LWE-Based Encryption Schemes and Their Applications In Privacy-Friendly Data Aggregation

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von
Daniela BECKER

aus
Quakenbrück

2018

1. Gutachter: Prof. Dr. Dr. habil. Karl-Heinz ZIMMERMANN,
Institut für Eingebettete Systeme, Technische Universität Hamburg
2. Gutachter: Prof. Dr. Chris BRZUSKA,
Department of Computer Science, Aalto University

Tag der mündlichen Prüfung: 10.07.2018

Vorsitzende des Prüfungsausschusses:
Prof. Dr. Sibylle SCHUPP, Institut für Softwaresysteme, Technische Universität Hamburg

Abstract

LWE-Based Encryption Schemes and Their Applications In Privacy-Friendly Data Aggregation

by Daniela BECKER

Since its introduction in 2005, the Learning With Errors (LWE) problem has had a profound impact in both the theoretical and the applied crypto world with a growing number of theoretical results and corresponding applications. The reason for the increased interest in the LWE problem is its hardness with respect to the lattice problems Decisional Approximate Shortest Vector Problem and Approximate Shortest Independent Vector Problem in the worst case. Thus, as a result of the LWE hardness assumption, LWE-based cryptographic systems are conjectured to be post-quantum secure.
In this thesis we consider two problems: privacy-preserving data aggregation and solutions for privacy-friendly social media marketing. The former problem was introduced by Shi *et al.* (NDSS, 2011). The authors provide a first solution to the sum aggregation problem with a scheme that is based on the Decisional Diffie-Hellman problem. Their solution can handle only a very limited plaintext space, i.e. binary inputs.
In the first part of this dissertation, we extend the plaintext space. Similar to Valovich (CoRR, 2016), we leverage a variant of the LWE problem, which is inherently additively homomorphic. In contrast to Valovich, our LWE variant does not incur parameter increases due to reductions. Our scheme performs significantly better in terms of both runtime and bandwidth efficiency. In particular, it allows for roughly 66000 times larger plaintexts while improving on decryption runtime by a factor of about 150 compared to Shi *et al.*'s scheme.
In the second part of this work, we apply our scheme in the context of digital advertising: we combine it with a lattice-based signature scheme and provide the first solution for social media marketing that preserves the privacy of the users. Our construction has strong privacy and security guarantees and ensures cryptographic verifiability of the computed results.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Dr. Dr. Karl-Heinz Zimmermann, for his guidance and mentorship since the beginning of my academic career. He has supported my every step on this path, and I will be forever grateful to him for helping me to unfold my potential.
I would also like to thank my second reviewer, Prof. Dr. Chris Brzuska, for his input and for being especially forthcoming during the organization of the last part of this journey.
One of my first introductions to formal methods was given to me by Prof. Dr. Sibylle Schupp, who has not only taught me how to construct rigorous proofs of correctness but whose lessons have also evolved my logical way of thinking. Her advice and support have been invaluable to me.
I would like to thank the Director of the Bosch Research and Technology Center North America in Pittsburgh, Christopher Martin, for giving me the opportunity to pursue a Corporate PhD within his department. I feel lucky to have had such a strong advocate in all of my endeavors. His professional mentorship and leadership continue to inspire me.
I was fortunate to work in an environment, where I was surrounded by excellent colleagues and inspiring projects at Bosch CR/RTC3 and more generally at Robert Bosch LLC. In particular, I would like to thank my supervisor Dr. Jorge Guajardo Merchan for his continuous guidance, support, and expert input. He has been a role model for my development into a full-fledged scientist.
Among the many incredible researchers that I have met, I would like to thank Prof. Dr. Manuel Blum for introducing me to theoretical cryptography; it has been an honor to learn from him. I would also like to thank Prof. Dr. Avrim Blum for our fruitful conversations, which led me to consider the application of Private Stream Aggregation.
Furthermore, I would like to thank Ron van den Akker and my incredible team at CollAction for allowing me to help launch and grow a unique project in my free time, which has been a very rewarding undertaking and which also created a meaningful counterbalance to my research work.
Last but most certainly not least, I owe my deepest gratitude to my family for their endless love and support. They have always been my bridge over troubled water.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *"One must acknowledge with cryptography no amount of violence will ever solve a math problem."* - Jacob Appelbaum [Ass+16].

1.1 Motivation

A lattice is an elegant geometric construct: it can be thought of as the set of intersection points of an infinite grid in multi-dimensional space. The concept of lattices and their associated problems have fascinated cryptographers for decades: earliest work on using lattice problems for cryptography dates back to 1997 when Ajtai and Dwork [AD97] proposed a lattice-based public-key encryption scheme following Ajtai's [Ajt96] seminal worst-case to average-case reductions for lattice problems. Concretely, Ajtai [Ajt96] showed that if there is no efficient algorithm that approximates the decision version of the Shortest Vector Problem (SVP)[1] with a polynomial approximation factor, then it is hard to solve the associated search problem exactly over a *random* choice of the underlying lattice[2] [MG02].

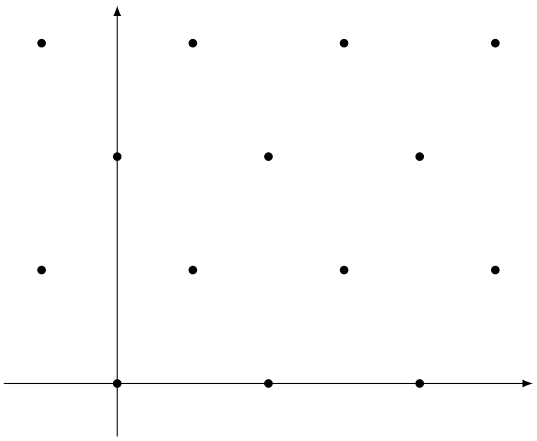

FIGURE 1.1: Example of a basic two-dimensional lattice

Observe that this relationship between the worst-case complexity of the former and the average-case complexity of the latter problem, is very useful from a cryptographic perspective: basing a cryptosystem on the latter

[1] Informally, SVP describes the following problem: given the basis of a lattice, find its shortest lattice vector. We refer to Section 2.1.1 for the formal definitions of some relevant related problems.

[2] Note that this lattice has to be chosen from a certain distribution that is easily sampleable as described by Ajtai [Ajt96].

problem would imply that on average, i.e. over a random choice of the selected inputs, e.g. the cryptographic keys, breaking the system is as hard as solving the former problem in its worst case. Therefore, Ajtai's [Ajt96] reduction created the first cryptographically meaningful lattice-based hardness assumption, which has become essential in proving the security of any lattice-based cryptographic construction.
While the strength of the underlying assumption was a great advancement from a theoretical perspective, the practicality of the scheme was significantly limited: with large key and ciphertext sizes and correspondingly slow encryption and decryption operations, it was considered impractical. Although later constructions (e.g. [Reg03]) greatly improved on these constraints, lattice-based schemes long kept the reputation for being inefficient. The introduction of the *Learning With Errors* (LWE) problem in 2005 by Regev [Reg05] overhauled this thinking: Regev's seminal work proposed a mathematical problem that has the rare property of being in the average case as hard as certain lattice problems in the *worst case*. Therefore, any LWE-based cryptosystem has security properties that are based on the hardness of worst-case lattice problems. At the same time, it can leverage the beautifully simplistic structure of the LWE problem, which allows to significantly improve efficiency.
There is now a consistently growing number of applications of the LWE problem both in theory and practice: from novel definitions of one-way functions and trapdoor constructions, to encryption and signature schemes, including some highly sought-after applications in identity-based encryption and fully homomorphic encryption.
Lattice-based cryptography, and specifically LWE-based cryptography, has received a lot of additional attention due to the fact that cryptography based on worst-case lattice problems is conjectured to be *post-quantum secure*, i.e. it remains secure against quantum adversaries. However, this insight has also lead to a race for more efficient LWE-based schemes in order to compete with other post-quantum secure solutions or directly with currently used classically secure schemes like RSA.
In this work, we construct novel LWE-based encryption schemes and formally analyze their correctness and security guarantees. With respect to our focus on *privacy-preserving data aggregation*, we showcase a particularly well-suited use case of LWE-based encryption due to the inherent properties of the LWE problem. We further show, how our resulting scheme can be efficiently applied to *privacy-preserving advertising in social media*, which has not been considered before.

1.2 Contributions

We summarize our contributions as follows.

