\mathsf{OT}})$ and $(\mathsf{msg}_1^{\mathsf{OT}*}, \ldots, \mathsf{msg}_\kappa^{\mathsf{OT}*})$ are indistinguishable. This also implies that functional keys for the same functions (with respect to either $\mathsf{pp}_{\mathsf{FE}}$ or $\mathsf{pp}_{\mathsf{FE}}{}^*$) can be obtained both from the real and simulated OT messages. In other words, FE ciphertexts and functional keys are consistent in both cases (i.e. they allow one to obtain the same decryption outputs) due to the correctness of the FE simulation, which concludes our proof. □

Corollary 4.3.2.1. *The protocol of Fig. 4.4 also achieves* strong *κ-relaxed function privacy if in (1) of Step I., all f_i are sampled from the same distribution as the elements of $\mathcal{F}_B$ and* $\mathsf{aux} = \perp$.

4.4 Concrete Instantiation for Inner Products

To demonstrate the practicality of our approach, we instantiate our generic rCPFE protocol ($\Pi_{\mathcal{F}}^{\text{rCPFE}}$ of Fig. 4.4). For this, we use the k-NA-SIM secure FE scheme of [ALS16] for the inner product functionality and the semi-honest 1 out of κ OT protocol of [Tze04], which leads us to an inner product rCPFE protocol for $k = 1$. For the detailed description of the inner product rCPFE (or IP-rCPFE for short) we refer to Fig. 4.5. For simplicity, in our description we use the following notation: $g^{\vec{x}} = (g^{x_1}, \ldots, g^{x_\ell})$ for $g \in \mathbb{G}$ and $\vec{x} \in \mathbb{Z}_p^\ell$. $\mathcal{IE}$ denotes an efficient injective encoding algorithm mapping messages $m \in \{0,1\}^\lambda$ to elements of $\mathbb{G}$, so that m can be efficiently recovered from $\mathcal{IE}(m)$. For more details on injective encodings, we refer to [FJT13].

Theorem 4.3.2 and the assumptions of [ALS16, Tze04] directly imply the following theorem.

Theorem 4.4.1. *There is a SIM secure rCPFE protocol (according to Definition 6) for inner product computation, achieving κ-relaxed function privacy, if the DDH assumption holds.*

Corollary 4.4.1.1. *The inner product rCPFE protocol derived from $\Pi_{\mathcal{F}}^{rCPFE}$ (on Fig. 4.4) also achieves* strong *κ-relaxed function privacy (as defined in Definition 6) if* $\mathsf{aux} = \perp$ *and the dummy function vectors are chosen from the same distribution as the real ones.*

We note that in DDH-based inner product FE schemes only a polynomial sized range of the possible inner product results can be efficiently decrypted and our protocol inherits this property.

4.4.1 Performance and Possible Optimizations

For our IP-CPFE protocol, we prepared a proof of concept implementation using the Charm framework [AGM⁺13]. To evaluate its performance in two scenarios, we compared its running times and communication costs with that of the state of the art secure arithmetic inner product computation method of the ABY framework [DSZ15]. For our experiments we used a commodity laptop with a 2.60GHz Intel® Core™ i7-6700HQ CPU and 4GB of RAM.

Simulating regression model evaluation. In the first use-case, we do not assume that the vectors have a special structure. The vectors to be multiplied can correspond to data and weight vectors of a binary regression model, in which case it is likely that the same model (weight vector) is evaluated over multiple inputs. Fig. 4.6a and 4.6d depict running times and overall communication costs respectively depending on the number of inputs to the same model. Fig. 4.6c and 4.6f show the cost of the dummy queries. In the same setting, our experiments show that without optimizations[4] IP-rCPFE reaches the running time of ABY for $\kappa \approx 6200$. For this scenario, we also propose a method (denoted as rCPFE opt) to pre-compute the dummy function queries of Step I. thus reducing both the online communication and computation costs. The key insight of this is that sending a value together with dummy values is essentially the same as hiding the value with a one-time pad (OTP) and attaching the OTP key together with dummy keys. The gain comes from the fact that the OTP keys can be computed and sent beforehand, moreover it is enough to transmit the used seeds for a pseudo-random generator instead of the entire keys (see details in Fig. 4.7). Security is not affected as long as $\mathsf{aux} = \perp$.

Sparse vector products for location privacy. The location privacy scenario of §4.1.2 implies the usage of sparse query vectors. Fig. 4.6b and 4.6e show how the number of queries (k) affects running time and message sizes respectively, when roughly 5% of the vector elements are non-zero. We note that as queries are related to real-time user requests, batching these requests, as done in Step I. of the protocol, can be unrealistic when data vectors are not changing in real time but, e.g. periodically. Because of this, in our implementation, we allowed P_2 to repeat Step I. for a single function and P_1 to answer the queries independently of encrypting the data.[5] While sparsity disables the above optimization, after masking the places of non-zero elements, the above idea can be extended for sparse vectors as long as other structural properties are not known about the vector in form of auxiliary information. See Fig. 4.8.

4.5 Conclusion and Open Directions

In this work, we attempted to draw attention to the problem of possibly sensitive partial information leakage in the context of private function evaluation. We proposed a definitional framework for protocols that aim to prevent such leakage and showed both generic and concrete protocols to solve the problem. The main advantage of our FE-based pro-

[4] We note that while our implementation is only a proof of concept without any code level optimization, ABY has a very efficient and parallelizable implementation.

[5] It means that (3)–(4) of Step II., and Step III. are repeated until the input data changes at the end of the period.

tocol is that it turns the privacy sacrifice required by controllability into performance improvement whenever more function evaluations are necessary.

Our work also leaves open several problems for future work. For instance, it would be important to investigate the effects of having different types of auxiliary information about the evaluated functions. Transmission and verification of dummy functions can be serious bottlenecks in our rCPFE in case of complex functions, making further efficiency improvements desirable. A first step towards this could be to find a way for restricting the set of forbidden functions – as most often very simple functions are the only undesired ones. Finally, looking for different trade-offs between function privacy and efficiency can also be interesting direction for future work.

Protocol $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$

PARAMETERS: κ, λ parametrizing security and function class $\langle\cdot,\cdot\rangle : \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \to \mathbb{Z}_p$
Inputs:

- P_1: $\vec{x}_i = (x_{i1}, \ldots, x_{i\ell}) \in \mathbb{Z}_p^\ell$ for $i \in [d]$, $\mathcal{F}_A \subset \mathbb{Z}_p^\ell$
- P_2: $\vec{y} \in \mathbb{Z}_p^\ell$

OUTPUTS:

- P_1: $\mathcal{F}_R$ s.t. $\vec{y} \in \mathcal{F}_R \subset \mathbb{Z}_p^\ell$ and $|\mathcal{F}_R| = \kappa$
- P_2: $\langle x_i, y\rangle \in \mathbb{Z}_p$ if $x \notin \mathcal{F}_A$ and $\perp$ otherwise

PROTOCOL:

ONLINE PHASE

Step I. To initiate an inner product computation with $\vec{y}$, P_2 does the following:

(1) samples a random matrix $\mathbf{Y}_{\text{top}} \leftarrow\!\!\$\, \mathbb{Z}_p^{(\kappa-1)\times\ell}$ and appends $\vec{y}$ after the last row forming $\mathbf{Y} = (y_{i,j}) \in \mathbb{Z}_p^{\kappa\times\ell}$,

(2) picks a random permutation σ on κ elements to permute the rows of $\mathbf{Y}$ s.t. $\hat{\mathbf{Y}} = (\hat{y}_{i,j}) = (y_{\sigma^{-1}(i)j}) \in \mathbb{Z}_p^{\kappa\times\ell}$,

(3) finally, sends $\mathcal{F}_R = \hat{\mathbf{Y}}$ to P_1.

Step II. Upon receiving a function request $\mathcal{F}_R$, P_1

(1) chooses a group $\mathbb{G}$ of order a λ-bit prime p, with generators $g, h_0, h_1 \in \mathbb{G}$, selects an injective encoding $\mathcal{IE} : \mathbb{Z}_p^2 \to \mathbb{G}$, and set $\mathsf{pp} = (\mathbb{G}, p, g, h_0, h_1, \mathcal{IE}^{-1})$

(2) samples random $S \leftarrow\!\!\$\, \mathbb{Z}_p$ and $\vec{s}, \vec{t} \leftarrow\!\!\$\, \mathbb{Z}_p^\ell$ to form $\mathsf{msk}_{\mathsf{FE}} = (S, \vec{s}, \vec{t})$,

(3) $\forall j \in [d]$ samples random $r_j \leftarrow\!\!\$\, \mathbb{Z}_p$ and to encrypt $\vec{x}_j$, computes

$$\mathsf{ct}_j = \left(c_j = g^{r_j}, d_j = g^{Sr_j}, \vec{e}_j = g^{\vec{x}_j + r(\vec{s} + S\vec{t})}\right)$$

(4) determines the index set of allowed vector queries $I := \{i \mid \hat{\vec{y}}_i \notin \mathcal{F}_A\}$,

(5) generate functional keys for all $i \in I : \mathsf{fsk}_{\hat{\vec{y}}_i} = (\hat{S}_i = \langle\hat{\vec{y}}_i, \vec{s}\rangle, \hat{T}_i = \langle\hat{\vec{y}}_i, \vec{t}\rangle)$,

(6) sends pp and $\{\mathsf{ct}_j\}_{j\in[d]}$ to P_2

Step III. P_1 and P_2 executes the following 1 out of κ OT protocol:

(1) P_2 samples random $r' \leftarrow\!\!\$\, \mathbb{Z}_p$, computes $R' = h_0^{r'} h_1^{\sigma(\kappa)}$ and sends R' to P_1.

(2) for $i \in [\kappa], P_1$ samples $k_i \leftarrow\!\!\$\, \mathbb{Z}_p$, prepares m_i s.t. $m_i = \mathsf{fsk}_{\hat{\vec{y}}_i}$ if $i \in I$ and $m_i = \perp$ for $i \notin I$, then computes the OT messages to be sent to P_1:

$$\mathsf{msg}_{\mathsf{i}}^{\mathsf{OT}} = (a_i = g^{k_i}, b_i = \mathcal{IE}(m_i) \cdot (R'/h^i)^{k_i}).$$

OFFLINE PHASE

P_2 can evaluate the inner products by executing the following steps:

(1) to extract the functional key from the OT messages, select $\mathsf{msg}_{\sigma(\kappa)}^{\mathsf{OT}}$ and compute

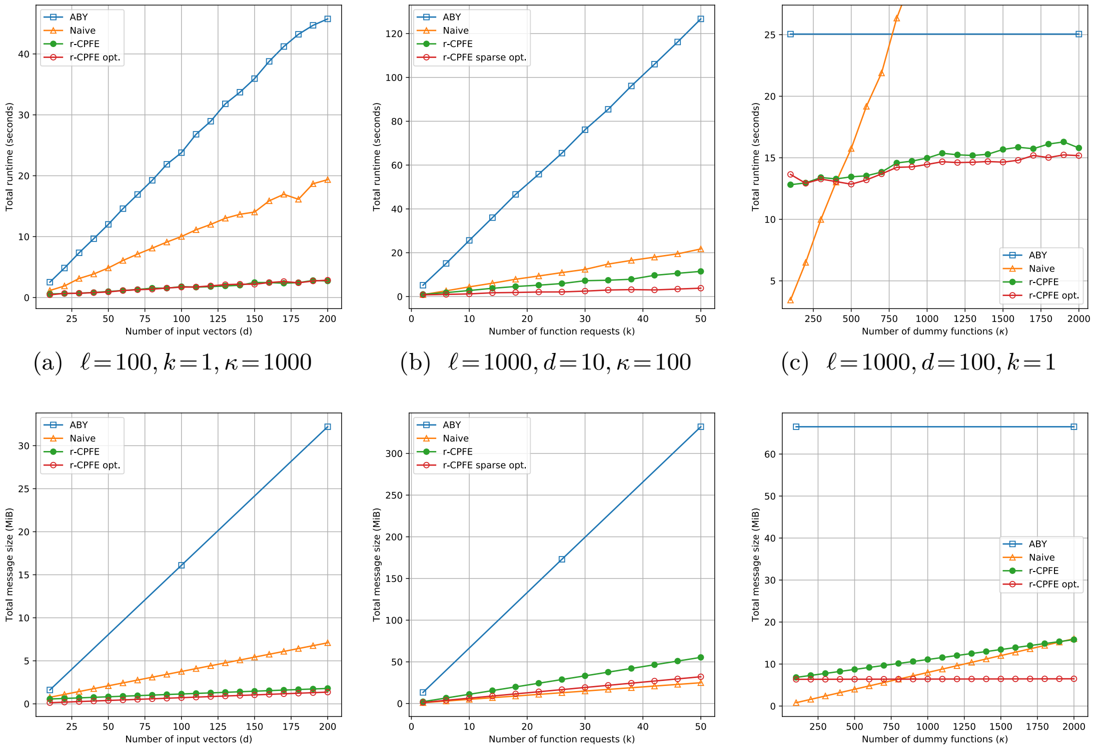
$$\mathcal{IE}^{-1}\left(b_{\sigma(\kappa)}/a_{\sigma(\kappa)}^{r'}\right) = \mu$$

(2) if $\mu = \perp$ then output $\perp$, otherwise $\mu = \mathsf{fsk}_{\hat{\vec{y}}_{\sigma(\kappa)}} = \mathsf{fsk}_{\vec{y}} = (S_\kappa, T_\kappa)$,

(3) $\forall j \in [d]$ compute $(\prod_{i\in[\ell]} e_{ji}^{y_{ji}})/(c_j^{S_{\sigma(\kappa)}} \cdot d_j^{T_{\sigma(\kappa)}}) = g^{\langle\vec{x},\vec{y}_j\rangle}$,

(4) if the discrete log of $g^{\langle\vec{x},\vec{y}_j\rangle}$ is contained in a predetermined range, it is computed and returned as the output, otherwise $\perp$ is returned.

Figure 4.5: An instantiation of the generic rCPFE construction for the inner product functionality.