Lattice-Based Private Stream Aggregation

- In Chapter 3 we introduce a new lattice-based Private Stream Aggregation (PSA) scheme called LaPS. We are able to resolve a main problem from Shi *et al.* [Shi+11]. In particular, our scheme allows for any plaintext size in contrast to Shi *et al.*'s [Shi+11] PSA scheme, which only allows for very small (i.e. binary) plaintext space. We

achieve this by leveraging a variant of the LWE problem as a hardness assumption. In contrast to Valovich's [Val16] LWE-based PSA scheme, our choice of LWE variant allows us to take full advantage of LWE's additively homomorphic properties and encrypt more efficiently. Our PSA scheme accomplishes higher bandwidth efficiency than the state-of-the-art while maintaining the same Differential Privacy-guarantees and providing the strong security notion of (conjectured) post-quantum security.

- LaPS's general design does not restrict the noise distribution to a particular privacy mechanism and we account for potential improvements in the ever evolving field of homomorphic encryption. We allow for the replacement of the additively homomorphic scheme that is part of our construction in a straightforward manner.

- We extend Shi *et al.*'s [Shi+11] PSA scheme to support multiple encryptions in contrast to their encrypt-once model, which limits the users to a single encryption per execution of the scheme.

- We instantiate our scheme with a reduced version of the BGV [BGV12] encryption scheme and the discrete Laplace privacy mechanism and we implement this instantiation, which to the best of our knowledge is the first implementation of a lattice-based PSA scheme. Our experimental results show that our scheme is practical. Moreover, it outperforms previous works in several aspects. First, because our construction is optimized to support a single operation (i.e. additive homomorphism), we are able to significantly reduce the BGV parameters by multiple orders of magnitude compared to [Dam+13]. Furthermore, we achieve 150 times faster decryption for the overall PSA scheme, while providing over 4 orders of magnitude larger plaintexts compared to [Shi+11].

Privacy-Preserving Social Media Marketing

- In Chapter 4 we consider the problem of social media advertisement and formally analyze it. Our architecture SOMAR achieves privacy of end user data in the Differential Privacy-sense and complies with the following requirements: users can make social-media induced purchases, merchants can sponsor influencers to advertise their products on their social media sites and influencers can receive aggregate user data about their followers.

- In SOMAR we eliminate the existing trust assumptions between a merchant and an influencer and replace it by cryptographic proofs of correctness. Therefore, we achieve verifiable data aggregation in the social media marketing model.

- In a concrete instantiation we show that our LaPS scheme can be directly applied to our SOMAR architecture by extending it with a lattice-based homomorphic aggregate signature scheme [Jin14], which also yields (conjectured) post-quantum security. Our experimental results show practicality of our construction.

We refer readers who are not familiar with lattice theory and the LWE literature to Chapter 2, which covers basic results and hardness theorems used in the chapters of this work. We end this thesis in Chapter 5, where we summarize our results and make some recommendations for future research.

Chapter 2

Lattices and Learning With Errors

The *Learning With Errors* (LWE) problem was introduced by Regev in his seminal work [Reg05] more than a decade ago. The interest in LWE and its variants originates from the problem's worst-case/average-case hardness and its conjectured post-quantum computer hardness due to its relation to the mathematical notion of *lattices* and lattice problems. At the same time, its compact structure can be more efficiently utilized within cryptosystems than previously used lattice problems. In this chapter we provide basic facts and terminology from lattice theory and introduce the LWE problem. Note that we restrict our treatment of lattice theory to the minimum necessary to understand the hardness properties of the LWE problem. For a comprehensive treatment of lattice theory we refer to [MG02].
The following lattice definitions and explanations are based on [MG02] and [Pei16].

2.1 Lattices

Formally, an n-*dimension*al lattice Λ is a subset of $\mathbb{R}^n$ that is an *additive subgroup* and *discrete*:

$$\Lambda = \Lambda(\mathbf{B}) := \mathbf{B} \cdot \mathbb{Z}^k = \Big\{ \sum_{i=1}^{k} z_i \mathbf{b}_i : z_i \in \mathbb{Z} \Big\},$$

where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]$ is the non-unique *basis* consisting of linearly independent basis vectors $\mathbf{b}_i$. The lattice is therefore generated as the set of all integer linear combinations of the basis vectors and k denotes the *rank* of Λ. We have already seen an example of a lattice in Figure 1.1, which shows a simple 2-dimensional lattice. A generally common example for a lattice is the integer lattice $\mathbb{Z}^n$.
The notion of the ith *successive minimum* λ_i for $i \in \{1, \ldots, n\}$ describes the smallest possible radius of a sphere that is centered in the origin such that i linearly independent lattice vectors in Λ are contained in it. Consequently, λ_1 corresponds to the length of the shortest lattice vector, the so-called *minimum distance* of Λ:

$$\lambda_1(\Lambda) := \min_{\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}} ||\mathbf{v}||.$$

We state "smoothing properties" of a lattice Λ using its *smoothing parameter* η_ϵ next, which is parametrized by the positive real tolerance $\epsilon > 0$. Intuitively, η_ϵ captures the amount of "*Gaussian "blur" required to "smooth out"*

all the discrete structure of Λ" [Pei16]. Note that $\log$ denotes the logarithm to base 2 unless noted otherwise.

Lemma 1 ([MR04, Lemma 3.3])**.** *For any* n*-dimensional lattice* Λ *and positive real* $\epsilon > 0$*, the smoothing parameter is at most*

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n(\Lambda).$$

In particular, $\eta_\epsilon(\Lambda) \leq \omega(\sqrt{\log n}) \cdot \lambda_n(\Lambda)$ *for some negligible function* $\epsilon(n) = n^{-\omega(1)}$*.*

2.1.1 Lattice Problems

We highlight the following two lattice problems due to their particular significance with regards to the LWE problem: the *decisional approximate Shortest Vector Problem* (GapSVP) and the *approximate Shortest Independent Vectors Problem* (SIVP)[1]. Note that both GapSVP and SIVP are *approximation* problems, where $\gamma(n) \geq 1$ denotes the *approximation factor* with respect to the lattice dimension n, i.e. perfect accuracy is achieved with $\gamma(n) = 1$.
GapSVP is a promise problem associated to SVP (see Section 1.1) that asks to distinguish between a Yes- and a No-instance. In this case, Yes-instances mean that for the given lattice basis $\mathbf{B}$ and a rational value $r \in \mathbb{Q}$, there exists a shortest lattice vector whose length is at most r. Conversely, No-instances represent the statement that all lattice vectors associated to the given basis $\mathbf{B}$ are strictly longer than $r \cdot \gamma(n)$. Note that in below definition r is set to 1.

Definition 1 (Decisional Approximate Shortest Vector Problem (GapSVP_γ))**.** *Given basis* $\mathbf{B}$ *for some lattice* $\Lambda = \Lambda(\mathbf{B})$ *with dimension* n*, decide whether* $\lambda_1(\Lambda) \leq 1$ *or* $\lambda_1(\Lambda) > \gamma(n)$*.*

SIVP intuitively asks to find a set of n linearly independent lattice vectors such that each vector is at most as long as the lattice's nth successive minimum $\lambda_n(\Lambda)$. The approximate version of the problem only requires the lattice vectors' lengths to be individually at most the approximation factor $\gamma(n)$ longer than $\lambda_n(\Lambda)$.

Definition 2 (Approximate Shortest Independent Vector Problem (SIVP_γ))**.** *Given a basis* $\mathbf{B}$ *for some full-rank lattice* $\Lambda = \Lambda(\mathbf{B})$*, i.e. where rank* $k =$ *dimension* n*, find a set* $\mathbf{S} = \{\mathbf{s}_i\} \subset \Lambda$ *of* n *linearly independent lattice vectors* $\mathbf{s}_i$*, where* $||\mathbf{s}_i|| \leq \gamma(n) \cdot \lambda_n(\Lambda)$ *for all* i*.*

2.1.2 Lattice-Based Cryptography

The following overview summarizes the main theoretical advances in the area of lattice-based cryptosystems - essentially up to the introduction of the LWE problem in 2005. We follow [Pei16] in our presentation.

[1] It appears to be unclear when these problems were first formulated. According to Ajtai [Ajt96] lattice problems related to finding a shortest vector in a lattice were first considered by Dirichlet in 1842. Ajtai [Ajt96] provides a somewhat more general formulation of what is now called the unique Shortest Vector Problem (unique-SVP). GapSVP and SIVP are related problems, we use the formulation due to [Pei16].

After Ajtai's [Ajt96] seminal presentation of worst-to-average-case reductions for lattice problems in 1996, the follow-up construction of a lattice-based public-key encryption scheme due to Ajtai and Dwork [AD97] was celebrated as a great theoretical advancement (it was later further improved by Regev [Reg03]). However, there was a growing desire for lattice-based schemes with better efficiency in terms of key and ciphertext sizes as well as runtimes.
One of the first attempts, the NTRU encryption scheme, introduced by Hoffstein, Pipher and Silverman [HPS98] in 1998, and its revisions ended up having a somewhat opposite problem: while it was considered comparatively efficient due to the use of algebraically structured lattices, i.e. by leveraging polynomial rings, its theoretical underpinning was never proven to be linkable to worst-case lattice problems. The only exception is the NTRU version introduced by Stehlé and Silverstein [SS11] in 2011, who reduce its security to the Ring-LWE problem (see Section 2.4.5), however with much larger parameters than the original, which negatively impacted the efficiency of the scheme.
Similarly, the GGH encryption and signature schemes [GGH97] did not provide a worst-case security proof at first (and were later broken in this initial form [Ngu99; NR06]). However, the idea of generating a "good" lattice basis consisting of short basis vectors and a "bad" basis with long and non-orthogonal lattice vectors for the same lattice, where the latter can be efficiently generated from the former but not vice-versa, became the central concept in developing lattice-based trapdoor functions. These trapdoor functions remain a crucial element of a myriad of modern lattice-based cryptographic constructions, such as the GPV signature scheme [GPV08].
Another important result on the way to LWE-based encryption, more precisely Ring-LWE-based encryption, is the one-way function due to Micciancio [Mic02; PR06; LM06]: it is defined over polynomial rings and its hardness is reduced from worst-case lattice problems over cyclic lattices. Previously quasi-quadratic key sizes were thereby reduced to quasi linear, which significantly improved the efficiency of the construction and any derived lattice-based schemes.

2.1.3 Post-Quantum Security

The topic of lattices, especially in its application to cryptography, gained wide popularity among academia and industry with the surge of developments around *quantum computers* (see e.g. [Wil11; BR18; Ibm]). While the creation of a fully functional quantum computer will be a break-through beyond technology, the literal "quantum leap" in computing power will immediately put the majority of currently deployed encryption techniques and security systems in jeopardy. As most of our known and used cryptographic systems are based on security notions that are *breakable* by quantum computers, e.g. by solving the factoring problem, these constitute a threat to global security architectures in their current form [Sho97; Ber09].
Although the commercial off-the-shelf availability of such a quantum computer is currently estimated to be little under a decade away [Bau+16], the research community has been actively looking for quantum-*secure* (or

quantum-*resistant*) solutions, which lead to the coining of the term *post-quantum cryptography*[2]. These cryptosystems are considered secure against quantum adversaries, since the *Shor algorithms* [Sho97] could not be applied, which efficiently solve the discrete logarithm problem and prime factorization using a "*hypothetical quantum computer*" [Sho97]. Hence, different from currently widely used cryptosystems like RSA or ECDSA, post-quantum schemes have not been found to be breakable by the Shor algorithms and are therefore conjectured to be secure both against classical and quantum computers.
Lattice-based cryptography is believed to belong to this category[3]. While all of the currently known post-quantum secure options have individual advantages and drawbacks, the particular attraction of lattice-based cryptography stems firstly, from the availability of worst-to-average-case reduction proofs. For instance, the LWE problem, which is as hard as worst-case lattice problems but more efficient in practice, can be used to formulate an appropriate hardness assumption. Secondly, practical applications of lattice-based cryptography are not restricted to encryption schemes or signature schemes alone but are versatile in that they cover the entire range of cryptographic systems.
Therefore, the constructions that we discuss in this work are indeed conjectured to be post-quantum secure. Nevertheless, we aim to show their immediate applicability as our lattice-based schemes improve on existing (classical) solutions both in terms of efficiency and breadth of functionalities.

2.2 Learning With Errors (LWE)

When *Learning With Errors* (LWE) was first introduced in the celebrated work of Regev [Reg05] in 2005, it was formulated as a generalization of the *Learning from Parity with Noise* (LPN) problem [BKW03]. LPN had been around for a few years at this time and had already built a reputation as a novel hardness assumption giving rise to a plethora of cryptographic constructions and applications (see e.g. [Pie12] for an overview of LPN-based systems). LWE was viewed as a breakthrough: Regev [Reg05] showed that the LWE problem can be reduced from the lattice problems GapSVP and SIVP in the *worst case*. As described previously, this implies LWE's conjectured post-quantum hardness.
In the following, we first present the basic structure of the LWE problem before highlighting the specific parameter instantiation defined by Regev [Reg05] that allows for the desired reduction from worst-case lattice problems (Section 2.2.1). We also discuss LWE's security guarantees from a practical perspective (Section 2.2.2). Subsequently, we present a selection of LWE-based encryption schemes (Section 2.3), before we review a number of relevant variations of the LWE problem and highlight their individual properties (Section 2.4). Note that the LWE problem has been defined and formulated in various different formats - here we follow the notation used

[2]Daniel J. Bernstein seems to have introduced the term in 2003 [BL16]. The first PQCrypto-conference took place in 2006.

[3]Other post-quantum solutions are *code-based* and *multivariate* cryptography (see e.g. [Ber09] for an overview).

by Regev [Reg09] unless noted otherwise.
The LWE problem describes the task of solving the following system of equations:

$$(\mathbf{a}_1, b_1 = \langle \mathbf{a}_1, \mathbf{s} \rangle + e_1)$$
$$\vdots$$
$$(\mathbf{a}_m, b_m = \langle \mathbf{a}_m, \mathbf{s} \rangle + e_m).$$

The *coefficients* $\mathbf{a}_i$ are drawn uniformly at random from $\mathbb{Z}_q^n$, multiplied with the wanted *secret* $\mathbf{s} \in \mathbb{Z}_q^n$ and subsequently perturbed by adding some *error* e_i. The latter is drawn from some error distribution χ. n is the secret's dimension and the *security parameter*, m determines the number of samples, and all operations are performed over $\mathbb{Z}_q$.
The LWE problem can also be formulated in more compact matrix notation, i.e. given $(\mathbf{A}, \mathbf{b})$ s.t. $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \chi^m$, recover $\mathbf{s}$. Note that the $x \xleftarrow{\$} S$ operation denotes choosing x from the uniform distribution over S.
Concretely, there are two problems associated to LWE: search LWE asks to recover the secret vector $\mathbf{s}$ as described above; decision LWE on the other hand asks to distinguish between a tuple $(\mathbf{a}, b)$ sampled from the LWE distribution $\mathcal{A}_{s,\chi}$ and a tuple sampled uniformly at random from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. We formally summarize these notions in Definition 3.

Definition 3 (LWE problem [Reg05; Reg09])**.** *Let n, m, $q = q(n) \leq \mathsf{poly}(n)$ be integers, and χ be some probability distribution over $\mathbb{Z}_q$. Then, $\mathcal{A}_{s,\chi}$ denotes the LWE distribution that is obtained by generating tuples of the form $(\mathbf{a}_i, \langle \mathbf{a_i}, \mathbf{s} \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where vectors $\mathbf{a}_i, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and error $e_i \in \mathbb{Z}_q$ is drawn according to distribution χ.*
Given some m samples from $\mathcal{A}_{s,\chi}$, search *$LWE_{q,\chi}$ describes the problem of recovering $\mathbf{s}$.*
Given a sample in $\mathbb{Z}_q^n \times \mathbb{Z}_q$, decision *$LWE_{q,\chi}$ describes the problem of determining whether it was sampled according to $\mathcal{A}_{s,\chi}$ or drawn uniformly at random from $\mathbb{Z}_q^n \times \mathbb{Z}_q$, respectively.*

Regev shows that for $n \geq 1$ and $2 \leq q \leq \mathsf{poly}(n)$, where q is a prime, both problems are equally hard except with negligible probability.
We adopt the convention that when referring to LWE, the search version is meant. Furthermore note that we may abuse notation and highlight certain parameters by adding them as a subscript, e.g. $LWE_{n,m,q,\chi}$.

2.2.1 Hardness

A remarkable property of the LWE problem is its reducibility from worst-case lattice problems under a certain parametrization. Regev initially shows this in the following setting: Let $\bar{\Psi}_\alpha$ be a distribution over $\mathbb{Z}_q$ that is shaped like the *discrete Gaussian* distribution that is centered around 0 with standard deviation αq, where $\alpha \in \mathbb{R}^+$ and all samples are reduced modulo 1, i.e. pick a number from the interval $[0, 1)$ according to the Gaussian distribution, multiply it by q and take the nearest integer [Reg09].