(a) $\ell = 100, k = 1, \kappa = 1000$

(b) $\ell = 1000, d = 10, \kappa = 100$

(c) $\ell = 1000, d = 100, k = 1$

(d) $\ell = 100, k = 1, \kappa = 1000$

(e) $\ell = 1000, d = 10, \kappa = 100$

(f) $\ell = 1000, d = 100, k = 1$

Figure 4.6: Comparisons of the overall running times (4.6a–4.6c) and communication costs (4.6d–4.6f) of our rCPFE protocols with the ABY framework [DSZ15] and the naive OT-based approach for inner product computation (ℓ denotes vector dimension, d and k are the number of input and "function" vectors, while κ is the number of dummy vectors).

Protocol $\Pi^{\text{rCPFE opt}}_{\langle\cdot,\cdot\rangle}$

Parameters: κ, λ parametrizing security, function class $\langle\cdot,\cdot\rangle : \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \to \mathbb{Z}_p$, and a pseudo-random number generator $\mathsf{PRG} : \{0,1\}^\vartheta \to \{0,1\}^{\log_2 p}$

Inputs:

- P_1: $\vec{x}_i = (x_{i1}, \dots, x_{i\ell}) \in \mathbb{Z}_p^\ell$ for $i \in [d]$, $\mathcal{F}_A \subset \mathbb{Z}_p^\ell$
- P_2: $\vec{y} \in \mathbb{Z}_p^\ell$

Protocol:

Precomputation

Pre-Step I. P_2 samples $q'_{i,j} \leftarrow\!\$\, \{0,1\}^\vartheta$ for the PRG to compute $q_{i,j} = \mathsf{PRG}(q'_{i,j}), \forall i \in [n],\ j \in [\ell]$. $\mathbf{Q} = (q_{i,j})$ is kept locally, while $\mathbf{Q}' = (q'_{i,j})$ is sent to P_1.

Pre-Step II. P_1 computes $\mathbf{Q} = (q_{i,j})$ by evaluating $\mathsf{PRG}(q'_{i,j}) = q_{i,j}, \forall i \in [n], j \in [\ell]$.

Online Phase

Step I. To generate function request $\mathcal{F}_R$ containing $\vec{y}$, P_2

(1) chooses a random index $i^* \leftarrow\!\$\, [n]$,

(2) encrypts $\vec{y}$ with the i^*th OTP key: $\vec{y}' = (y_1 \oplus q_{i^*,1}, \dots, y_\ell \oplus q_{i^*,\ell})$,

(3) finally, sends $\vec{y}'$, as a succinct description of $\mathcal{F}_R$, to P_1.

Step II. Upon receiving a function request $\mathcal{F}_R$, P_1

(0) decrypts $\mathcal{F}_R$ with all the OTP keys: $\vec{\tilde{y}}_i = (y'_1 \oplus q_{i,1}, \dots, y'_\ell \oplus q_{i,\ell}), \forall i \in n$

(1)-(6) executes Step II. of $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$ (Fig. 4.5).

Step III. P_1 and P_2 executes the following 1 out of κ OT protocol:

(1) P_2 samples random $r' \leftarrow\!\$\, \mathbb{Z}_p$, computes $R' = h_0^{r'} h_1^{i^*}$ and sends R' to P_1.

(2) P_1 acts as in Step III. of $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$ (Fig. 4.5).

Offline Phase

It is the same as in $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$ (Fig. 4.5).

Figure 4.7: "OTP" optimization for the inner product rCPFE in case of uniformly distributed function vectors of limited size.

Protocol $\Pi^{\text{rCPFE sparse opt}}_{\langle\cdot,\cdot\rangle}$

Parameters: κ, λ parametrizing security, function class $\langle\cdot,\cdot\rangle : \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \to \mathbb{Z}_p$, and a pseudo-random number generator $\mathsf{PRG} : \{0,1\}^{\vartheta} \to \{0,1\}^{\log_2 p}$

Inputs:

- P_1: $\vec{x}_i = (x_{i1}, \ldots, x_{i\ell}) \in \mathbb{Z}_p^\ell$ for $i \in [d]$, $\mathcal{F}_A \subset \mathbb{Z}_p^\ell$
- P_2: $\vec{y} \in \mathbb{Z}_p^\ell$

Protocol:

Precomputation

Pre-Step I. P_2 does the following:

(1) samples PRG seed $q'_{i,j} \leftarrow_{\$} \{0,1\}^{\vartheta}$ to compute $q_{i,j} = \mathsf{PRG}(q'_{i,j}), \forall i \in [n], j \in [\ell]$,

(2) selects n random permutations $\Sigma = (\sigma_i, \ldots, \sigma_n)$, each on ℓ elements,

(3) $\mathbf{Q} = (q_{i,j})$ is kept locally, while Σ and $\mathbf{Q}' = (q'_{i,j})$ are sent to P_1.

Pre-Step II. P_1 computes $\mathbf{Q} = (q_{i,j})$ by evaluating $\mathsf{PRG}(q'_{i,j}) = q_{i,j}, \forall i \in [n], j \in [\ell]$.

Online Phase

Step I. To generate request $\mathcal{F}_R$ for sparse $\vec{y}$ (containing $\delta \ll \ell$ non-zero values), P_2

(1) prepares the list of non-zero vector positions $(j_1, \ldots, j_\delta)$ s.t. $y_{j_m} \neq 0$, $\forall m \in [\delta]$,

(2) chooses a random index $i^* \leftarrow_{\$} [n]$,

(3) computes $Y := \{(a_m = \sigma_{i^*}(j_m), b_m = y_{j_m} \oplus q_{i^*,\sigma_{i^*}(j_m)})\}_{m \in [\delta]}$, where $\sigma_{i^*} \in \Sigma$,

(4) finally, sends Y, as a succinct description of $\mathcal{F}_R$, to P_1.

Step II. Upon receiving a function request $\mathcal{F}_R$, P_1

(0) prepares $\vec{\hat{y}}_i$ vectors by setting $\hat{y}_{i,\sigma_i^{-1}(a_m)} = b_m \oplus q_{i,a_m}$, $\forall\ m \in [\delta]$ and $\hat{y}_{i,j} = 0$ otherwise,

(1)-(6) executes Step II. of $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$ (Fig. 4.5).

Step III. P_1 and P_2 executes the following 1 out of κ OT protocol:

(1) P_2 samples random $r' \leftarrow_{\$} \mathbb{Z}_p$, computes $R' = h_0^{r'} h_1^{i^*}$ and sends R' to P_1.

(2) P_1 acts as in Step III. of $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$ (Fig. 4.5).

Offline Phase

It is the same as in $\Pi^{\text{rCPFE}}_{\langle\cdot,\cdot\rangle}$ (Fig. 4.5).

Figure 4.8: "OTP" optimization for rCPFE for the inner product functionality in case of sparse function vectors with uniformly distributed nonzero values.

Chapter 5

User Revocation in Multi-Authority Attribute-Based Encryption

5.1 Introduction

Besides cost savings, the promise of increasing flexibility is the main driving force for outsourcing data storage. On the other hand, giving out data raises the issue of privacy and security, which leads us to the necessity of encryption. Traditional cryptosystems were designed to confidentially transmit data to a target recipient (e.g. from Alice to Bob) and this seems to restrict the range of opportunities and flexibility offered by a cloud environment. Imagine the following scenario: some companies are cooperating on a cryptography project and from each, employees are working together. Suppose that Alice wants to share some data with those who are working on a specific subtask, and with the managers of the project from the different companies. Encryption of the data with traditional techniques, causes that recipients must be determined formerly, moreover either they have to share the same private key or several encrypted versions (with different keys) must be stored. These undermine security, efficiency and also flexibility, which the cloud should provide.

Attribute-based encryption (ABE), proposed by Sahai and Waters [SW05], is intended for one-to-many encryption in which ciphertexts are encrypted for those who are able to fulfil certain requirements, i.e. their attributes satisfy a given policy. ABE schemes have two main branches depending on role of attributes and policies in the system. One of them[1] is called CP-ABE, where ciphertexts are associated with access policies, determined by the encryptor, and attributes describe the user, accordingly attributes are embedded in the users' secret keys. A ciphertext can be decrypted by someone if and only if, his or her attributes satisfy the access structure given in the ci-

[1] The other branch is called key-policy ABE, where the role of attributes and policies are the opposite, compared to CP-ABE.

phertext. Note that using ABE, data sharing is possible without prior knowledge of who will be the receiver that can preserve the flexibility of the cloud even after encryption.

Returning to the previous example, using CP-ABE Alice can encrypt with an access policy expressed by the following Boolean formula: "CryptoProject" AND ("Subtask Y" OR "Manager"). Uploading the ciphertext to the cloud, it can be easily accessed by the employees of each company, but the data can be recovered only by those who own a set of attributes in their secret keys which satisfies the access policy (e.g. "CryptoProject", "Subtask Y"). Flexible identification of user groups has its own price that we have to pay when an individual user has to be identified. The typical example, when we have to do this is user revocation. In everyday use, a tool for changing a user's rights is essential as unexpected events may occur and affect these. An occasion when someone has to be revoked can be dismissal or the revealing of malicious activity. Revocation is especially hard problem in ABE, since different users may hold functionally the same secret keys related with the same attribute set (aside from randomization). We emphasise that user revocation is applied in *exceptional cases* like the above-mentioned, as all other cases can be handled simpler, with the proper use of attributes (e.g. an attribute can include its planned validity like "CryptoProject2020").

The example also shows that attributes or credentials issued across different trust domains are essential in practice and these have to be verified inside the different organisations (e.g. "Manager" attribute). Wile "textbook" ABE schemes can only handle one central attribute authority, there exist multi-authority ABE variants. In [C5, J2] we were looking for answers for the following question.

Is it possible to realize efficient user revocation in the multi-authority CP-ABE setting without updating the keys of users who had common attribute secret keys with the revoked user?

5.1.1 Contributions

Building on the results of [LW11] and [LSW10], we propose a scheme that adds identity-based user revocation feature to distributed CP-ABE. With this extension, we achieve a scheme with multiple, independent attribute authorities, in which revocation of specific users (with a given identifier) from the system with all of their attributes is possible without updates of attribute public and secret keys (neither periodically, nor after revocation event). We avoid re-encryption of all ciphertexts the access structures of which contain a subset of attributes of the revoked user. The revocation right can be given directly to the encryptor, just like the right to define the access structure which fits to the cloud computing and data market scenarios. We prove the security of our construction in the generic bilinear group and ROM models.

To build our scheme, we use the prime order group construction of Lewko and Waters [LW11], because of its favourable property of having independent attribute authorities. In order to achieve direct, a.k.a. identity-based revocation, we supplement the distributed system with a central authority. Although it seems to contradict with the original aim of distributing the key generation right, this additional authority would generate only secret keys for global identifiers ($GID \in \mathbb{Z}_p$) of users and the attribute key generation remains distributed. Our central authority does not possess any information that alone would give advantage during decryption, in contrast to single authority schemes, where the authority is able to decrypt all ciphertexts. Regarding this, we can say that our system remains distributed, in spite of launching a central authority.

5.1.2 Applications

Both revocable and distributed CP-ABE schemes can enhance flexible access control in cloud-based secure data storage. Such "optimized" CP-ABE could hide fresh symmetric keys, which are used to efficiently encode large amounts of data, and reveal them only for authorized users, who can be identified through expressive access policies. The sketched cloud storage scenario includes scenario $\left[\substack{\mathsf{U0}\\ \mathsf{T1101}}\right]$ from the data market problem (see Fig. 5.1 for the application of our scheme in that scenario). In that specific use-case, the DO can control the access rights to his or her data through the use of CP-ABE. Having a central role in the data market, the DB is capable of being the central authority and provide identity keys[2] for the VASPs upon sign up to the data market. Maintenance and publication of a revocation list can also be part of the DB's responsibilities. Attribute authorities can be run by any entities, that are independent of the market (i.e. has no interest of obtaining data, sold in the market) and can be trusted by the DOs to provide authentic information about the VASPs. Examples for such entities may include regulatory bodies, certification authorities, but also non-governmental organizations, consumer protection offices, etc. Our revocable multi-authority CP-ABE enables DOs to encrypt their data without prior knowledge of who exactly will decrypt it. At the same time, constraints on the possible decryptor can be determined in encryption time (through the access control policy) based on attributes of VASPs, which were recognised by independent authorities.

In case of cloud storage, updates to the encrypted data may occur. In this context, lazy re-encryption of the ciphertexts is immediately provided, as long as fresh symmetric keys and the up-to-date revocation list are used for encryption, whenever some data is updated.

5.2 Related Work

The concept of ABE was first proposed by Sahai and Waters [SW05] as a generalization of identity-based encryption. Bethencourt et al. [BSW07] worked out the first CP-ABE scheme in which the encryptor must decide who should or should not have access to the data that she encrypts (ciphertexts are associated with policies, and users' keys are associated with sets of descriptive attributes). This concept was further improved by Waters in [Wat11].

[2] As the information, needed for the generation of the identity keys, is not enough for the decryption of ciphertexts, a DB or a cloud service provider can securely incorporate the task of the central authority without any conflict of interest. See the details in §5.3.3.

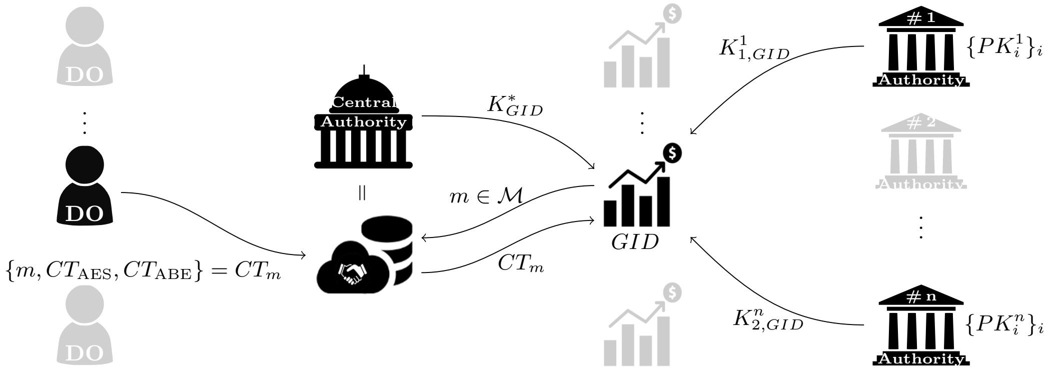


Figure 5.1: Access control in data markets (fulfiling the requirements depicted in Fig. 3.2a corresponding to scenario $[{}^{\mathsf{U0}}_{\mathsf{T1101}}]$) using our revocable multi-authority CP-ABE scheme. The figure represents the interactions of a single DO and VASP (inactive participants are represented in grey), where $m \in \mathcal{M}$ represents some metadata, CT_m is the ciphertext corresponding to the data belonging to m. CT_{AES} is an encryption of the plaintext data, using a symmetric-key cipher (e.g. AES) with a fresh key, and CT_{ABE} is the encryption of the symmetric key, using our scheme with an access policy, the necessary attribute public keys and the up-to-date revocation list RL.