Note that Regev's [Reg05] original hardness result is provided for the continuous Gaussian distribution. More recent definitions of LWE-based systems typically refer directly to the discrete Gaussian as in Definition 4. The previously described naive method of rounding to the nearest integer gives an intuition for discretization. However, this method does not produce a true discrete Gaussian, as Lindner and Peikert [LP11] remark. Peikert [Pei10] provides an appropriate randomized rounding method.

Definition 4 (Discrete Gaussian Distribution [LP11]). *For a lattice Λ and a positive real $\sigma > 0$, the discrete Gaussian distribution $D_{\Lambda,\sigma}$ over Λ with parameter σ is the probability distribution having support Λ that assigns a probability proportional to $\exp(-\pi||\mathbf{x}||^2/\sigma^2)$ to each $\mathbf{x} \in \Lambda$.*

When the LWE error is drawn from the distribution $\bar{\Psi}_\alpha$ with standard deviation αq, where $\alpha \in (0,1)$ and $\alpha q > 2\sqrt{n}$, efficiently solving LWE implies an efficient quantum solution for GapSVP and SIVP over n-dimensional lattices up to an approximation factor $\gamma = \tilde{\mathcal{O}}(n/\alpha)$ in the worst case. This culminates in the LWE-assumption.

Lemma 2 (LWE-assumption [Reg05, Theorem 1.1]). *For integers n, q and $\alpha \in (0,1)$ s.t. $\alpha q > 2\sqrt{n}$, if there exists a PPT algorithm solving $LWE_{q,\bar{\Psi}_\alpha}$, then there exists an efficient quantum algorithm that approximates the decisional GapSVP and the SIVP problem on n-dimensional lattices to within $\gamma = \tilde{\mathcal{O}}(n/\alpha)$ in the worst case.*

Note that the standard deviation of the error distribution αq determines the magnitude of the error in the equation system. As Micciancio and Peikert [MP13] point out, the relation of (roughly) $q \geq \sqrt{n}/\alpha$ is the tightest possible in order to obtain the relation to worst-case lattice problems and therefore *optimal*.
Observe that Regev's quantum reduction from lattice problems to LWE is formulated for the search version of the problem. Peikert *et al.* [PRSD17] recently showed a result that is identical to Lemma 2 but directly extends to decision LWE.
As mentioned previously, the LWE-assumption has emerged as a novel hardness assumption, that has been since utilized to prove security of various cryptosystems (see [Pei16] for an overview - we discuss a selection of encryption schemes in more detail in Section 2.3).

2.2.2 Practical Security of LWE-Based Systems

Besides its connection to worst-case lattice problems, the concrete security of an LWE-based system highly depends on the parameters used in a particular instantiation. In particular, the modulus q, the security parameter n, which corresponds to the secret key dimension, and the Gaussian parameter σ, impact the concrete *bit-security* of a given encryption scheme. As recently analyzed in Herold *et al.*'s [HKM18] work, which surveys the existing solution algorithms for LWE, the asymptotic complexity of solving LWE is $2^{\mathcal{O}(n)}$, regardless of the approach, i.e. whether lattice-based or combinatorial techniques are used. However, actual runtimes of the individual algorithms reveal that "*LWE's complexity changes as a function of the LWE-parameters*" [HKM18]. Therefore, the general idea is to apply the best known attacks to the LWE problem and thereby determine lower bounds

for the parameter instantiation. In this section, we give an overview of known attacks and discuss Lindner and Peikert's [LP11] results in more detail as they are currently considered the baseline[4] for the computation of LWE bit-security levels.

Distinguishing attack. The *distinguishing attack* [MR09; RS10] is directed at decision LWE and aims to distinguish between LWE and uniformly random samples. It reduces LWE to the *Short Integer Solution* (SIS) problem and attacks the SIS-instances. The SIS problem [Ajt96] describes the task of finding a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ that satisfies a given norm such that $\mathbf{Az} = \sum_i \mathbf{a}_i \cdot z_i = \mathbf{0} \in \mathbb{Z}_q^n$, where matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is composed of m uniformly random column vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ [Pei16, Definition 4.1.1]. The SIS problem also reduces from worst-case lattice problems and it can be seen as dual to LWE. We omit the details of the algorithm here as attacks on search LWE are known to be inherently more powerful, since they actually recover the secret vector[5] [LP11].

Combinatorial attack. Some *combinatorial* attacks have been proposed to solve LWE. The deployed algorithms are generally a derivation of the *BKW* algorithm [BKW03; Wag02], which actually targets the LPN problem. Since Regev [Reg05] introduced LWE as a generalization of LPN, the generalized BKW algorithm also solves LWE. It was later improved in several other works [Alb+14; APS15; KF15] but the general structure remained mostly the same. Given an LWE instance $\{(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)\}$, in a first stage, the left-hand side of the LWE equations, i.e. the coefficient vectors $\mathbf{a}_i$, are reduced in dimension. This results in a decrease of the "bias" of the right-hand sides b_i, which in the second stage serves to distinguish between LWE samples and uniform samples. The algorithm also has asymptotic complexity $2^{\mathcal{O}(n)}$ but to date its runtimes were not able to outperform *lattice basis reduction* techniques [LP11], which we detail next.

Lattice basis reduction. The *LLL* algorithm was introduced by Lenstra, Lenstra and Lovász [LLL82]. It takes a lattice basis $\mathbf{B}$ as input and returns an LLL-reduced basis for $\Lambda(\mathbf{B})$, resulting in basis vectors that are very short and almost orthogonal. The guarantee of the LLL-reduction is that the output contains a lattice vector that is at most $\gamma(n)\lambda_1$ long, where $\gamma(n)$ denotes the approximation factor as before. On a high level, the algorithm iterates through all input basis vectors pairwise, reduces them and orders each pair by length, until they cannot be reduced anymore [MG02]. The *BKZ* algorithm due to Schnorr and Euchner [SE94] is a blockwise generalization of this approach and is considered the best approximation algorithm in high dimension according to [CN11]. Chen and Nguyen [CN11] significantly improve the BKZ algorithm's runtime in their implementation by incorporating the pruning technique due to Gama *et al.* [GNR10].

[4] Note that this is the case even though there have been follow-up works that improve over the efficiency of Lindner and Peikert's [LP11] attack, e.g. [LN13].

[5] In addition, the presented decoding attack from Lindner and Peikert [LP11] also yields a significantly higher advantage in solving search LWE while providing a better time/advantage ratio than the distinguishing attack.

Lattice reduction and decoding attack. Lindner and Peikert's [LP11] findings in particular have shaped the understanding of how LWE parameters need to be instantiated in order to provide certain bit-security guarantees. The authors combine lattice basis reduction techniques with *bounded-distance decoding* and attack the standard search LWE problem as defined by Regev [Reg05]. This combination is especially efficient as the reduced basis from the first step is used in order to execute the decoding attack in the second part. Observe that lattice basis reduction is individually considered more efficient than a combinatorial attack like the BKW algorithm and their decoding attack alone is more powerful than the distinguishing attack. Therefore, Lindner and Peikert's [LP11] attack leverages "the best of the best" algorithms and thereby beats previous proposals somewhat automatically.

For the decoding part of the attack Lindner and Peikert [LP11] extend the *nearest-plane* algorithm by Babai [Bab85] and adapt it to the particular Gaussian distribution of the error-term in LWE. In fact, the LWE problem can be formulated as a *Bounded-Distance Decoding* (BDD) problem, where given a lattice basis and a target point with a certain guaranteed distance to the lattice, the task is to find the unique lattice vector that is closest to the target point [Pei16, Definition 2.2.5]. The right-hand side $\mathbf{b}$ of an LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$ is the target point and the lattice is defined as $\Lambda = \Lambda(\mathbf{A})$, where $\mathbf{As} \in \Lambda$ [LP11; Pei16]. Therefore, solutions to the decoding problem also provide solutions to the search LWE problem.

Babai's [Bab85] nearest-plane algorithm expects a lattice basis $\mathbf{B}$ and a target point $\mathbf{t}$ as inputs and returns a lattice point $\mathbf{v}$ that is somewhat close to the target. More specifically, the output $\mathbf{v}$ is indeed the desired unique lattice point iff it is close enough to the fundamental parallelepiped of the orthogonalized[6] basis vectors $\tilde{\mathbf{B}}$. Observe that this means that LWE-error $\mathbf{e}$ would have to lie within that parallelepiped. Lindner and Peikert [LP11] generalize the nearest-plane algorithm in such a way that the shape of this parallelepiped is most likely to yield the correct solution. They achieve this by recursing over several distinct planes that are chosen to "*capture the most probability mass of the Gaussian error distribution of* $\mathbf{e}$" [LP11].

Note that the decoding algorithm works best if the input basis $\mathbf{B}$ is maximally reduced. Therefore, Lindner and Peikert [LP11] first run a BKZ-reduction[7] as implemented by Shoup in the NTL library [Sho] before inputting the result into the decoding algorithm.

Finally, they obtain the runtime results of running this attack on different parameter sets and thereby determine lower bounds on the respective values for secret dimension n, modulus q and Gaussian parameter σ.

Arora-Ge attack. Finally, the *Arora-Ge attack* due to Arora and Ge [AG11] exploits an unbounded number of LWE-samples by using a linearization technique, which reduces the problem of solving an LWE-equation system to solving a linear equation system. They achieve complexity $2^{\tilde{\mathcal{O}}(\sigma^2)}$, which

[6] $\tilde{\mathbf{B}}$ is the Gram-Schmidt orthogonalization of the vectors in $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ and the fundamental parallelepiped is defined as $\mathcal{P}_{1/2}(\mathbf{B}) := \mathbf{B} \cdot [-\frac{1}{2}, \frac{1}{2})^k = \{ \sum_{i \in [k]} c_i \cdot \mathbf{b}_i : c_i \in [-\frac{1}{2}, \frac{1}{2})\}$, see e.g. [LP11].

[7] It is noteworthy that Lindner and Peikert [LP11] combine the notions of the *Hermite factor* [GN08] and the *Geometric Series Assumption* [Sch03] as a quality measure of the reduced basis. We refer to [LP11, Section 5.1] for more details.

is subexponential for $\sigma \leq \sqrt{n}$ and exponential for $\sigma > \sqrt{n}$, where σ is the Gaussian parameter and n is the secret dimension, as before. Consequently, this imposes a lower bound on the error magnitude, which is essentially defined by σ. Micciancio and Mol [MM11] were the first to propose limiting the number of available LWE-samples in order to mitigate this attack, i.e. limit parameter m. Micciancio and Peikert [MP13] later continued to explore this idea and indeed show that LWE remains hard even for small errors when the number of samples is limited accordingly (see Section 2.4.2).

We refer to [HKM18, Table 1] for an asymptotic comparison of the mentioned attacks and their concrete significance in terms of parameters.

2.3 Encryption Schemes

Over time LWE has given rise to a myriad of different cryptosystems - together with a growing family of LWE-variants, the number of resulting applications has only increased. Our results will mainly leverage LWE-based *encryption*. Hence in this section, we present a selection of LWE-based encryption schemes, which have shaped the state-of-the-art of lattice-based encryption.

2.3.1 Regev's Encryption Scheme

With the introduction of the LWE problem and the proof of its relation to worst-case lattice problems, Regev [Reg05] also proposed the following public-key encryption scheme, which is still a go-to basis for modern lattice-based cryptosystems. We here present its definition using the more compact matrix notation as shown in [AGV09].

Definition 5 (Regev's Encryption Scheme (RPKE) [Reg05; AGV09])**.** *For the public key encryption scheme RPKE = (*RGen, REnc, RDec*), let $m(n), q(n)$ and $\alpha(n)$ be parameters of the scheme, where n is the security parameter. $q(n)$ is a prime between n^2 and $2n^2$, $m(n) = (1+\epsilon)(n+1)\log q$ for some constant ϵ and $\alpha(n) = o(1/(\sqrt{n}\log n))$. All additions are performed over $\mathbb{Z}_q$.*

- RGen(1^n) *randomly selects a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, a vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and a vector $\mathbf{e} \leftarrow \bar{\Psi}_\alpha^m$, i.e. each entry e_i is chosen independently from the probability distribution $\bar{\Psi}_\alpha$. Output $pk = (\mathbf{A}, \mathbf{As}+\mathbf{e})$ and $sk = \mathbf{s}$.*

- REnc$(pk, \mu \in \{0,1\})$*, where μ is the bit to be encrypted: Pick a random vector $\mathbf{r} \in \{0,1\}^m$. Output $\mathbf{c} = (\mathbf{c}_0, c_1) = (\mathbf{rA}, \mathbf{r}(\mathbf{As}+\mathbf{e}) + \mu\lfloor\frac{q}{2}\rceil)$ as the ciphertext.*

- RDec$(sk, \mathbf{c})$ *computes $\mu' = |c_1 - \mathbf{c}_0 \cdot \mathbf{s}|$. Output 0 if μ' is closer to 0 than to $\lceil\frac{q}{2}\rceil \mod q$, and 1 otherwise.*

Note that the parameters in Definition 5 guarantee correctness and semantic security under the LWE-assumption (Lemma 2). A critical aspect is the resulting performance: the public key size is $\mathcal{O}(mn\log q) = \tilde{\mathcal{O}}(n^2)$ and the encryption blowup is a factor of $\mathcal{O}(n\log q) = \tilde{\mathcal{O}}(n)$. Additionally, the plaintext size is limited to a single bit. As Regev [Reg09] points out, one may assume that the users of the scheme share the public matrix $\mathbf{A}$ beforehand [Ajt05].

Then pk would only consist of $\mathbf{As} + \mathbf{e}$ and the public key size is reduced to $\mathcal{O}(m \log q) = \tilde{\mathcal{O}}(n)$.
Peikert *et al.* [PVW08] observe that parts of the public key pk and the randomness $\mathbf{r}$ in the encryption step can be securely reused $l = \mathcal{O}(n)$ number of times. By taking advantage of this fact they reduce the encryption blowup to $\mathcal{O}(1)$ and encrypt n-bit messages at essentially the same cost as 1-bit messages in Regev's scheme. Consequently, both the secret $\mathbf{S} \in \mathbb{Z}_q^{n \times l}$ and error $\mathbf{E} \in \mathbb{Z}_q^{l \times m}$ are matrices - as opposed to vectors. They propose the following multi-bit encryption scheme.

Definition 6 (Multi-bit Encryption [PVW08])**.** *For the public key encryption scheme MPKE = (*MGen, MEnc, MDec*), let $m(n), q(n), p(n)$ and $\alpha(n)$ be parameters of the scheme, where n is the security parameter. $q \geq 4pm$ is a prime, $p(n) = \mathsf{poly}(n) \geq 2$ is an integer and $\alpha \leq 1/(p\sqrt{mg})$, where $g(n) = \omega(\sqrt{\log n})$ and $m \geq 3(n + l) \log q$.*
The amortization factor is denoted by integer $l(n) = \mathcal{O}(n) \geq 1$. The domain of messages lies in $\mathbb{Z}_p^l$. All operations are performed over $\mathbb{Z}_q$.

- MGen(1^n) *picks matrices $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times l}$ each uniformly at random. Choose $\mathbf{E} \leftarrow \bar{\Psi}_\alpha^{l \times m}$ where each entry $e_{i,j}$ is drawn independently from the probability distribution $\bar{\Psi}_\alpha$. Output pk = ($\mathbf{A}$, $\mathbf{P} = \mathbf{S}^T\mathbf{A} + \mathbf{E}$) and sk = $\mathbf{S}$.*

- MEnc$(pk, \mathbf{v})$*, where $\mathbf{v} \in \mathbb{Z}_p^l$ is the message to be encrypted: Pick a vector $\mathbf{e}$ at random from $\{0,1\}^m$. Output $(\mathbf{u}, \mathbf{c}) = (\mathbf{Ae}, \mathbf{Pe} + \mathbf{t})$ as the ciphertext, where $t(v) = \lfloor v \cdot \frac{p}{q} \rceil \in \mathbb{Z}_q$ and $\mathbf{t} = t(\mathbf{v}) = (t(v_1), \ldots, t(v_l))^T$.*

- MDec$(sk, (\mathbf{u}, \mathbf{c}))$ *computes $\mathbf{w} = \mathbf{c} - \mathbf{S}^T\mathbf{u}$. Output $\mathbf{v}'$, where each $\mathbf{v}'_i$ is s.t. $\mathbf{w}_i - t(\mathbf{v}'_i)$ is closest to 0.*

Note that the public key size remains asymptotically the same as in Regev's scheme at $\tilde{\mathcal{O}}(n^2)$.
Again, semantic security holds based on the LWE-assumption (Lemma 2).