The problem of building ABE systems with multiple authorities was first considered by Chase [Cha07] with a solution that introduced the concept of using a global identifier (GID) for tying users' keys together. Her system relied on a central authority and was limited to expressing a strict AND policy over a pre-determined set of authorities. Decentralized ABE of Lewko and Waters [LW11] does not require any central authority and any party can become an authority while there is no requirement for any global coordination (different authorities need not even be aware of each other) other than the creation of an initial set of common reference parameters. With this it avoids placing absolute trust in a single designated entity, which must remain active and uncorrupted throughout the lifetime of the system. Several other multi-authority schemes (e.g. [RNS11, WLWG11]) were shaped to the needs of cloud computing, although these lack for efficient user revocation.

Attribute revocation with the help of expiring attributes was proposed by Bethencourt et al. [BSW07]. For single authority schemes Sahai et al. [SSW12] introduced methods for secure delegation of tasks to third parties and user revocation through piecewise key generation. Ruj et al. [RNS11], Wang et al. [WLWG11] and Yang et al. [YJRZ13] show traditional attribute revocation (in multi-authority setting) causing serious computational overhead, because of the need for key re-generation and ciphertext re-encryption. A different approach is identity-based revocation, two types of which were applied to the scheme of Waters [Wat11]. Liang et al. [LLLS10] gives the right of controlling the revoked set to a "system manager" while Li et al. [LZW$^+$13], follow [LSW10], from the field of broadcast encryption systems and give the revocation right directly to the encryptor. This later was further developed by Li et al. [LXZ13] achieving full security with the help of dual system encryption. For this approach, but for KP-ABE, Qian and Dong [QD11] showed fully secure solution.

To the best of our knowledge no multi-authority system is integrated with identity-based user revocation and our works [C5, J2] were the first ones in this direction.

5.3 Revocation Scheme for Multi-Authority CP-ABE

5.3.1 Algorithms of Revocable Multi-Authority CP-ABE

A multi-authority CP-ABE system with identity-based user revocation is comprised of the following algorithms:

Global Setup$(\lambda) \to GP$

The global setup algorithm takes in the security parameter λ and outputs global parameters GP for the system.

Central Authority Setup$(GP) \to (SK^*, PK^*)$

The central authority (CA) runs this algorithm with GP as input to produce its own secret key and public key pair, SK^*, PK^*.

Identity KeyGen$(GP, RL, GID, SK^*) \to K^*_{GID}$

The CA runs this algorithm upon a user request for identity secret key. It checks whether the request is valid and if yes (i.e. the user's global identifier, denoted by GID, is not part of the revocation list RL: $GID \notin RL$), generates K^*_{GID} using the global parameters and the secret key of the CA.

Authority Setup$(GP) \to (PK, SK)$

Each attribute authority runs the authority setup algorithm with GP as input to produce its own secret key and public key pair, SK, PK.

KeyGen$(GP, SK, GID, i) \to K_{i,GID}$

The attribute key generation algorithm takes in an identity GID, the global parameters, an attribute i belonging to some authority, and the secret key SK for this authority. It produces a key $K_{i,GID}$ for this attribute-identity pair.

Encrypt$(GP, \mathfrak{m}, (A, \rho), \{PK\}, PK^*, RL) \to CT$

The encryption algorithm takes in a message $\mathfrak{m}$, an access matrix (A, ρ), the set of public keys for relevant authorities, the public key of the central authority, the revoked user list and the global parameters. It outputs a ciphertext CT.

Decrypt$(GP, CT, (A, \rho), \{K_{i,GID}\}, K^*_{GID}, RL) \to \mathfrak{m}$

The decryption algorithm takes in the global parameters, the revoked user list, the ciphertext, identity key and a collection of keys corresponding to attribute, identity pairs all with the same fixed identity GID. It outputs either the message $\mathfrak{m}$ when the collection of attributes i satisfies the access matrix corresponding to the ciphertext. Otherwise, decryption fails.

5.3.2 Security Model

We now define (chosen plaintext) security of multi-authority CP-ABE system with identity-based revocation. Security is defined through a *security game* between an attacker algorithm $\mathcal{A}$ and a challenger. We assume that adversaries can corrupt authorities only statically, but key queries are made adaptively. The definition reflects the scenario where all users in the revoked set RL get together and collude (this is because the adversary can get all of the private keys for the revoked set). Informally, $\mathcal{A}$ can determine a set of corrupted attribute authorities, ask for any identity and attribute keys and specify messages, on which it will be challenged using the revocation list and access matrix of its choice. The only (natural) restriction in the above choices is that $\mathcal{A}$ cannot ask for a set of keys that allow decryption, in combination with any keys that can be obtained from corrupt authorities in case of a non revoked GID_k. In case of revoked identities we can be less restrictive: corrupted attributes alone cannot satisfy the access policy, but it might be satisfied together with attributes from honest authorities. $\mathcal{A}$ wins the game if it respects the rules and can decide which of its challenge messages were encrypted by the challenger. The formal security game consists of the following rounds:

Setup. The challenger runs the Global Setup algorithm to obtain the global public parameters GP. $\mathcal{A}$ specifies a set $AA' \subseteq AA$ of corrupt attribute authorities and uses the Authority Setup to obtain public and private keys. For honest authorities in $AA \backslash AA'$ and for the Central Authority, the challenger obtains the corresponding keys by running the Authority Setup and Central Authority Setup algorithms, and gives the public keys to the attacker.

Key Query Phase. $\mathcal{A}$ adaptively issues private key queries for identities GID_k (which denotes the k^{th} GID query). The challenger gives $\mathcal{A}$ the corresponding identity keys $K^*_{GID_k}$ by running the Identity KeyGen algorithm. Let UL denote the set of all queried GID_k. $\mathcal{A}$ also makes attribute key queries by submitting pairs of (i, GID_k) to the challenger, where i is an attribute belonging to a good authority. The challenger responds by giving the attacker the corresponding key, K_{i,GID_k}.

Challenge. The attacker gives the challenger two messages $\mathfrak{m}_0, \mathfrak{m}_1$, a set $RL \subseteq UL$ of revoked identities and an access matrix (A, ρ).

RL and A must satisfy the following constraints. Let V denote the subset of rows of A labelled by attributes controlled by corrupt authorities. For each identity $GID_k \in UL$, let V_{GID_k} denote the subset of rows of A labelled by attributes i for which the attacker has queried (i, GID_k). For each $GID_k \in UL \setminus RL$, we require that the subspace spanned by $V \cup V_{GID_k}$ must not include $(1, 0, \dots, 0)$ while for $GID_k \in RL$, it is allowed and we only require that the subspace spanned by V must not include $(1, 0, \dots, 0)$.

The attacker must also give the challenger the public keys for any corrupt authorities whose attributes appear in the labelling ρ.

The challenger flips a random coin $\beta \in (0, 1)$ and sends the attacker an encryption of M_β under access matrix (A, ρ) with the revoked set RL.

Key Query Phase 2. The attacker may submit additional attribute key queries (i, GID_k), as long as they do not violate the constraint on the challenge revocation list RL and matrix (A, ρ).

Guess. $\mathcal{A}$ must submit a guess β' for β. The attacker wins if $\beta' = \beta$. The attacker's advantage in this game is defined to be $\mathbb{P}(\beta' = \beta) - \frac{1}{2}$.

Definition 7. *We say that a multi-authority CP-ABE system with identity-based revocation is (chosen-plaintext) secure (against static corruption of attribute authorities) if, for all revocations sets RL of size polynomial in the security parameter, all polynomial time adversary has at most a negligible advantage in the above defined security game.*

5.3.3 Extension of LW11 with Direct Revocation

Intuition

We face with the challenges of identity-based revocation. To realize the targeted features, we use some ideas from public key broadcast encryption systems [LSW10][3]. We use secret sharing in the exponent. Suppose an encryption algorithm needs to create a ciphertext in the presence of r revoked identities, contained in the revocation set $RL = GID_1^*, \ldots, GID_r^*$. The algorithm will create an exponent $s^* \in \mathbb{Z}_p$ and split it into r random shares $s_1, \ldots, s_r$ such that $\sum_{k=1}^{r} s_k = s^*$. It will then create a ciphertext such that any revoked user with GID_k^* will not be able to incorporate the k^{th} share and thus cannot decrypt the message.

This approach presents the following challenges. First, we need to make inevitable that the decryptor needs to do the GID comparisons even if his attributes satisfy the access structure of the ciphertext. Second, we need to make sure that no revoked user with GID_k^* can obtain any information about share s_k. Third, we need to worry about collusion attacks between multiple revoked users.

To address the first challenge, we are going to take advantage of the technique of [LW11] that is used to prevent collusion attacks. Here the secret s, used for the encryption, is divided into shares, which are further blinded with shares of zero. This structure allows for the decryption algorithm to both reconstruct the main secret and to "unblind" it in parallel. If a user with a particular identifier GID satisfies the access structure, he can reconstruct s in the exponent by raising the group elements to the

[3] Cao and Liu [CL14] points out an inherent drawback of the [LSW10] scheme, namely that for malicious users it is worth to exchange their decryption keys in order to maximize their interests. However, we utilize similar techniques as [LSW10], our system is not vulnerable to this kind of misuse. The reason for this is that unlike in broadcast encryption, where having a non-revoked secret key is the only requirement for decryption, in ABE multiple attribute secret keys are required for decryption. Keys that were issued for different users cannot be used together thus limiting the possible benefit of collusion. We also note that the flaw of [LSW10]'s security proof, claimed by [CL14] does not affect our results, as we use different proof technique.

proper exponents. This operation will simultaneously reconstruct the share of 0 and thus the $e(H(GID), g)$ blinding terms will cancel out. When we would like to make this algorithm necessary but not enough for decryption, it is straightforward to spoil the “unblinding” of the secret by changing the shares of zero in the exponent to shares of another random number, $s^* \in \mathbb{Z}_p$. Thus we can require an other computation, namely the comparison of the decryptor's and the revoked users' GIDs. If correspondence is found, the algorithm stops, otherwise reveals the blinding, enabling decryption.

The second challenge is addressed by the following method. A user with $GID \neq GID_k^*$ can obtain two linearly independent equations (in the exponent) involving the share s_k, which he will use to solve for the share s_k. However, if $GID = GID_k^*$, the obtained equations are going to be linearly dependent and the user will not be able to solve the system.

In the third case, the attack we need to worry about is where a user with GID_k^* processes ciphertext share l, while another user with GID_l^* processes share k, and then they combine their results. To prevent collusion, we use $H(GID)$ as the base of the identity secret key, such that during decryption each user recovers shares $s_k \cdot \log_g H(GID)$ in the exponent, disallowing the combination of shares from different users.

Our Construction

To make the following notions more understandable, in Table 5.1 we summarize the new keys and variables (compared to [LW11]) which we introduce in our construction. Based on the above principles, the proposed algorithms are the following:

Notation	Meaning	Role
PK^*	$\{g^a, g^{1/b}\}$	public key of the Central Authority
SK^*	$\{a, b\}$	secret key of the Central Authority
K_{GID}^*	$H(GID)^{(GID+a)b}$	global identity secret key of a user
$C_{1,k}^*$	$\left(g^a g^{GID_k^*}\right)^{-s_k}$	revoked user identification in CT
$C_{2,k}^*$	$g^{s_k/b}$	k^{th} secret share in the CT
RL	$\{GID_1^*, \dots, GID_r^*\}$	list of r revoked users

Table 5.1: The summary of our new notations compared to [LW11].

Global Setup$(\lambda) \rightarrow GP$

In the global setup, a bilinear group $\mathbb{G}_0$ of prime order p is chosen. The global public parameters, GP, are p and a generator g of $\mathbb{G}_0$, and a function H mapping global identities $GID \in \mathbb{Z}_p$ to elements of $\mathbb{G}_0$ (this is modelled as a random oracle in the security proof).

Central Authority Setup$(GP) \rightarrow (SK^*, PK^*)$

The algorithm chooses random exponents $a, b \in \mathbb{Z}_p$, keeps them as secret key

$SK^* = \{a, b\}$ and publishes $PK^* = \{g^a, g^{1/b}\}$.

Identity KeyGen$(GP, RL, GID, SK^*) \rightarrow K^*_{GID}$

Upon the request of a user it first checks whether the user is on the list of revoked users (RL) or it has been queried before, if yes refuses the request, otherwise computes $H(GID)$ and generates the global identity secret key:

$$K^*_{GID} = H(GID)^{(GID+a)b}.$$

Authority Setup$(GP) \rightarrow (PK, SK)$

For each attribute i belonging to the authority (these indices i are not reused between authorities), the authority chooses two random exponents $\alpha_i, y_i \in \mathbb{Z}_p$ and publishes $PK = \{e(g,g)^{\alpha_i}, g^{y_i}\ \forall i\}$ as its public key. It keeps $SK = \{\alpha_i, y_i\ \forall i\}$ as its secret key.

KeyGen$(GP, SK, GID, i) \rightarrow K_{i,GID}$

To create a key for a GID, for attribute i belonging to an authority, the authority computes:

$$K_{i,GID} = g^{\alpha_i} H(GID)^{y_i}$$

Encrypt$(GP, \mathfrak{m}, (A, \rho), \{PK\}, PK^*, RL) \rightarrow CT$

The encryption algorithm takes in a message $\mathfrak{m}$, an $n \times \ell$ access matrix A with ρ mapping its rows to attributes, the global parameters, the public keys of the relevant authorities, the user identity public key and the most recent list of revoked users.