2.3.2 LP Encryption Scheme

Lindner and Peikert [LP11] achieve a significant improvement in terms of key size: compared to RPKE, concrete key sizes in their encryption scheme LP are "*up to 10 times smaller*" [LP11] while achieving a higher bit-security level, where they compare to the parameters presented in [MR09]. The LP encryption scheme, which we restate in the following, is considered the most efficient LWE-based public-key encryption scheme.

Definition 7 (LP [LP11])**.** *For the public key encryption scheme LP = (*LGen, LEnc, LDec*), let n_1, n_2, q, l and s_k, s_e be parameters of the scheme where $q \geq 2$, $n_1, n_2 \geq 1$, $l \geq 1$ and $s_k \cdot s_e \leq \frac{\sqrt{2}\pi}{c} \cdot \frac{t}{\sqrt{(n_1+n_2)\cdot \ln(2/\delta)}}$ for some $c \geq 1$ and $\delta > 0$.*
Let encode $: \Sigma \to \mathbb{Z}_q$ *and* decode $: \mathbb{Z}_q \to \Sigma$ *be error-tolerant encoding and decoding functions such that* decode(encode$(m) + e \mod q) = m$ *for any integer $e \in [-t, t)$ where $t \geq 1$ is the error tolerance. Component-wise application allows for encoding and decoding of vectors.*
$\mathbf{A} \in \mathbb{Z}_q^{n_1 \times n_2}$ is a matrix that is chosen uniformly at random and shared among all users.

- $\mathsf{LGen}(1^l)$ *samples* $\mathbf{R} \leftarrow D_{\mathbb{Z},s_k}^{n_1 \times l}$ *and* $\mathbf{S} \leftarrow D_{\mathbb{Z},s_k}^{n_2 \times l}$. *Output* $pk = \mathbf{P} = \mathbf{R} - \mathbf{AS}$ *and* $sk = \mathbf{S}$.
- $\mathsf{LEnc}(pk, \mathbf{m})$, *where* $\mathbf{m} \in \Sigma^l$ *is the message to be encrypted: Draw vectors* $\mathbf{e}_1 \in \mathbb{Z}^{n_1}, \mathbf{e}_2 \in \mathbb{Z}^{n_2}, \mathbf{e}_3 \in \mathbb{Z}^l$ *according to* $D_{\mathbb{Z},s_e}$. *Compute* $\bar{\mathbf{m}} = \mathsf{encode}(\mathbf{m})$ *and output the ciphertext* $(\mathbf{c}_1 = \mathbf{e}_1^t\mathbf{A} + \mathbf{e}_2^t, \mathbf{c}_2 = \mathbf{e}_1^t\mathbf{P} + \mathbf{e}_3^t + \bar{\mathbf{m}}^t)$.
- $\mathsf{LDec}(sk, (\mathbf{c}_1, \mathbf{c}_2))$ *outputs* $\mathsf{decode}(\mathbf{c}_1^t \cdot \mathbf{S} + \mathbf{c}_2^t)^t$.

For alphabet $\Sigma = \{0, 1\}$, the authors give the following example for the error-tolerant encoder and decoder: $\mathsf{encode}(m) := m \cdot \lfloor \frac{q}{2} \rfloor$ and $\mathsf{decode}(\bar{m}) := 0$ if $\bar{m} \in [-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor) \subset \mathbb{Z}_q$, and 1 otherwise. The error tolerance is $t = \lfloor \frac{q}{4} \rfloor$. LP is secure under the LWE-assumption (Lemma 2) and keys and ciphertexts are roughly of size $2n^2 \log q$ for $n_1 = n_2 = n$.

2.4 LWE Variants

Although LWE has a distinctly simple structure, the instantiation of LWE-based schemes raised some efficiency concerns in practice: They are generally speaking more efficient than previously known lattice-based cryptosystems. However, taking into account asymptotic key and ciphertext lengths alone, LWE-based encryption schemes are simply incomparable to commonly used systems like RSA. This is primarily due to the fact that for "just one extra pseudo-random number" LWE-based encryption requires "n extra random numbers" [Reg10]. Therefore, early LWE-based encryption schemes only allowed for 1 bit at a time-encryption with comparatively large key and ciphertext lengths, i.e. Regev's [Reg05] encryption scheme (see Section 2.3.1). Additionally, choosing an exponential modulus q causes the resulting LWE components to grow in magnitude. Lastly, running a discrete Gaussian sampler in order to sample the errors in LWE is generally more complex than sampling uniformly at random, i.e. leading to longer encryption run-times [CGW14]. In the urge of closing the gap between theory and practice, new versions of LWE have been introduced by breaking the problem down into its components, exchanging parts, and putting them back together.
Here we provide an overview of some proposed problem variants. We only present a small subset of all existing LWE variants that are relevant in our context. Note that while the initial motivation for creating new versions of LWE was mainly efficiency improvement, over the last decade a countless number of variations has been proposed. The majority of variants were developed for special-case applications and some could eventually not compete with the efficiency of the original definition of LWE.
We structure the findings according to the components of LWE, i.e. *coefficients* (Section 2.4.1), *error* (Section 2.4.2) and *secret* (Section 2.4.3). In fact, some results from Section 2.4.2 and Section 2.4.3 have also been combined in an effort to jointly improve the outcome, which we evaluate in Section 2.4.4.

2.4.1 Coefficients

Observing that the coefficients in the LWE problem, i.e. the matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ in an LWE-instance $(\mathbf{A}, \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, take up most space when stored in memory, Galbraith [Gal13] proposed to draw $\mathbf{A}$ from the

binary instead of the q-ary field: He shows that any standard LWE-instance $(\mathbf{A}, \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ can be formulated as an instance of his variant $(\mathbf{A}', \mathbf{A}'\mathbf{s}' + \mathbf{e}')$, where $\mathbf{A}' \in \{0,1\}^{m \times n'}$, at the cost of increasing the secret's dimension $n' = n\lfloor \log q \rfloor$.
We restate the definition of this LWE-variant below, which we denote *Learning With Errors from Parity* (LWEP). The name reflects the fact that this variant is a hybrid between the LWE and the LPN problem [BKW03]. The latter is a special case of LWE, in which all components are binary.

Definition 8 (Learning With Errors from Parity (LWEP) [Gal13])**.** *Let n, m be integers, q be a prime and χ be some probability distribution over $\mathbb{Z}_q$. Then, $L_{s,\chi}$ denotes the LWEP distribution that results from taking tuples of the form $(\mathbf{a}_i, \langle \mathbf{a_i}, \mathbf{s} \rangle + e_i) \in \{0,1\}^n \times \mathbb{Z}_q$, where vector $\mathbf{a}_i \xleftarrow{\$} \{0,1\}^n$, vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and error $e_i \in \mathbb{Z}_q$ is drawn according to distribution χ.*
Given some m samples from $L_{s,\chi}$, the Learning With Errors from Parity problem[8] *$LWEP_{n,m,q,\chi}$ describes the problem of recovering* $\mathbf{s}$.

Galbraith [Gal13] proposes an LWEP-based version of Regev's [Reg05] encryption scheme (see Section 2.3.1), where the ciphertexts $\mathbf{c} = (\mathbf{c}_0, c_1) = (\mathbf{rA}, \mathbf{r}(\mathbf{As} + \mathbf{e}) + \mu\lfloor \frac{q}{2} \rceil)$ look just like in Regev's scheme and the error is sampled from the Gaussian distribution $\bar{\Psi}_\alpha$, except that $\mathbf{A}$ is binary. Note that if vector $\mathbf{r}$ would be known to the adversary, this would be sufficient to recover the message, simply by subtracting $\mathbf{r}(\mathbf{As} + e)$ from the right-hand side c_1. Galbraith [Gal13] considers different lattice-based attacks to retrieve $\mathbf{r}$ from the left-hand side $\mathbf{c}_0 = \mathbf{rA}$ and concludes that LWEP is safe to use for encryption under a certain parameter setting. He gives concrete guidelines for parameter magnitudes and suggests that for $(n, m) = (256, 400)$ and $(n, m, q, \sigma) = (256, 640, 4093, 3.33)$ his LWEP-based encryption provides moderate and high security, respectively.
However, Herold and May [HM17] recently broke Galbraith's encryption scheme and were able to recover the plaintext message from the LWEP-based ciphertexts. While Galbraith [Gal13] regarded the problem of computing $\mathbf{r}$, given $\mathbf{rA}$, as a *vectorial integer subset sum* problem that should be solved by finding a closest vector in the corresponding lattice, Herold and May [HM17] recognized that the problem can be formulated as an *Integer Linear Programming* (ILP) problem. An ILP problem asks to find an integral solution $\mathbf{r} \in \mathbb{Z}^m$ for an equation system of m linear equations over the integers. They solve this problem in polynomial time by removing the integral requirement of the solution using an LP relaxation. Ultimately, they break both the moderate- and high security-LWEP instances provided by Galbraith [Gal13]. In particular, the authors find that for $m \leq 2n$, LWEP-based encryption as defined by Galbraith [Gal13] is especially easy to break and therefore insecure. Note that Herold and May's [HM17] results only break this particular instantiation of LWEP and not the hardness of the LWEP problem itself. This is why, they are only able to recover the plaintext message and not the secret, as the authors remark [HM17].

[8] Note that we here focus on the search problem - the decision variant can be defined analogously to decision LWE (Definition 3).

2.4.2 Error

While the original definition of LWE requires the error $\mathbf{e}$ in an LWE-instance $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ to be drawn from the discrete Gaussian distribution $\bar{\Psi}_\alpha$, the correct implementation of the sampling process itself is non-trivial and may negatively affect performance in practice [Fol14; Saa15; CGW14; DM13]. This lead to the investigation of alternative error distributions - in particular whether the error could be securely sampled from the *uniform* distribution. Note that $\mathcal{U}(S)$ denotes the uniform distribution over S.
An LWE-variant, where the error is sampled from a (small) uniform distribution, is proposed by both Micciancio and Peikert [MP13], and Döttling and Müller-Quade [DM13], however with slightly different results, which we discuss in the following.

Lemma 3 (LWE with uniform error [MP13, Theorem 4.6])**.** *Let $0 < k \leq n \leq m - \omega(\log k) \leq k^{\mathcal{O}(1)}$, $l = m - n + k$, $s \geq (Cm)^{l/(n-k)}$ for a large enough constant C and q be a prime such that $\max\{3\sqrt{k}, (4s)^{m/(m-n)}\} \leq q \leq k^{\mathcal{O}(1)}$.*
For any set $X \subseteq \{-s, \ldots, s\}^m$ where $|X| \geq s^m$, if there exists a PPT algorithm solving $LWE_{q,\chi=\mathcal{U}(X)}$, then there exists an efficient quantum algorithm that solves worst-case lattice problems on k-dimensional lattices to within approximation factor $\gamma = \tilde{\mathcal{O}}(\sqrt{k}/q)$.

In direct comparison to the traditional LWE-assumption (Lemma 2), the approximation factor γ, i.e. the accuracy of solving the respective worst-case lattice problem, remains roughly the same. However, depending on the value of k, the underlying lattice assumption in Lemma 3 becomes potentially stronger: since k is sub-linear in n, the dimension of the lattice problem is smaller than in Lemma 2.
Note that the parameter settings in Lemma 3 also allow for *binary* errors by setting $s = 2$ and $X = \{0, 1\}$ and still achieve the same hardness guarantees with regards to worst-case lattice problems as stated in Lemma 4. However, since there is a dependency between s and k, this impacts the dimension of the underlying worst-case lattice problem.

Lemma 4 (LWE with binary error [MP13, Theorem 1.2])**.** *Let security parameter n and $m = n \cdot (1 + \Omega(1/\log n))$ be integers and $q \geq n^{\mathcal{O}(1)}$ be a sufficiently large polynomially bounded prime modulus. If there exists a PPT algorithm solving $LWE_{q,\chi=\mathcal{U}(\{0,1\})}$, then there exists an efficient quantum algorithm that solves worst-case lattice problems on $\Theta(n/\log n)$-dimensional lattices to within approximation factor $\gamma = \tilde{\mathcal{O}}(\sqrt{n} \cdot q)$.*

From a formal point of view, Micciancio and Peikert [MP13] resort to the SIS problem in order to prove their results and achieve the above notion. In contrast, Döttling and Müller-Quade [DM13] utilize the notion of *lossy codes* in order to formulate a similar version of LWE with uniform error as stated in Lemma 5. A lossy code is essentially a pseudorandom code, i.e. indistinguishable from a random code, that when used for encoding provably annihilates the message after adding a certain error.

Lemma 5 (LWE with uniform error [DM13, Theorem 1])**.** *Let $q(n)$ be the modulus and $m(n) = \mathsf{poly}(n)$ be an integer with $m \geq 3n$ where n is the security parameter. Let $c \in (0, 1)$ be an arbitrarily small constant.*
For $\rho(n) \in (0, 1/10)$ such that $\rho q \geq 2n^{0.5+c}m$, if there exists a PPT algorithm

that solves $LWE_{q,\mathcal{U}([-\rho q,\rho q])}$, then there exists an efficient quantum algorithm that solves worst-case lattice problems on $n/2$-dimensional lattices to within approximation factor $\gamma = \tilde{\mathcal{O}}(n^{1+c}m/\rho)$.

As Micciancio and Peikert [MP13] point out, the main difference is in the magnitude of the error: while according to Lemma 3 and 4 the error can be shrunk to being binary (and is always smaller than $\sqrt{n}$), Lemma 5 requires the error to be at least roughly $\sqrt{n} \cdot m$ due to the constraint on ρq. In fact this is also larger than the lower bound on the error magnitude imposed by the original LWE-assumption (Lemma 2).
Both works require a bounded number of LWE-samples m due to the Arora-Ge attack [AG11] (see Section 2.2.2). As the error magnitude directly relates to the number of samples, Döttling and Müller-Quade's [DM13] result allows for a larger m, namely polynomial in n.
Fuller *et al.* [FMR13] observe that as long as the given LWE-instance remains an under-determined equation system with regards to the secret, a small number of elements of the error-vector can be securely set to 0. In their resulting variant of LWE they utilize the notion of a *symbol-fixing source*, which denotes a distribution that outputs α fixed symbols and m random samples over a pre-defined alphabet $\mathcal{Z}$.

Lemma 6 (LWE with some fixed errors [FMR13, Theorem 5.2])**.** *Let m, α be polynomial in n, $q = \mathsf{poly}(n)$ be a prime and $\beta \in \mathbb{Z}^+$ such that $q^{-\beta} = \mathsf{negl}(n)$, where n is the security parameter. For the uniform distribution $\mathcal{U}$ over $\mathcal{Z}^m$, an alphabet $\mathcal{Z} \subset \mathbb{F}_q$ and an $(m+\alpha, m, |\mathcal{Z}|)$ symbol-fixing source W over $\mathcal{Z}^{m+\alpha}$, if there exists a PPT algorithm that solves decisional $LWE_{n+\alpha+\beta,m+\alpha,q,W}$, then there exists a PPT algorithm that solves decisional $LWE_{n,m,q,\mathcal{U}}$.*

Fixing parts of the error vector implies extending the secret's dimension and the overall number of provided samples, accordingly. The authors also generalize the above result to hold for arbitrary (hence not necessarily uniform) distributions over $\mathbb{F}_q$.
Note that the above result can only enjoy hardness based on worst-case lattice problems, when properly linked to the LWE-assumption (Lemma 2): this can be achieved by setting the error distribution to be uniform as in Lemma 6, which results in basing hardness on LWE with uniform errors. The latter, in turn, is known to be as hard to solve as standard LWE due to Lemma 3 or 5 depending on the chosen parameters. Alternatively, one may choose the original discrete Gaussian distribution, such that m entries are chosen according to the discrete Gaussian and the remaining α symbols are fixed. Consequently, Fuller *et al.*'s [FMR13] result essentially states that solving LWE is still hard even when a few components of the error-vector are set to 0.

2.4.2.1 Augmented LWE (A-LWE)

The possibility of increasing the amount of data that can be hidden inside an LWE-term is explored by El Bansarkhani *et al.* [EDB15], who use *message embedding* where auxiliary information is placed into the error-term. They construct the *Augmented* LWE (A-LWE) problem, where essentially the error term $\mathbf{e} \in \mathbb{Z}_q^m$ is indistinguishable from a discrete Gaussian distributed vector but effectively encodes some message $\mathbf{m} \in \{0,1\}^m$. Consequently,

the search version of A-LWE has two variants, namely search-m A-LWE and search-s A-LWE, which denote recovering either message $\mathbf{m}$ or secret $\mathbf{s}$ from an A-LWE sample, respectively.
Note that El Bansarkhani *et al.* [EDB15] utilize the concept of a *gadget matrix*, which is denoted $\mathbf{G}$ in the following. It was first introduced by Micciancio and Peikert [MP12] and it is computed using the Kronecker product $\otimes$ of the identity matrix $\mathbf{I}$ and the vector $\mathbf{g}$, which is constructed as detailed next.