It chooses random $s, s^* \in \mathbb{Z}_p$ and a random vector $v \in \mathbb{Z}_p^\ell$ with s as its first entry. Let λ_x denote $A_x \cdot v$, where A_x is row x of A. It also chooses a random vector $w \in \mathbb{Z}_p^\ell$ with s^* as its first entry. Let ω_x denote $A_x \cdot w$.

For each row A_x of A, it chooses a random $r_x \in \mathbb{Z}_p$ and supposed that the number of revoked users is $|RL| = r$ it chooses s_k such that $s^* = \sum_{k=1}^{r} s_k$. The CT ciphertext is computed as

$$C_0 = \mathfrak{m} \cdot e(g,g)^s,$$
$$C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\rho(x)} r_x},$$
$$C_{2,x} = g^{r_x}, \quad C_{3,x} = g^{y_{\rho(x)} r_x} g^{\omega_x},$$
$$C^*_{1,k} = \left(g^a g^{GID^*_k}\right)^{-s_k}, \quad C^*_{2,k} = g^{s_k/b}$$

for all $x = 1, \ldots, n$ and $k = 1, \ldots, r$.

Decrypt$(GP, CT, (A, \rho), \{K_{i,GID}\}, K^*_{GID}, RL) \rightarrow \mathfrak{m}$

We assume the ciphertext is encrypted under an access matrix (A, ρ). If the decryptor is not on the list of revoked users (RL) and has the secret keys K^*_{GID} for his GID and $\{K_{i,GID}\}$ for a subset of rows A_x of A, such that $(1, 0, \ldots, 0)$ is in the span of these rows, then the decryptor proceeds as follows. First chooses constants $c_x \in \mathbb{Z}_p$ such that $\sum_x c_x A_x = (1, 0, \dots, 0)$ and denoting $r = |RL|$ computes:

$$\frac{\mathscr{A}}{\mathscr{B}} = \frac{\prod_x \left(\frac{C_{1,x} \cdot e(H(GID), C_{3,x})}{e(K_{\rho(x),GID}, C_{2,x})} \right)^{c_x}}{\prod_{k=1}^{r} \left(e(K^*_{GID}, C^*_{2,k}) e(C^*_{1,k}, H(GID)) \right)^{1/(GID - GID^*_k)}}$$

which equals to $e(g,g)^s$, so the message can be obtained as $\mathfrak{m} = C_0 / e(g,g)^s$.

To see the soundness of the Decryption algorithm observe that after substituting the corresponding values we get the following:

$$\begin{aligned}
\mathscr{A} &= \prod_x \left(e(g,g)^{\lambda_x + \omega_x \log_g H(GID)} \right)^{c_x} \\
&= e(g,g)^{\sum_x \lambda_x c_x} \cdot e(H(GID), g)^{\sum_x \omega_x c_x} \\
&= e(g,g)^{s + s^* \log_g H(GID)} \\
\mathscr{B} &= \prod_{k=1}^{r} \left(e(g,g)^{(GID - GID^*_k) s_k \log_g H(GID)} \right)^{1/(GID - GID^*_k)} \\
&= e(g,g)^{-\sum_{k=1}^{r} s_k \log_g H(GID)} = e(g,g)^{s^* \log_g H(GID)}
\end{aligned}$$

Remark 1. *We note that alternatively, it is also possible to give revocation right directly to the encryptor by simply publishing a user list instead of RL. In this case, RL would be defined by the user, separately for each ciphertext, and attached to CT.*

5.4 Evaluation

5.4.1 Performance

Traditional, attribute-based user revocation (e.g. [WLWG11, RNS11, YJRZ13]) affects attributes, thus the revocation of a user may cause the update of all the users' attribute secret keys who had common attribute with the revoked user and a general attribute can affect big proportion of the users. Moreover, if re-encryption is applied after revocation, all ciphertext will be affected that contain any of the revoked user's attributes in their access structure.

In our scheme, a revocation event does not have any effect on the attributes as it is based on identity. On the other hand, it is a trade-off as it has some computational overhead on the encryption and decryption algorithms. The necessary extra computation of authorities (in case of attribute-based revocation) is reduced and distributed between the largest set of parties, the users, preventing a possible performance bottleneck of the system. At the same time, the extra communication is also reduced to the publication of the revoked user list. Our revocation scheme has the following costs.

The ciphertext has $2r$ additional elements, if the number of revoked users is r. For the computation of these values $3r$ exponentiations and r multiplications are needed in $\mathbb{G}_0$. Alternatively, the revoked user list may contain $g^a g^{GID_i^*}$ instead of the global identifiers. In this case the encryptor need to do only $2r$ additional exponentiations in $\mathbb{G}_0$, compared with the scheme of [LW11], to compute the ciphertext. The overhead of the decryption algorithm is $2r$ pairing operations, r multiplications and exponentiations in group $\mathbb{G}_1$.

Note that, as in all model that uses LSSS to express the access structure, the access matrix and the mapping ρ must be part of the ciphertext, increasing its length. However, it is possible to reduce this length by attaching only a formatted Boolean formula instead and compute the necessary components of LSSS more efficiently, using the algorithm of Liu and Cao from [LC10].

5.4.2 Security Analysis

Before giving the formal proof, we point out that from a user's the point of view, whose attributes have never satisfied the access structure defined in the ciphertext, our construction is at least as secure as the one by [LW11], because the computation of $\mathscr{A}$ is equivalent to the decryption computation given there. However in our case, it is not enough to obtain the message. Changing the first entry of the blinding vector w from zero to a random number (as we did), causes that the blinding will not cancel out from $\mathscr{A}$, but we need to compute $\mathscr{B}$ to remove the masking. $\mathscr{B}$ can be computed using GIDs different from the GID^*s of the revocation list and we ensure that the decryptor must use the same GID in both $\mathscr{A}$ and $\mathscr{B}$ by incorporating $H(GID)$ in both the identity and attribute secret keys.

We are going to prove the security of our construction in the generic bilinear group model (a variant of GGM) previously used in [BBG05, BSW07, LW11], modelling H as a random oracle. Security in this model assures us that an adversary cannot break the scheme with only black-box access to the group operations and H. Intuitively, this means that if there are any vulnerabilities in our construction, then these must exploit specific mathematical properties of elliptic curve groups or cryptographic hash functions used when instantiating the scheme.

Theorem 5.4.1. *For any adversary $\mathcal{A}$, let q be a bound on the total number of group elements it receives from queries it makes to the group oracles and from its interaction with the security game, described in §5.3.2. The above described construction is secure according to Definition 7 in the generic bilinear group and random oracle models. The advantage of $\mathcal{A}$ is $\mathcal{O}(q^2/p)$.*

In our proof, we are going to use the following strategy. First, we identify events that occur only with negligible probability, namely that the attacker is able to guess certain values successfully and that the oracle returns the same value for different queries. Assuming that these do not happen, we examine the (exponent) values which the attacker can obtain during the game. We show that $\mathcal{A}$ can recognise the challenge ciphertext only if is has used $GID_K \notin RL$ with a satisfying attribute set or has broken the rules of the game.

Proof. We describe the generic bilinear model as in [BBG05]. We let ψ_0 and ψ_1 be two random encodings of the additive group $\mathbb{Z}_p$. More specifically, each of ψ_0, ψ_1 is an injective map from $\mathbb{Z}_p$ to $\{0,1\}^m$, for $m > 3\log(p)$. We define the groups $\mathbb{G}_0 = \{\psi_0(x) : x \in \mathbb{Z}_p\}$ and $\mathbb{G}_1 = \{\psi_1(x) : x \in \mathbb{Z}_p\}$. We assume to have access to oracles which compute the induced group operations in $\mathbb{G}_0$ and $\mathbb{G}_1$ and an oracle which computes a non-degenerate bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$. We refer to $\mathbb{G}_0$ as a generic bilinear group. To simplify our notations let g denote $\psi_0(1)$, g^x denote $\psi_0(x)$, $e(g,g)$ denote $\psi_1(1)$, and $e(g,g)^y$ denote $\psi_1(y)$.

The challenger and the attacker play the security game (see in §5.3.2) and compute each value with respect to the generic bilinear group and random oracle models (i.e. send queries to the group oracle that responds with randomly assigned values). When $\mathcal{A}$ requests e.g. $H(GID_k)$ for some GID_k for the first time, the challenger chooses a random value $h_{GID_k} \in \mathbb{Z}_p$, queries the group oracle for $g^{h_{GID_k}}$, and gives this value to the attacker as $H(GID_k)$. It stores this value so that it can reply consistently to any subsequent requests for $H(GID_k)$.

We are going to show that in order to determine $\beta \in \{0,1\}$, $\mathcal{A}$ has to be able to compute $e(g,g)^{s^* h_{GID_k}}$ for any $k = 1, \ldots, r$, which is possible only with negligible probability without breaking the rules of the game.

We can assume that each of the attacker's queries to the group oracles either have input values that were given to $\mathcal{A}$ during the security game or were received from the oracles in response to previous queries. This is because of the fact that both ψ_0 and ψ_1 are random injective maps from $\mathbb{Z}_p$ into a set of at least p^3 elements, so the probability of the attacker being able to guess an element in the image of ψ_0, ψ_1 which it has not previously obtained is negligible.

Under this condition, we can think of each of the attacker's queries as a multi-variate expressions[4] in the variables $y_i, \alpha_i, \lambda_x, r_x, \omega_x, h_{GID_k}, a, b, s_k$, where i ranges over the attributes controlled by uncorrupted authorities, x ranges over the rows of the challenge access matrix, k ranges over the revoked identities. (We can also think of λ, ω_x as linear combinations of the variables $s, v_2, \ldots, v_\ell$ and $s^*, w_2, \ldots, w_\ell$.)

Furthermore we also assume that for each pair of different queries (corresponding to different polynomials), $\mathcal{A}$ receives different answers from the oracle. Since the maximum degree of polynomials is 8 (see the possible polynomials later), using the Schwartz-Zippel lemma [Sch80] we get that the probability of a collusion is $\mathcal{O}(1/p)$ and a union bound shows that the probability of that any such collusion happens during the game is $\mathcal{O}(q^2/p)$, which is negligible. Now suppose that it does not happen.

In order to determine β, the attacker clearly needs to recover s. [LW11] showed that without a satisfying set of attributes an attacker cannot make a query of the form $c(s+0\cdot h_{GID_k})$ (where c is a constant) thus has only negligible advantage in distinguishing an encoded message from a random group element (when using their original scheme). This result implies that in our modified construction, the attacker cannot make a query of the form $c(s + s^* h_{GID_k})$ without a satisfying set of attributes (as the first element of the blinding vector w is changed to s^* from 0) which also shows - following their

[4] These expressions can appear in the exponent of $e(g,g)$.

reasoning - that an expression in the form cs cannot be formed either. In our case, however, the possession of the necessary attributes are not enough to make a cs query, but $-c(s^* h_{GID_k})$ is also indispensable for this.

It can be seen that the case when $GID_k \in UL \setminus RL$ is equivalent to the original scheme of [LW11]. Consequently, from now on we can assume that all $GID_k \in RL$ and the challenge access policy is satisfied, thus simulating that all revoked users are colluding and prior to their revocation they were all able to decrypt. We will show that $\mathcal{A}$ cannot make a query of the form $-c(s^* h_{GID_k})$ and so not cs.

Based on the above assumptions the attacker can form queries which are linear combinations of

$$1, h_{GID_k}, y_i, \alpha_i + h_{GID_k} y_i, \lambda_x + \alpha_{\rho(x)} r_x, r_x, y_{\rho(x)} r_x + \omega_x,$$
$$a, 1/b, b h_{GID_k}(GID_k^* + a), s_k(a + GID_k^*), s_k/b,$$

the product of any two of these and α_i. (Note that GID_k^* for all $k = 1, \ldots, r$ and α_i, y_i for attributes i controlled by corrupted authorities are constants, known by the attacker.) In these queries shares of s^* can appear in two different forms: as ω_x and s_k, so we investigate whether $\mathcal{A}$ can achieve the desired value from these or not.

1. In order to gain $s^* h_{GID_k}$ by utilizing ω_x, $\mathcal{A}$ must use the product $h_{GID_k} y_{\rho(x)} r_x + h_{GID_k} \omega_x$ for all rows of A, as these are the only terms which contain $h_{GID_k} \omega_x$ and thus which can lead to $s^* h_{GID_k}$. To cancel out $h_{GID_k} y_{\rho(x)} r_x$ the attacker should form this product, which is possible only if $y_{\rho(x)}$ or r_x are known constants, because these elements appear alone in the above list and besides those, $\mathcal{A}$ can only form the product of any two but not three. However if $y_{\rho(x)}$ or r_x are constants for all x, that contradicts with the rules of the security game, because in that case corrupted attributes alone would satisfy the access structure.

2. When trying to obtain $s^* h_{GID_k}$ using s_k, we can observe that in each possible query term, s_k appears as multiplier either in all monads or in none of them. Evidently, terms without s_k are useless (see Table 5.2 for the relevant terms) for the attackers purposes and terms containing the $s_k h_{GID_l}$ monad can be useful. As it can be seen in Table 5.2, there are two types of terms which contain the necessary monad:

$$s_k a h_{GID_l} + GID_k^* s_k h_{GID_l}$$

and

$$s_k a h_{GID_l} + GID_l^* s_k h_{GID_l}.$$

Multiplying their subtraction by $c/(GID_k^* - GID_l^*)$ it is possible to gain $c \cdot s_k h_{GID_l}$, if $k \neq l$. In case of $k = l$ the two terms are equal, and $s_k a h_{GID_l}$ cannot be cancelled out, as no other terms contain this product. Nevertheless, according to our assumption that $GID_l^* \in RL$ for all $l = 1, \ldots, r$ there must be a $k = l$ as k runs over $1, \ldots, r$. We conclude that it is possible to gain $s_k h_{GID_l}$ for all k for any fixed l, if the attacker has used some $GID_l \notin RL$, which is again contradiction.

s_k/b
$s_k s_l/b^2$
$s_k a/b$
s_k/b^2
$s_k a h_{GID_l} + GID_l^* s_k h_{GID_l}$
$s_k h_{GID_l}/b$
$s_k s_l a/b + GID_k^* s_k s_l/b$
$s_k a + GID_k^* s_k$
$s_k s_l a^2 + GID_k^* GID_l^* s_k s_l + (GID_k^* + GID_l^*) s_k s_l a$
$s_k a^2 + GID_k^* s_k a$
$s_k a/b + GID_k^* s_k/b$
$s_k b h_{GID_l}(a^2 + (GID_k^* + GID_l^*)a + GID_k^* GID_l^*)$
$s_k a h_{GID_l} + GID_k^* s_k h_{GID_l}$

Table 5.2: Possible relevant query terms.

Hence, we have shown that under conditions that hold with all but $\mathcal{O}(q^2/p)$ probability, $\mathcal{A}$ cannot query $c(s^* h_{GID_k})$ (neither using ω_x nor s_k) therefore cannot get s without breaking the rules of the security game. It follows than, that the advantage of $\mathcal{A}$ is at most $\mathcal{O}(q^2/p)$. □

5.5 Conclusion

This chapter proposed the first scheme for efficient identity-based user revocation in multi-authority CP-ABE with several advantageous features compared with attribute-based revocation. Our results fulfil the specific needs of cloud environments, including scenario $\left[\begin{smallmatrix}\mathsf{U0}\\ \mathsf{T1101}\end{smallmatrix}\right]$ in the context of data markets, which we identified in §3.3.3. Our work leaves open the following questions.

Secure delegation of the revocation related computations of encryption to the cloud service provider could have multiple benefits. It would further distribute the computational overhead of revocation. Re-encryption would also become cheaper, as the re-computation of only a part of the ciphertexts would be enough for the update. Furthermore, re-encryption would become possible, without the encryptor's help, who could even be offline during the entire process. Steps in this direction would be both challenging and useful, without assuming trusted CSP.

The security of our construction is proved in the generic bilinear group model. However, we believe it would be possible to achieve full security by adapting the dual system encryption methodology, which was also used by Lewko and Waters [LW11] in their composite order group construction. This type of work would be interesting even if it resulted in a moderate loss of efficiency from our existing system.

In recent works [CZY20, ACGU20], functional encryption with fine-grained access control to the function output was investigated. It would be interesting to see whether our identity-based revocation method can be adapted to this more general setting to achieve an even more flexible cryptosystem.

Chapter 6

Searchable Symmetric-Key Encryption for Restricted Search

6.1 Introduction

In §5, we proposed a cryptosystem that enables data owners to use a data marketplace even if she of he does not trust the data broker (recall the scenario $\left[{}^{\mathsf{U0}}_{\mathsf{T1101}}\right]$ of Fig. 3.2a and the proposed solution in Fig. 5.1). In this part, we take one step further and investigate a variant of the previous scenario, in which the VASPs also do not trust the DB and would like to hide their access patterns to the data (this corresponds to scenario $\left[{}^{\mathsf{U0}}_{\mathsf{T1111}}\right]$ and the goal depicted on Fig. 3.2b).[1] Accordingly, we need a tool that can also hide the metadata, which is attached to the ciphertext, but in a way that still enables the VASPs to fin what they want to buy. To find solutions, we turn our attention towards searchable encryption.

Computation on hidden data and especially SSE has become an extensively studied area of cryptography in the last more than one and a half decade, since the appearance of the seminal paper of [SWP00]. The concept of SSE allows the secure storage of sensitive data on untrusted servers in the cloud without losing all the flexibility that plaintext data would allow. More precisely it supports keyword search over the ciphertexts in the following way: encrypted queries called trapdoors can be sent to the server which can test whether any of the stored ciphertexts matches the keyword underlying the trapdoor.

The two natural approaches towards realizing SSE are called “forward” and “inverted index”. The first one is to attach (or even include) one-way mappings of searchable keywords to the encrypted data. This leads to linear search complexity in the number of documents as the server has to go through all of them with a sequential scan to find all the matches for a trapdoor. A more sophisticated arrangement of the ciphertexts is to build an “inverted index”. In this case, the documents (or their IDs) are sorted based on the one-way mappings of keywords which are related to them. The latter solution allows logarithmic search complexity in the number of keywords. This clear

[1] Note that this goal does not hider the DB in its task to sell data, e.g. if pricing is based on the number of data accesses and not on what was accessed.

benefit caused that the inverted index approach became prevalent in SSE design which made significant progress in the past years [PCY+17]. At the same time, these solutions are rather complex and while operating smoothly on huge *static* databases, handling the *rapid expansion* of the database turns out to be more troublesome as the underlying data structure has to be updated without information leakage (see Table 6.1 for details).

In this part, we are looking for the answer to the following question.

> *What is the best suitable method for realizing encrypted search when the encrypted database is rapidly growing but it is enough to search parts of the database?*

6.1.1 Contributions

In [C4, J1] we aimed to build SSE that is optimized for the above scenario and provably fulfils the strongest security requirements towards SSE schemes. In the sequel, we introduce these results. Our construction is built from an arbitrary **IND-CPA** secure symmetric-key cipher and an **EU-CMA** secure MAC function working over asymmetric (Type-3) bilinear groups in which the symmetric external Diffie–Hellman assumption (SXDH) holds. The modular approach allows for flexible choices of the underlying primitives helping the adoption of our scheme to concrete applications.

In order to address the above-described challenges, we return to the forward index method to design an SSE scheme that handles newly encrypted data without requiring any updates on the already stored database. The additional benefit of this approach is that it allows periodical signing of the encrypted database (that can be particularly useful in case of log files) ensuring that the database was not modified, while the same is impossible in dynamically changing inverted indices. The core of our constriction is a keyword encryption and trapdoor generation method, both with randomized outputs that allows equality-test to determine whether a given trapdoor-ciphertext pair corresponds to the same keyword or not. These special keyword encryptions can then be attached to the ordinary encryption of a document (or file) and stored together on an honest but curious server. The security of our scheme against adaptive chosen-keyword attacks (**IND-CKA2**) is proven in the standard model under the SXDH assumption.

6.1.2 Applications

To motivate our study, we mention some use-cases, where finding *all* the occurrences of a given keyword in the whole database is not the goal. Particularly, they highlight that finding some correspondence in a properly chosen part of the database can also be meaningful and important. The other common property of the examples is that new entries are frequently added to the database. Therefore rapid updates are crucial, hitting the Achilles heel of the widespread inverted index approach.

Searching log entries. A device, deployed in an untrusted environment stores its own event logs in an SSE encrypted form with the event type as a keyword and possibly with a public time-stamp. It is often plausible to assume that the different events occur relatively often, resulting in frequent updates, and a remote operator is more likely to search for a specific event in a given time frame than in the entire database. He can send a trapdoor together with a tract of time to the device that checks its encrypted database and replies with the positive test results.

Vehicular data privacy. The onboard unit of a vehicle regularly sends encrypted aggregate data to an honest but curious remote server. The data packets are tagged with a few predefined characteristic features of the given packet, e.g. the oil level was under the limit, speeding was detected, the seatbelt was not fastened etc. An authority, possibly maintained by the car manufacturer as part of their services, can issue trapdoors for the car service/insurance company/police who can send these to the server and check whether some hypothetical event has occurred in a time period of interest. This process speeds up the actions as it can happen even in the absence of the vehicle and preserves the privacy of the car owner as no data, unrelated to an event, has to be decrypted as the relevance of some hidden data can be tested using SSE.

Data markets. Assume that metadata consists of multiple records that describes the data in different granularity in the data market scenario $[\substack{\mathsf{U0}\\\mathsf{T1111}}]$. In this case, the most general information is possibly not sensitive, e.g. a time frame when the data was recorded[2], and the VASP would only want to hide more specific informations of the bought data. In that case, only a portion of the database has to be searched that is possible after the following extension of the mechanism, sketched in Fig. 5.1. DOs also encrypt fine-grained metadata of their data using SSE. The SSE keys of a DO are encrypted with ABE such as the symmetric key, used for data encryption with the only difference that the SSE keys are not new for every update but used for a longer time. Before accessing some data, the VASPs have to obtain[3] the SSE key of a certain user to be able to create a trapdoor, that allows finding the wished entry in the encrypted database without revealing to the DB, what exactly was accessed.

6.2 Related Work

Several aspects of searchable encryption have been studied in the past years. We mention only a few of them and refer to recent surveys [BHJP14, PCY$^+$17] for an extensive summary on SSE. [CGKO06] captured first formally the intuitive goal of minimizing information leakage during keyword search (see §6.3.2). Schemes were put forward that allow not only single but also conjunctive/disjunctive keyword search [CJJ$^+$14]. Ranked keyword search over encrypted data was proposed by [XWSW16, YLL$^+$14]. Dynamic SSE schemes were designed to handle large databases with possible updates (see details in §6.4.2). Concurrently to our work [KKL$^+$17] proposed a combination of the forward

[2] Note that in case of an up-to-date service, the "age" of the accessed data will be leaked anyway because the DB can deduce this from the date and time, when the DO uploaded the affected entries.

[3] Of course, this is only possible if the access policy of the DO, allows the VASP to search his or her data.

and inverted index approach. Keyword search in the public-key setting (PEKS) was introduced by [BCOP04] and later improved in several directions [BHJP14].

6.3 Forward-Index SSE

6.3.1 Definition of SSE

Using the terminology of [BHJP14] we are interested in the single writer/single reader setting (while from the viewpoint of a data market, the multiple writer/multiple reader setting is also interesting). The server is assumed to be semi-trusted (honest-but-curious) and the communication channel between the user and the server is supposed to be authenticated. Next, we define the algorithms and the security of an SSE scheme that fits into the scenarios, described in §6.1.2.

SSE.Setup$(\lambda) \rightarrow (\mathsf{P}, \mathsf{sk})$ Upon receiving a security parameter λ it outputs the system parameters P and a secret key sk.

SSE.Enc$(\mathsf{P}, \mathsf{sk}, m, w) \rightarrow (C)$ Using the secret key sk it computes ciphertext C that encrypts m under keyword w.

SSE.TrpdGen$(\mathsf{P}, \mathsf{sk}, \hat{w}) \rightarrow (T)$ Using the secret key sk it computes a trapdoor T that can be used to test whether some ciphertext C was encrypted under keyword $\hat{w}$ or not.

SSE.Dec$(\mathsf{P}, \mathsf{sk}, C) \rightarrow (m)$ It decrypts ciphertext C with secret-key sk and outputs the resulting plaintext m.

SSE.Test$(\mathsf{P}, T, C) \rightarrow \{0,1\}$ The equality testing algorithm outputs 1 if T and C encodes the same keyword, i.e. $w = \hat{w}$ and 0 otherwise.

6.3.2 Security Model for SSE

The commonly used security model for SSE was defined by [CGKO06] to capture the intuition that, in the course of using the scheme, the remotely stored files and search queries together do not leak more information about the underlying data than the search pattern and the search outcome. In our security definition, we follow [CGKO06] but we formulate it – to the best of our knowledge for the first time – in the context of a forward index.

While in the inverted index-based approach the index and the ciphertexts are handled separately, in our case of a forward index, it is natural to view the "index" as part of the ciphertext. We formulate the model in this way, defining indistinguishability under adaptive chosen keyword attack (**IND-CKA2**) through a game between a challenger and an adversary. In the game, the adversary has to recognize which one of two challenge datasets (consisting of messages and their keywords chosen by herself) was encrypted by the challenger. While we are interested in a setting where the database is not static but can be expanded dynamically, in the security model we still restrict the adversary to query encryptions once and query only trapdoors adaptively. The reason behind this is that (in both of our motivating scenarios) searchability in a subset of the database can be enforced by choosing time-dependent keywords (i.e. "error-01-01-2018" instead of "error"). With the natural assumption that searches only apply to time periods that are already over (and thus updates does not affect them) without loss of generality, we can restrict our attention to one specific – now static – subset of the database (i.e. data from a given time frame). Note that in a forward index even the knowledge of the order of ciphertexts can help the attacker, that is why our challenger provides her with a random permutation of ciphertexts prepared from the randomly chosen challenge message set. The adversary has access not only to the encryptions themselves but also to a trapdoor generation oracle that can be queried adaptively with pairs of keywords corresponding to the two challenge sets. The oracle answers consistently with a trapdoor for that keyword which belongs to the encrypted challenge data set. The only restriction is that the queried keywords cannot separate the two challenge sets, as we are interested in information leakage beyond the search result.

For the ease of exposition, we assume that there is a single keyword for each message, but this can be easily generalized. More formally, we use the subsequent definition of security following [CGKO06, §4.2.2].

Definition 8 (**IND-CKA2** security)**.** *Let SSE = (**Setup**, **Enc**, **TrpdGen**, **Dec**, **Test**) be a secret-key searchable encryption scheme, $\lambda \in \mathbb{N}$ a security parameter, and $\mathcal{A} = (\mathcal{A}_0, \dots, \mathcal{A}_{q+1})$ a non-uniform adversary. Consider the probabilistic experiment* $\mathbf{IND\text{-}CKA2}_{\mathbf{SSE},\mathcal{A}}(\lambda)$ *depicted on Fig. 6.1 with the restriction that the number of keyword matches between the challenge message sets and the corresponding trapdoor queries are equal, i.e.*

$$\#\{i | \hat{w}_j^0 = w_i^0 \textit{ for } j \in [k]\} = \#\{i | \hat{w}_j^1 = w_i^1 \textit{ for } j \in [k]\}$$

for all $k = 1, \dots, q$, where q is some polynomial of the security parameter λ. We say that an SSE scheme is secure in the sense of adaptive indistinguishability if for all polynomial-time adversaries $\mathcal{A} = (\mathcal{A}_0, \dots, \mathcal{A}_{q+1})$,

$$\Pr(\mathbf{IND\text{-}CKA2}_{\mathbf{SSE},\mathcal{A}}(\lambda) = 1) \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

6.3.3 Proposed Scheme

In this part, we first describe a randomized MAC function based on any deterministic one in the form described in §2.5.1 and then utilize the resulting MAC to build our new SSE algorithms. We analyse both the security and the performance of the proposed scheme in §6.4.1 and 6.4.2 respectively.

Randomized MAC for SSE

In our SSE construction, we are going to make crucial use of two properties of the used MAC function. First, it needs to work over a bilinear pairing group, i.e. depending on

```
$\textbf{IND-CKA2}_{\textbf{SSE},\mathcal{A}}(\lambda)$ Security Game
$\mathsf{sk} \leftarrow_{\$} \textbf{SSE.Setup}(1^{\lambda})$
$b \leftarrow_{\$} \{0,1\}$
$(\mathsf{state}_{\mathcal{A}_0}, D^0, D^1) \leftarrow \mathcal{A}_0(1^{\lambda})$
parse $D^b$ as $\{(m_1^b, w_1^b), \ldots, (m_n^b, w_n^b)\}$
for $1 \leq j \leq n$,
    $C_i^b \leftarrow_{\$} \textbf{SSE.Enc}(\mathsf{sk}, m_i^b, w_i^b)$,
$C^b := (C_{\pi_1}^b, \ldots, C_{\pi_n}^b)$ for a random permutation $\pi$,
for $1 \leq j \leq q$,
    $(\mathsf{state}_{\mathcal{A}_j}, \hat{w}_j^0, \hat{w}_j^1) \leftarrow \mathcal{A}_j(\mathsf{state}_{\mathcal{A}_{j-1}}, C^b, \{T_i^b\}_{i \in [j]})$
    $T_j^b \leftarrow_{\$} \textbf{SSE.TrpdGen}(\mathsf{sk}, \hat{w}_j^b)$
$b' \leftarrow \mathcal{A}_{q+1}(\mathsf{state}_{\mathcal{A}_q}, C^b, \{T_j\}_{i=1,\ldots,q})$
return $b = b'$
```

Figure 6.1: **IND-CKA2** security game for forward index SSE schemes.

its input, it should prepare the output tag in one of the groups $\mathbb{G}_1$, $\mathbb{G}_2$. It also needs to be randomized meaning that a verification algorithm needs to be explicitly defined, as preparing the MAC of the same message two times results in different tags. We propose a simple solution, denoted by MAC', fulfilling these requirements, based on any deterministic MAC described in §2.5.1.

MAC.KeyGen'$(\lambda, \mathcal{G}) \rightarrow \mathsf{sk}_{\textbf{MAC}}$ It is identical to the original **MAC.KeyGen**(λ).

MAC'$(g_i, \mathsf{sk}_{\textbf{MAC}}, m) \rightarrow \tau$ On input g_i, which is a generator element of group $\mathbb{G}_i$, the output is prepared in this group. In order to do so, the algorithm first samples a random $r \in \mathbb{Z}_p^*$ and sets $\alpha = g_i^r$. Then using α, it computes $\textbf{MAC}(\mathsf{sk}_{\text{MAC}}, m) = \alpha^{F(\mathsf{sk}_{\text{MAC}}, m)} := \tau'$. The output is the two-tuple $\tau = (\alpha, \tau')$.

MAC.Verify'$(\mathsf{sk}_{\textbf{MAC}}, m, \tau) \rightarrow \{0,1\}$ It takes α from τ and checks if $\alpha^{F(\mathsf{sk}_{\text{MAC}}, m)} = \tau'$. If so it outputs 1, otherwise 0.

The **EU-CMA** security of the this randomized MAC follows from the security of the underlying deterministic MAC.

SSE Construction

The intuition behind the construction of our **Test** algorithm is fairly simple. We build the trapdoors for a specific keyword and the keyword related ciphertext components in a symmetric manner: both are randomised MACs of the underlying keyword, however, represented in distinct groups $\mathbb{G}_1$ or $\mathbb{G}_2$. This enables us to test equality by "mixing" the ciphertext and the trapdoor in two different ways (using the pairing operation) that are equal only if the underlying keywords are the same. Using distinct groups prevents the testability both among ciphertexts and among trapdoors. In more detail, the algorithms are the following.

SSE.Setup$(\lambda) \rightarrow (\mathsf{P}, \mathsf{sk})$ It proceeds with the following steps:

- samples an instance of Type-3 pairing groups $\mathcal{G} = \{p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e\}$,
- chooses an instance of the above defined MAC' that implicitly defines function $F : \mathbb{Z}_p^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*$,
- runs the **MAC.KeyGen**' algorithm of the chosen MAC function to generate $\mathsf{sk}_{\mathrm{MAC}}$,
- samples secret keys $\mathsf{sk}_{\mathrm{SE}}$ for the used SE scheme,
- samples secret key $\mathsf{sk}_{\mathrm{SSE}} \leftarrow\!\!{\scriptstyle\$}\ \mathbb{Z}_p^*$,
- outputs the public parameters $\mathsf{P} = (\mathcal{G}, F, g^{\mathsf{sk}_{\mathrm{SSE}}})$ and the secret key $\mathsf{sk} = \{\mathsf{sk}_{\mathrm{MAC}}, \mathsf{sk}_{\mathrm{SE}}, \mathsf{sk}_{\mathrm{SSE}}\}$.

SSE.Enc$(\mathsf{P}, \mathsf{sk}, m, w) \rightarrow (C)$ The encryption of data m is executed using the encryption algorithm of a symmetric-key cipher while for the encryption of the keyword we utilize first MAC':

- it runs **MAC**'$(g_1, \mathsf{sk}_{\mathrm{MAC}}, w) = (\alpha = g_1^r, \tau' = g_1^{rF(\mathsf{sk}_{\mathrm{MAC}}, w)})$ where $r \in \mathbb{Z}_p^*$ is sampled randomly,
- finally it computes the following ciphertext:

$$C = \left(c_1 = \alpha, c_2 = \tau'^{\mathsf{sk}_{\mathrm{SSE}}}, c_3 = \mathbf{SE.Enc}(\mathsf{sk}_{\mathrm{SE}}, m)\right).$$

SSE.TrpdGen$(\mathsf{P}, \mathsf{sk}, \hat{w}) \rightarrow (T)$ Upon receiving a keyword $\hat{w}$

- it runs **MAC**'$(g_2, \mathsf{sk}_{\mathrm{MAC}}, \hat{w}) = (\alpha = g_2^{r'}, \tau' = g_2^{r'F(\mathsf{sk}_{\mathrm{MAC}}, \hat{w})})$ where $r' \in \mathbb{Z}_p^*$ is sampled randomly,
- and computes the SSE trapdoor in the following form:

$$T = \left(t_1 = \alpha, t_2 = \tau'^{\mathsf{sk}_{\mathrm{SSE}}}\right).$$

SSE.Dec$(\mathsf{sk}, C) \rightarrow (m)$ After parsing C as (c_1, c_2, c_3), in order to recover the encrypted data the decryption algorithm of the symmetric-key scheme is used and $\mathbf{SE.Dec}(\mathsf{sk}_{\mathrm{SE}}, c_3) = m$ is returned, while the rest of the ciphertext is not affected.

SSE.Test$(\mathsf{P}, T, C) \rightarrow \{0, 1\}$ To test whether a ciphertext C (parsed as (c_1, c_2, c_3)) was encoded using the same keyword that is hidden in trapdoor T (parsed as (t_1, t_2)) the following equality is checked:

$$e(c_2, t_1) = e(c_1, t_2).$$

If the equality holds the output is 1, otherwise 0.

The correctness of the **SSE.Dec** and **SSE.Test** algorithms follows after substitution of the proper values into the formulas, i.e. in case of the latter one $e(c_2, t_1) = e(g_1, g_2)^{rr'\mathsf{sk}_{\mathrm{SSE}}F(\mathsf{sk}_{\mathrm{MAC}}, w)}$ that is equal to $e(g_1, g_2)^{rr'\mathsf{sk}_{\mathrm{SSE}}F(\mathsf{sk}_{\mathrm{MAC}}, \hat{w})} = e(c_1, t_2)$ iff $w = \hat{w}$.

6.4 Evaluation

6.4.1 Security Analysis

In this part, we formulate the main theorem of this chapter and its proof.

Theorem 6.4.1. *If the used symmetric-key encryption scheme is* **IND-CPA** *secure, the underlying MAC function is* **EU-CMA** *secure and the SXDH assumption holds in the pairing group $\mathcal{G}$, then the proposed SSE scheme is* **IND-CKA2** *secure according to Definition 8.*

Before proving the theorem, we prove three lemmas that are going to be used in the proof of the theorem.

Lemma 6.4.2. *The SSE ciphertext $C = (c_1, c_2, c_3)$ alone (when no trapdoor is issued) is semantically (***IND-CPA***) secure, if the underlying SE scheme is* **IND-CPA** *secure and the SXDH assumption holds.*

Proof sketch. By our assumption, the ciphertext of the symmetric-key cipher is semantically secure and thus in order to prove the lemma we have to show that the other two components c_1, c_2 of the SSE ciphertext is also indistinguishable from truly random values (when trapdoors do not help to distinguish them). Note that the structure of these values is reminiscent of ElGamal ciphertexts. The difference is that the "message" $F(\mathsf{sk}_{\mathrm{MAC}}, w)$ is not multiplied by the randomizing factor $g_i^{r\mathsf{sk}_{\mathrm{SSE}}}$ but exponentiated to it, however, the same reduction of the DDH assumption in $\mathbb{G}_1$ to the security of the scheme works here just like in case of the ElGamal cryptosystem. In the reduction the simulator answers for queries for keywords w_j (for some j, smaller than the allowed number queries) with $c_1 = g_1^{r_j}, c_2 = g_1^{r_j b_1 F(\mathsf{sk}_{\mathrm{MAC}}, w_j)}$, while in the challenge phase sends $c_1 = g_1^{a_1}, c_2 = g_1^{R \cdot F(\mathsf{sk}_{\mathrm{MAC}}, w_b)}$ for randomly chosen $b \in \{0, 1\}$. Finally the output of the simulator is 1 if the guess b' of the adversary is $b = b'$ and 0 otherwise. □

Lemma 6.4.3. *As long as the SXDH assumption holds and testing equality with ciphertexts does not help distinguishing the SSE trapdoor $T = (t_1, t_2)$ from a truly random tuple, T hides the underlying keyword with* **IND-CPA** *security.*

Proof sketch. Note that, just like c_1, c_2 from C, the structure of T also resembles the ElGamal ciphertext and thus the proof of the lemma again follows the blueprint of reducing the security of the ElGamal cryptosystem to the DDH assumption (i.e. in our case, to the SXDH assumption in group $\mathbb{G}_2$). □

Lemma 6.4.4. *The SSE trapdoor $T = (t_1, t_2)$ is existentially-unforgeable (***EU-CMA*** secure).*

Proof. We are going to show that if there exists an efficient algorithm $\mathcal{A}$ that can forge trapdoor $T = (t_1, t_2)$, then there also exist $\mathcal{B}$ that using $\mathcal{A}$ can forge the underlying MAC' function, contradicting with its **EU-CMA** security.

According to Definition 3, $\mathcal{B}$ has access to a CMA oracle which replies with $\tau = (\alpha, \tau')$ for a message request w. This allows $\mathcal{B}$ to answer the trapdoor requests of $\mathcal{A}$ for any keyword w with a valid trapdoor $T = (t_1 = \alpha = g_2^{r'}, t_2 = \tau'^{\mathsf{sk}_{\mathrm{SSE}}} = g_2^{r'\mathsf{sk}_{\mathrm{SSE}} F(\mathsf{sk}_{\mathrm{MAC}}, w)})$, after generating an SSE secret key $\mathsf{sk}_{\mathrm{SSE}} \in \mathbb{Z}_p^*$. When $\mathcal{A}$ outputs the forged trapdoor $\bar{T} = \left(\bar{t_1} = g_2^{\bar{r}}, \bar{t_2} = g_2^{\bar{r}\mathsf{sk}_{\mathrm{SSE}} F(\mathsf{sk}_{\mathrm{MAC}}, \bar{w})}\right)$ for keyword $\bar{w}$, that was not requested previously, $\mathcal{B}$ can compute $\bar{\tau} = (\bar{\alpha} = t_1, \bar{\tau'} = \bar{t_2})^{1/\mathsf{sk}_{\mathrm{SSE}}}$ and output the pair $(\bar{\tau'}, \bar{w})$ as the forged message-tag pair, breaking the security of the underlying MAC'. □

Now we are ready to prove our main theorem.

Proof of Theorem 6.4.1. We are going to define several hybrid games where in the last hybrid the attacker receives encryptions of random values under random keywords instead of the challenge message and trapdoors for random keywords instead of the queried ones, thus she cannot have a non-negligible advantage in that game. In order to prove the theorem, we systematically show that the subsequent hybrids are indistinguishable for the adversary and thus her advantage in the original game is essentially negligible as well.

For the ease of exposition we introduce some notations. Let $\vec{W} = (W_1, \ldots, W_m)$ be the vector of all queried keyword values either for a trapdoor or as part of the encryption query (thus $m \leq n + q$). We denote the set of message (keyword) query indices that match with W_j by $I_{W_j} = \{i | w_i^b = W_j\}$ (and by $\hat{I}_{W_j} = \{i | \hat{w}_i^b = W_j\}$ respectively). Let R be an initially empty list of two-tuples (W_i, R_i) in which the challenger can store random R_i values for each W_i. Finally we denote the components of C_i^b by $c_{1,i}, c_{2,i}, c_{3,i}$ and the components of T_i^b by $t_{1,i}, t_{2,i}$. We define the following hybrid games.

Hyb_0: it corresponds to the original game.

Hyb_1: the same as Hyb_0 but in C^b, the third ciphertext components $c_{3,i}$ are substituted with random values of the ciphertext space for all $i \in [n]$.

$\mathsf{Hyb}_{2,1}$: the same as Hyb_1, with the following exception. A random $R_1 \leftarrow_\$ \mathbb{Z}_p^*$ is sampled and (W_1, R_1) is appended to R. For each $i \in I_{W_1}$ and the challenger answers with C_i^b in which the second parameter is $c_{2,i} = g_1^{r_i R_1}$ instead of $c_{2,i} = g_1^{r_i \mathsf{sk}_{\mathrm{SSE}} F(\mathsf{sk}_{\mathrm{MAC}}, w_i^b)}$. Similarly for each $i \in \hat{I}_{W_1}$ the challenger answers to trapdoor queries for $\hat{w}_i^b$ with T_i^b in which $t_{2,i} = g_2^{r'_i R_1}$ instead of $t_{2,i} = g_2^{r'_i \mathsf{sk}_{\mathrm{SSE}} F(\mathsf{sk}_{\mathrm{MAC}}, \hat{w}_i^b)}$.

$\vdots$

$\mathsf{Hyb}_{2,m}$: the same as $\mathsf{Hyb}_{2,m-1}$, with the following exception. A random $R_m \leftarrow_\$ \mathbb{Z}_p^*$ is sampled and (W_m, R_m) is appended to R. For each $i \in I_{W_m}$ the challenger answers with C_i^b in which the second parameter is $c_{2,i} = g_1^{r_i R_m}$ instead of $c_{2,i} = g_1^{r_i \mathsf{sk}_{\mathrm{SSE}} F(\mathsf{sk}_{\mathrm{MAC}}, w_i^b)}$. Similarly for each $i \in \hat{I}_{W_m}$ the challenger answers to trapdoor queries for $\hat{w}_i^b$ with T_i^b in which $t_{2,i} = g_2^{r'_i R_m}$ instead of $t_{2,i} = g_2^{r'_i \mathsf{sk}_{\mathrm{SSE}} F(\mathsf{sk}_{\mathrm{MAC}}, \hat{w}_m^b)}$.

Now we are going to show that the subsequent hybrids are indistinguishable for the attacker.

Claim 6.4.5. *The computational indistinguishability of* Hyb_0 *and* Hyb_1 *follows from Lemmas 6.4.2 and 6.4.3.*

Lemma 6.4.2 states that $c_{3,i}$ is semantically secure or in other words that the adversary can have at most negligible advantage in distinguishing it from a random value. At the same time, keywords are possibly not independent of the data and thus the other ciphertext components $c_{1,i}, c_{2,i}$ can help $\mathcal{A}$to recognize the shift between the hybrids. However, even if $w_i = f(m_i)$ for some function f, the testing algorithm cannot help $\mathcal{A}$ because the results of it are identical in the two hybrids. The only remaining possibility is that $c_{1,i}, c_{2,i}$ leaks information about w_i (and thus indirectly about m_i) but this would contradict with Lemma 6.4.3.

Consequently, $\mathcal{A}$ can have at most $2n+q$ times negligible advantage in distinguishing Hyb_0 and Hyb_1, that is still negligible.

Claim 6.4.6. Hyb_1 *and* $\mathsf{Hyb}_{2,1}$ *are computationally indistinguishable because of Lemmas 6.4.2, 6.4.3 and 6.4.4.*

First of all, we notice that the **SSE.Test** algorithm does not help the adversary in distinguishing these hybrids as it clearly has the same outputs in both cases. Also note that semantic security of the ciphertexts and the trapdoors (guaranteed by Lemmas 6.4.2, 6.4.3) are necessary but not sufficient to imply indistinguishability. The reason is that in Hyb_1, $\mathcal{A}$ might be able to use the trapdoor queries to gather valid keyword-trapdoor pairs adaptively in order to generate a valid trapdoor T_{q+1} for keyword $\hat{w}_{q+1}$ that was not queried in the game, thus it might not fulfil the constraints of the game[4]. However, according to Lemma 6.4.4, the trapdoors are **EU-CMA** secure meaning that $\mathcal{A}$ can have only negligible advantage even in Hyb_1 to generate a non-queried keyword-trapdoor pair just like in $\mathsf{Hyb}_{2,1}$ in which the trapdoor is independent of the actual keyword. This shows that the attacker can have at most $(|I_{W_1} + \hat{I}_{W_1}| + 1)$ times negligible advantage in distinguishing Hyb_1 and $\mathsf{Hyb}_{2,1}$.

Note that the indistinguishability of the remaining hybrids follows from an analogous argument (as we repeat the same steps in those cases) showing that the following claim holds as well.

Claim 6.4.7. *The computational indistinguishability of* $\mathsf{Hyb}_{2,i-1}$ *and* $\mathsf{Hyb}_{2,i}$ *for any* $i \in [2, m]$ *follows from Lemmas 6.4.2, 6.4.3 and 6.4.4.*

By showing the indistinguishability of the hybrids we have proven the theorem. □

Forward privacy guarantees that trapdoors can only be used to test keywords of documents which were already part of the database at the time of issuing the trapdoor. We remark that by default, our scheme is not forward secure (trapdoors can be stored

[4] We note that in case of Claim 6.4.5 the same was not a problem, as the extra knowledge of another valid message-ciphertext pair would not help the attacker in the game.

by the server and can be used to test future ciphertexts) but the already mentioned usage of time-dependent keywords can remedy this deficiency, thus achieving forward (and also backward) privacy between the time periods. Note that the same solution of time-specific keywords would result in an infinitely growing inverted index.

6.4.2 Performance Evaluation

Scheme	**Model**	**Security**	**Fw/Bw Privacy**	**Update Complexity**	**Update Privacy**	**Search Complexity**
[SWP00]	Standard	**IND-CPA**	× / ×	$O(b)$	×	$O(n \cdot b)$
[vLSD+10]	Standard	**IND-CKA2**	× / ×	$O(w_D)^*$	×	$O(\log W)$
[KPR12]	ROM	**CKA2**	× / ×	$O(w_D)$	×	$O(n_w)$
[KP13]	ROM	**CKA2**	× / ×	$O(\log n)^*$	✓	$O(n_w \log n)$
[CJJ+14]	ROM	**CKA2**	× / ×	$O(w_D + W \log n)$	×	$O(n_w + a + d)$
[SPS14]	ROM	**CKA2**	✓ / ×	$O(w_D \log(nW))^*$	✓	$O(n_w + d)$
[HK14]	ROM	**CKA2**	× / ×	$O(nw_D/D)^{**}$	×	$O(nw_D/D)^{**}$
[YG15]	ROM	**CKA2**	✓ / ✓	$O(W)$	✓	$O(W)$
[Gaj16]	Standard	**IND-CKA2**	× / ×	$O(w_D \cdot W)$	×	$O(\log n)$
[KKL+17]	ROM	**CKA2**	✓ / ×	$O(w_D)^*$	✓	$O(n_w)^*$
Our scheme	Standard	**IND-CKA2**	✓ / ✓	$O(w_D)$	✓	$O(n \cdot w_D)$

Table 6.1: Comparison of our results and dynamic SSE Schemes. (For notations and evaluation, see §6.4.2.)

We compare our results with dynamic SSE schemes which are the most suitable in the literature for the use cases that we considered in this work. Table 6.1 shows a comparison using the following notations: n denotes the number of documents (data entries), w_D is the number of keywords per a specific document, W is the total number of distinct keywords in the database, n_w is the number of documents matching the searched keyword w, a is the total number of additions to the database and d is the total number of deletions, b is the bit length of encrypted documents. * indicates that update requires some rounds of interaction between the server and the client and ** denotes amortized complexity.

As we expected, Table 6.1 confirms that our search strategy of sequential scan is not competitive, unless only a small portion of the database (e.g. at most $(\log W)/w_D$ ciphertexts) is enough to scan, that is realistic in the investigated scenarios. The most important benefits of our scheme include resistance against adaptive chosen-keyword attacks in the standard model and non-interactive update of ciphertexts with low complexity, depending only on the number of keywords. Moreover, our ciphertexts (including the index) and trapdoors are very short, consisting of $w_D + 2$ and 2 group elements respectively.

Let us emphasize that updating the database with a new record is straightforward in our approach. The client encrypts the data together with the keywords and the server only has to store the received ciphertexts contrary to other solutions where the server has to “find the place” of the new entry in the index. This latter process also harms the privacy of updates in most cases by leaking information about the added keywords (e.g. all documents with common keywords can be identified). In our case, only the number of added keywords is leaked, however, in the targeted applications it is plausible to assume that the number of keywords is not varying among the different “documents”.

6.5 Conclusion

In this chapter, we investigated the problem of frequent updates to encrypted databases that allow for keyword search. We revisited the forward index approach of SSE and demonstrated that in those applications where only a part of the database is enough to be scanned, it can outperform schemes using the wide-spread inverted index approach. The **IND-CKA2** security of our modular construction can be proven in the plain model under standard assumptions. To the best of our knowledge, ours is the first scheme to achieve update privacy in the standard model.

While we considered the single writer/single reader setting, in the context of data markets all the other SSE setups could be interesting (especially the ones with multiple writers) and it is an interesting question how our approach could be adopted to these scenarios.

Another direction could be to investigate whether a similar result is achievable without pairings or not. The elimination of the pairing operation could result in significant efficiency improvement.

Chapter 7

Conclusion

Our need for data is apparent today. Utilizing the data we produce everyday is not yet solved. The concept of data markets is still emerging but the development is beyond question. In this dissertation, we investigated the security aspects of such virtual markets, extending a previous work [C3] on the same topic. After the structuring of the problem domain, we can see that adoption of existing cryptographic primitives is possible in many cases. However, if they needed to be integrated to achieve all of the goals, then this integration requires further efforts. On the other hand, some problems require the improvement of currently available primitives because they can only support very restricted functionalities (e.g. see our survey on such primitive [B1]). For the simultaneous function privacy and verifiability problem, we need different trade-offs to obtain meaningful results. In [C2] we presented one such solution, that can be applied to the simple functionality of inner-product computation. The next results are directly applicable in the secure cloud storage problem, in order to enhance flexibility of the encryption. Our works [C5, J2], on CP-ABE aims to make access control more flexible, even when a user has to be revoked from the system. A diferent aspect of flexibility was investigated in [C4, J1]. Namely, we proposed an efficient method to both enable rapid append operations to data and fast search, whenever it is enough to look through only smaller parts of a database.

List of Publications

Conference and Workshop Publications

[C1] Gergely Biczók, Máté Horváth, Szilveszter Szebeni, István Lám and Levente Buttyán. The cost of having been pwned: A security service provider's perspective. *The 3rd International Workshop on Emerging Technologies for Authorization and Authentication (Co-Located with ESORICS 2020) – ETAA 2020*.

[C2] Máté Horváth, Levente Buttyán, Gábor Székely and Dóra Neubrandt. There Is Always an Exception: Controlling Partial Information Leakage in Secure Computation. *J. H. Seo (Ed.): Information Security and Cryptology - ICISC 2019, LNCS 11975, pp. 133–149, 2020.*

[C3] Máté Horváth, Levente Buttyán. Problem Domain Analysis of IoT-Driven Secure Data Markets. In: E. Gelenbe et al. (eds) *Security in Computer and Information Sciences. Euro-CYBERSEC 2018. Communications in Computer and Information Science*, vol 821. pp. 57–67, Springer, 2018.

[C4] Máté Horváth, István Vajda. Searchable symmetric encryption: Sequential scan can be practical. *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp 1–5, 2017.

[C5] Máté Horváth. Attribute-Based Encryption Optimized for Cloud Computing. G.F. Italiano et al. (eds), *SOFSEM 2015: Theory and Practice of Computer Science – 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24–29, 2015. Proceedings*, *Lecture Notes in Computer Science* vol. 8939, pp. 566–577, Springer, 2015.

[C6] Máté Horváth. Private Key Delegation in Attribute-Based Encryption. *Mesterpróba – Conference for last year MSc and first year PhD students*, pp. 21–24, 2015.

Journal Publications

[J1] Máté Horváth, István Vajda. Searchable Symmetric Encryption for Restricted Search. *Journal of Communications Software and Systems*, 14(1):104–111, 2018.

[J2] Máté Horváth. Attribute-Based Encryption Optimized for Cloud Computing. *Infocommunications Journal*, 7(2):1–9, 2015.

Book

[B1] Máté Horváth, Levente Buttyán. Cryptographic Obfuscation: A Survey. *Springer Briefs in Computer Science, ISBN 978-3-319-98040-9.* Springer, 2020. Manuscript is available here: `http://eprint.iacr.org/2015/412`.

Bibliography

[ABC$^+$15] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A. Reuter, and Martin Strand. A guide to fully homomorphic encryption. Cryptology ePrint Archive, Report 2015/1192, 2015. http://eprint.iacr.org/2015/1192.

[ABCP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *Proceedings of Public-Key Cryptography - PKC 2015*, volume 9020 of *LNCS*, pages 733–751. Springer, 2015.

[ABP15] Michel Abdalla, Fabrice Benhamouda, and Alain Passelègue. An algebraic framework for pseudorandom functions and applications to related-key security. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 388–409, 2015.

[ACGU20] Michel Abdalla, Dario Catalano, Romain Gay, and Bogdan Ursu. Inner-product functional encryption with fine-grained access control. Cryptology ePrint Archive, Report 2020/577, 2020. https://eprint.iacr.org/2020/577.

[AGM$^+$13] Joseph A. Akinyele, Christina Garman, Ian Miers, Matthew W. Pagano, Michael Rushanan, Matthew Green, and Aviel D. Rubin. Charm: a framework for rapidly prototyping cryptosystems. *J. Cryptographic Engineering*, 3(2):111–128, 2013.

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 333–362. Springer, 2016.

[Bar16] Boaz Barak. Lecture notes from the Cryptography course at Harvard University, Spring 2016. https://intensecrypto.org/public/lec_17_SFE.html, Accessed 4. October 2020.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology–EUROCRYPT 2005*, pages 440–456. Springer, 2005.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 506–522, 2004.

[Bei96] Amos Beimel. *Secure schemes for secret sharing and key distribution.* PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[BGJS16] Saikrishna Badrinarayanan, Vipul Goyal, Aayush Jain, and Amit Sahai. Verifiable functional encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 557–587, 2016.

[BHJP14] Christoph Bösch, Pieter H. Hartel, Willem Jonker, and Andreas Peter. A survey of provably secure searchable encryption. *ACM Comput. Surv.*, 47(2):18:1–18:51, 2014.

[BKM$^+$18] Allison Bishop, Lucas Kowalczyk, Tal Malkin, Valerio Pastro, Mariana Raykova, and Kevin Shi. A simple obfuscation scheme for pattern-matching with wildcards. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 731–752. Springer, 2018.

[BMR10] Dan Boneh, Hart William Montgomery, and Ananth Raghunathan. Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 131–140, 2010.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73, 1993.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *Theory of Cryptography - TCC 2011. Proceedings*, volume 6597 of *LNCS*, pages 253–273. Springer, 2011.

[CDKS19] Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 395–412. ACM, 2019.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004.

[CGKO06] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 79–88. ACM, 2006.

[Cha07] Melissa Chase. Multi-authority Attribute Based Encryption. In *Theory of Cryptography*, volume 4392 of *LNCS*, pages 515–534. Springer Berlin Heidelberg, 2007.

[CHM10] Sanjit Chatterjee, Darrel Hankerson, and Alfred Menezes. On the efficiency and security of pairing-based protocols in the type 1 and type 4 settings. In M. Anwar Hasan and Tor Helleseth, editors, *Arithmetic of Finite Fields, Third International Workshop, WAIFI 2010, Istanbul, Turkey, June 27-30, 2010. Proceedings*, volume 6087 of *Lecture Notes in Computer Science*, pages 114–134. Springer, 2010.

[CIK$^+$01] Ran Canetti, Yuval Ishai, Ravi Kumar, Michael K. Reiter, Ronitt Rubinfeld, and Rebecca N. Wright. Selective private function evaluation with applications to private statistics. In Ajay D. Kshemkalyani and Nir Shavit, editors, *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, PODC 2001,*, pages 293–304. ACM, 2001.

[CJJ$^+$14] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[CL14] Zhengjun Cao and Lihua Liu. Analysis of Lewko-Sahai-Waters Revocation System. Cryptology ePrint Archive, Report 2014/937, 2014. http://eprint.iacr.org/.

[CM11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.

[CMM+20] Gabriella Cattaneo, Giorgio Micheletti, Chrysoula Mitta, Mike Glennon, and Carla La Croce. *The European data market monitoring tool, Key facts & figures, first policy conclusions, data landscape and quantified stories: d2.9 final study report.* Publications Office of the EU, July 2020.

[CT05] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005, Proceedings*, volume 3386 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2005.

[CXL16] Zhao Chang, Dong Xie, and Feifei Li. Oblivious RAM: A dissection and experimental evaluation. *Proc. VLDB Endow.*, 9(12):1113–1124, 2016.

[CZY20] Yuechen Chen, Linru Zhang, and Siu-Ming Yiu. Decentralized multi-client attribute based functional encryption. In Pierangela Samarati, Sabrina De Capitani di Vimercati, Mohammad S. Obaidat, and Jalel Ben-Othman, editors, *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020 - Volume 2: SECRYPT, Lieusaint, Paris, France, July 8-10, 2020*, pages 118–129. ScitePress, 2020.

[Dat] Data Market Austria Project. https://datamarket.at/en/, Accessed 4. October 2020.

[Dat17] Datum Network GmbH. Datum Network - The decentralized data marketplace (White Paper V15). Technical report, Datum Network GmbH., 2017. https://datum.org/, Accessed 4. October 2020.

[DC14] Changyu Dong and Liqun Chen. A fast secure dot product protocol with application to privacy preserving association rule mining. In Vincent S. Tseng, Tu Bao Ho, Zhi-Hua Zhou, Arbee L. P. Chen, and Hung-Yu Kao, editors, *Advances in Knowledge Discovery and Data Mining. Proceedings, Part I*, volume 8443 of *Lecture Notes in Computer Science*, pages 606–617. Springer, 2014.

[Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, pages 100–109, 2002.

[DP20] Damien Desfontaines and Balázs Pejó. Sok: Differential privacies. *Proceedings on Privacy Enhancing Technologies*, 2020(2):288 – 313, 01 Apr. 2020.

[DSZ15] Daniel Demmler, Thomas Schneider, and Michael Zohner. ABY - A framework for efficient mixed-protocol secure two-party computation. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015.* The Internet Society, 2015.

[EN16] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1388–1401. ACM, 2016.

[FJT13] Pierre-Alain Fouque, Antoine Joux, and Mehdi Tibouchi. Injective encodings to elliptic curves. In Colin Boyd and Leonie Simpson, editors, *Information Security and Privacy - 18th Australasian Conference, ACISP 2013. Proceedings*, volume 7959 of *LNCS*, pages 203–218. Springer, 2013.

[Gaj16] Sebastian Gajek. Dynamic symmetric searchable encryption from constrained functional encryption. In *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, pages 75–89, 2016.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 40–49. IEEE Computer Society, 2013.

[GGHZ16] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 480–511. Springer, 2016.

[GK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 505–522, 2016.

[GLL$^+$19] Ran Gilad-Bachrach, Kim Laine, Kristin E. Lauter, Peter Rindal, and Mike Rosulek. Secure data exchange: A marketplace in the cloud. In Radu Sion and Charalampos Papamanthou, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop, CCSW@CCS 2019, London, UK, November 11, 2019*, pages 117–128. ACM, 2019.

[GLLM04] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. On private scalar product computation for privacy-preserving data mining. In Choonsik Park and Seongtaek Chee, editors, *Information Security and Cryptology - ICISC 2004, 7th International Conference, Seoul, Korea, December 2-3, 2004, Revised Selected Papers*, volume 3506 of *Lecture Notes in Computer Science*, pages 104–120. Springer, 2004.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2012.

[HK14] Florian Hahn and Florian Kerschbaum. Searchable encryption with secure and efficient updates. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 310–320, 2014.

[IOT] IOTA Data Marketplace. https://data.iota.org/, Accessed: 4. October 2020.

[KKL$^+$17] Kee Sung Kim, Minkyu Kim, Dongsoo Lee, Je Hong Park, and Woo-Hwan Kim. Forward secure dynamic searchable symmetric encryption with efficient updates. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1449–1463. ACM, 2017.

[KKRT16] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious PRF with applications to private set intersection. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 818–829. ACM, 2016.

[KKW17] W. Sean Kennedy, Vladimir Kolesnikov, and Gordon T. Wilfong. Overlaying conditional circuit clauses for secure computation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 499–528. Springer, 2017.

[KM07] Neal Koblitz and Alfred Menezes. Another look at generic groups. *Advances in Mathematics of Communications*, 1(1):13–28, 2007.

[KM15] Neal Koblitz and Alfred J. Menezes. The random oracle model: a twenty-year retrospective. *Des. Codes Cryptography*, 77(2-3):587–610, 2015.

[KP13] Seny Kamara and Charalampos Papamanthou. Parallel and dynamic searchable symmetric encryption. In *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, pages 258–274, 2013.

[KPR12] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic searchable symmetric encryption. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 965–976, 2012.

[KS08] Vladimir Kolesnikov and Thomas Schneider. A practical universal circuit construction and secure evaluation of private functions. In Gene Tsudik, editor, *Financial Cryptography and Data Security, 12th International Conference, FC 2008, Cozumel, Mexico, January 28-31, 2008, Revised Selected Papers*, volume 5143 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2008.

[KS16] Ágnes Kiss and Thomas Schneider. Valiant's universal circuit is practical. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 699–728. Springer, 2016.

[LC10] Zhen Liu and Zhenfu Cao. On Efficiently Transferring the Linear Secret-Sharing Scheme Matrix in Ciphertext-Policy Attribute-Based Encryption. *IACR Cryptology ePrint Archive*, 2010:374, 2010.

[LLLS10] Xiaohui Liang, Rongxing Lu, Xiaodong Lin, and Xuemin Sherman Shen. Ciphertext policy attribute based encryption with efficient revocation. *TechnicalReport, University of Waterloo*, 2010.

[LLMS14] Chao Li, Daniel Yang Li, Gerome Miklau, and Dan Suciu. A theory of pricing private data. *ACM Trans. Database Syst.*, 39(4):34:1–34:28, 2014.

[LP09] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):5, 2009.

[LSW10] Allison Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *IEEE Symposium on Security and Privacy*, pages 273–285, 2010.

[LW09] Allison B. Lewko and Brent Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, pages 112–120, 2009.

[LW11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Advances in Cryptology–EUROCRYPT 2011*, pages 568–588. Springer, 2011.

[LXZ13] Qinyi Li, Hu Xiong, and Fengli Zhang. Broadcast revocation scheme in composite-order bilinear group and its application to attribute-based encryption. *International Journal of Security and Networks*, 8(1):1–12, 2013.

[LZW$^+$13] Yang Li, Jianming Zhu, Xiuli Wang, Yanmei Chai, and Shuai Shao. Optimized Ciphertext-Policy Attribute-Based Encryption with Efficient Revocation. *International Journal of Security & Its Applications*, 7(6), 2013.

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, pages 1–12, 2005.

[MBC14] Travis Mayberry, Erik-Oliver Blass, and Agnes Hui Chan. Efficient private file retrieval by combining ORAM and PIR. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society, 2014.

[NAP$^+$14] Muhammad Naveed, Shashank Agrawal, Manoj Prabhakaran, XiaoFeng Wang, Erman Ayday, Jean-Pierre Hubaux, and Carl A. Gunter. Controlled functional encryption. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 1280–1291. ACM, 2014.

[Nec94] V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, Feb 1994.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 458–467, 1997.

[NSW$^+$17] Kobbi Nissim, Thomas Steinke, Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, David O'Brien, and Salil Vadhan. Differential privacy: A primer for a non-technical audience (preliminary version), 2017.

[Oce] Ocean Protocol Foundation Ltd. A decentralized data exchange protocol to unlock data for artificial intelligence (technical primer). https://oceanprotocol.com/, Accessed 4. October 2020.

[OLJ$^+$19] Boris Otto, Dominik Lis, Jan Jürjens, Jan Cirullies, Falk Howar, Sven Meister, Markus Spiekermann, Heinrich Pettenpohl, Frederik Möller, Jakob Rehof, and Sebastian Opriel. Data ecosystems. conceptual foundations, constituents and recommendations for action. Technical report, Fraunhofer Institute for Software and Systems Engineering ISST, 10 2019.

[OS07] Rafail Ostrovsky and William E. Skeith. A survey of single-database private information retrieval: Techniques and applications. In *Public Key Cryptography – PKC 2007*, pages 393–411. Springer Berlin Heidelberg, 2007.

[PCY$^+$17] Geong Sen Poh, Ji-Jian Chin, Wei-Chuen Yau, Kim-Kwang Raymond Choo, and Moesfa Soeheila Mohamad. Searchable symmetric encryption: Designs and challenges. *ACM Comput. Surv.*, 50(3):40:1–40:37, 2017.

[PSS09] Annika Paus, Ahmad-Reza Sadeghi, and Thomas Schneider. Practical secure evaluation of semi-private functions. In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *Applied Cryptography and Network Security, 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2-5, 2009. Proceedings*, volume 5536 of *Lecture Notes in Computer Science*, pages 89–106, 2009.

[QD11] Jun-lei Qian and Xiao-lei Dong. Fully secure revocable attribute-based encryption. *Journal of Shanghai Jiaotong University (Science)*, 16:490–496, 2011.

[RNS11] Sushmita Ruj, Amiya Nayak, and Ivan Stojmenovic. Dacc: Distributed access control in clouds. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 91–98, 2011.

[RR17] Peter Rindal and Mike Rosulek. Improved private set intersection against malicious adversaries. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 235–259, 2017.

[SAB15] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*. IEEE, aug 2015.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, pages 256–266, 1997.

[SPS14] Emil Stefanov, Charalampos Papamanthou, and Elaine Shi. Practical dynamic searchable encryption with small leakage. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[SSW12] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *Advances in Cryptology–CRYPTO 2012*, pages 199–217. Springer, 2012.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[SWP00] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 14-17, 2000*, pages 44–55, 2000.

[Tze04] Wen-Guey Tzeng. Efficient 1-out-of-n oblivious transfer schemes with universally usable parameters. *IEEE Trans. Computers*, 53(2):232–240, 2004.

[Val76] Leslie G. Valiant. Universal circuits (preliminary report). In Ashok K. Chandra, Detlef Wotschke, Emily P. Friedman, and Michael A. Harrison, editors, *Proceedings of the 8th Annual ACM Symposium on Theory of Computing, May 3-5, 1976, Hershey, Pennsylvania, USA*, pages 196–203. ACM, 1976.

[vLSD$^+$10] Peter van Liesdonk, Saeed Sedghi, Jeroen Doumen, Pieter H. Hartel, and Willem Jonker. Computationally efficient searchable symmetric encryption. In *Secure Data Management, 7th VLDB Workshop, SDM 2010, Singapore, September 17, 2010. Proceedings*, pages 87–100, 2010.

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography–PKC 2011*, pages 53–70. Springer, 2011.

[WLWG11] Guojun Wang, Qin Liu, Jie Wu, and Minyi Guo. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Computers & Security*, 30(5):320–331, 2011.

[XWSW16] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans. Parallel Distrib. Syst.*, 27(2):340–352, 2016.

[YG15] Attila Altay Yavuz and Jorge Guajardo. Dynamic searchable symmetric encryption with minimal leakage and efficient updates on commodity hardware. In *Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers*, pages 241–259, 2015.

[YJRZ13] Kan Yang, Xiaohua Jia, Kui Ren, and Bo Zhang. DAC-MACS: Effective data access control for multi-authority cloud storage systems. In *INFOCOM, 2013 Proceedings IEEE*, pages 2895–2903, 2013.

[YLL$^+$14] Yi Yang, Hongwei Li, Wenchao Liu, Haomiao Yao, and Mi Wen. Secure dynamic searchable symmetric encryption with constant document update cost. In *IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, December 8-12, 2014*, pages 775–780, 2014.

[Zhe20] Xiao Zheng. Data trading with differential privacy in data market. In *Proceedings of 2020 the 6th International Conference on Computing and Data Engineering*, ICCDE 2020, page 112–115, New York, NY, USA, 2020. Association for Computing Machinery.

[ZNP15] Guy Zyskind, Oz Nathan, and Alex Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *CoRR*, abs/1506.03471, 2015.

[ZWH$^+$15] Youwen Zhu, Zhikuan Wang, Bilal Hassan, Yue Zhang, Jian Wang, and Cheng Qian. Fast secure scalar product protocol with (almost) optimal efficiency. In Song Guo, Xiaofei Liao, Fangming Liu, and Yanmin Zhu, editors, *Collaborative Computing: Networking, Applications, and Worksharing - 11th International Conference, CollaborateCom 2015, Wuhan, China, November 10-11, 2015. Proceedings*, volume 163 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 234–242. Springer, 2015.