Definition 9 (A-LWE problem [EDB15])**.** *Let n, m, q, l, x be integers, where $l = \lceil \log q \rceil$ and $m = x \cdot l$. Let $H : \mathbb{Z}_q^n \to \{0,1\}^m$ be some function. Let $\mathbf{g}^T = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_q^l$ and $\mathbf{G} = \mathbf{I}_{m/l} \otimes \mathbf{g}^T \in \mathbb{Z}_q^{m/l \times m}$.*
For $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, define the A-LWE *distribution $L_{n,m,\sigma}^{\text{A-LWE}}(\mathbf{m})$ with $\mathbf{m} \in \{0,1\}^m$ to be the distribution over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ obtained as follows:*

- *Set $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m}) \in \mathbb{Z}_q^{m/l}$.*
- *Sample $\mathbf{e} \leftarrow D_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),\sigma} \in \mathbb{Z}_q^m$.*
- *Return $(\mathbf{A}, \mathbf{b}^T)$ where $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$.*

Given polynomially many samples from $L_{n,m,\sigma}^{\text{A-LWE}}(\mathbf{m})$ and input $\mathbf{m} \in \mathbb{Z}_q^{m/l}$, search-s *A-LWE$_{n,m,\sigma}^H$ describes the problem of recovering $\mathbf{s}$.*
Given polynomially many samples from $L_{n,m,\sigma}^{\text{A-LWE}}(\mathbf{m})$, search-m *A-LWE$_{n,m,\sigma}^H$ describes the problem of recovering $\mathbf{m}$.*
Given a sample in $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$, decision *A-LWE$_{n,m,\sigma}^H$ describes the problem of determining whether it was sampled according to $L_{n,m,\sigma}^{\text{A-LWE}}(\mathbf{m})$ or drawn uniformly at random from $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q$, respectively.*

The corresponding security properties with respect to the LWE-assumption are stated next.

Lemma 7 (A-LWE-assumption [EDB15, Theorem 2])**.** *Let κ be the security parameter. Let $n, m, q, l = \lceil \log q \rceil$ be integers, $H : \mathbb{Z}_q^n \to \{0,1\}^m$ be a hash function modeled as a random oracle and $\mathbf{G}$ be the gadget matrix $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^T$ where $\mathbf{g}^T = (1, \ldots, 2^{l-1})$. For a real $\epsilon = \mathsf{negl}(\kappa) > 0$, let $\sigma \geq \eta_\epsilon(\Lambda^{\perp}(\mathbf{G}))$, let $\mathbb{H}_\infty(\mathbf{s}) > \kappa$. Then, if there exists a PPT algorithm that solves* search-s *A-LWE$_{n,m,\sigma}^H$, then there exists a PPT algorithm that solves LWE$_{n,m,\sigma}$. If there exists a PPT algorithm that solves* decision *A-LWE$_{n,m,\sigma}^H$ or* search-m *A-LWE$_{n,m,\sigma}^H$, then there exists a PPT algorithm that solves* decision *LWE$_{n,m,\sigma}$.*

In other words, A-LWE terms are essentially indistinguishable from LWE-terms (as long as the error distribution is properly shaped) and inherits the security properties of LWE while allowing to embed a message into the error term. Note that different from most other LWE-variants, this reduction does not require the costly increase of parameters.
Additionally, El Bansarkhani *et al.* [EDB15] show a straightforward way of using this message embedding technique for encryption, where a plaintext message $\mathbf{m}$ is embedded into the error-term of an A-LWE term, which then constitutes the ciphertext. In fact, we will leverage this approach in our constructions, which we present in Chapter 3.

2.4.3 Secret

Goldwasser *et al.* [Gol+10] investigate the hardness of the LWE-problem in the presence of leakage, i.e. when parts of the secret are leaked to the adversary. They conclude that LWE remains hard to solve even if the secret is sampled from an arbitrary distribution, as long as some minimum entropy is guaranteed. We restate this "minimum" version of LWE next.

Lemma 8 (Entropy-k LWE [Gol+10, Theorem 4])**.** *Let $n', q \geq 1$ be integers where q is super-polynomial in n', $\alpha, \beta > 0$ such that $\alpha/\beta = \mathsf{negl}(n')$ and $\mathcal{D}$ be an arbitrary distribution over $\{0,1\}^{n'}$ with min-entropy at least k where n' is the security parameter. For any $n \leq \frac{k-\omega(\log n')}{\log q}$ if there exists a PPT algorithm that solves decisional $LWE_{n',q,\beta}(\mathcal{D})$, i.e. where the secret $\mathbf{s}$ is sampled from $\mathcal{D}$, then there exists a PPT algorithm that solves decisional $LWE_{n,q,\alpha}$.*

This means that LWE is *leakage resilient*, i.e. secret information can be leaked to a certain extent without compromising security. Observe that the above reduction comes at the cost of increasing the secret's dimension from n to $n' \approx n \log q$.
Subsequently, Brakerski *et al.* [Bra+13] followed by Micciancio [Mic18] give a somewhat stronger reduction for decisional LWE with binary secrets, i.e. binLWE, which is essentially shown to be equivalent to traditional decisional LWE. In what follows, the distribution $\mathcal{A}^{\pm}$ is defined analogously to the LWE distribution $\mathcal{A}$ (Definition 3) but with the secret sampled from the binary distribution.

Lemma 9 (binLWE [Mic18, Theorem 4.1])**.** *Assume the distribution $\mathcal{A}_{q,k,n+1,\sigma}$ is pseudorandom for some $\sigma > \omega(\sqrt{\log n})$, $k \geq \omega(\log n)$, and $(n+1) \geq (k+1) \cdot (\log(q)+1)$. Then the distribution $\mathcal{A}^{\pm}_{q,n,n^{\mathcal{O}(1)},\sigma'}$ is also pseudorandom for $\sigma' = 2\sigma\sqrt{n+1}$.*

Note that the transformation to the LWE-assumption also requires a secret dimension shift from n to k, which is considered likely "*unavoidable, as it preserves the bit-length of the secret*" [Mic18].

2.4.4 Error and Secret

By "*putting the LWE distribution into Hermite normal form*" [App+09] Applebaum *et al.* construct a variant of LWE, where *both* the error and the secret are sampled from the error distribution and show that this variant is equivalent to standard LWE.

Lemma 10 (Error and secret from error distribution [App+09, Lemma 1])**.** *Let $q = p^e$ be a prime power, if there exists a PPT algorithm solving decisional $LWE_{q,\mathbf{s}',\mathbf{e}}$, where $\mathbf{s}' \leftarrow \chi^n$ and $\mathbf{e} \leftarrow \chi^m$, then there exists a PPT algorithm that solves decisional $LWE_{q,\mathbf{s},\mathbf{e}}$, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \chi^m$.*

This result has great significance, since it can be applied to virtually any variation of LWE that modifies the error distribution. In fact, it opened up the possibility of combining previously unrelated variants.
For example, Cabarcas *et al.* [CGW14] use exactly this technique and combine Micciancio and Peikert's [MP13] result on LWE with small uniformly distributed errors (Lemma 3) with Applebaum *et al.*'s [App+09] finding

(Lemma 5) that the secret in the LWE-term can be securely sampled from the same distribution as the error. Consequently, they construct an LWE-version, where both the secret and the error are sampled from a uniform distribution over a set of relatively small values.
While this "chaining" of results has become a convenient proof technique commonly used with LWE-based cryptosystems, some LWE-variants impose specific parameter restrictions that amplify through the transformations and eventually trickle down to the regular LWE-assumption, which causes significantly increased parameter magnitudes. In the case of Cabarcas *et al.*'s [CGW14] LWE with small uniform secret and error, they instantiate Lindner and Peikert's [LP11] encryption scheme LP (see Section 2.3.2) with their LWE variant, resulting in the new scheme U-LP, and realize that the resulting efficiency has in fact *de*creased.
Although modifying the underlying LWE problem of the LP-scheme was intended to improve the efficiency of the scheme by avoiding the need for Gaussian sampling, it turns out that it actually had a negative impact on performance. U-LP requires a larger modulus and bigger error magnitude than LP: $q = \mathcal{O}(n^{3.7})$ and $t = \mathcal{O}(n^{1.4})$ where $\mathcal{U}(\{0, \ldots, t-1\})$ is the uniform distribution for the secret and the error in U-LP - compared to $q = \mathcal{O}(n^2)$ and Gaussian parameter $s_e = \mathcal{O}(n^{1/2})$ in LP.
Consequently, the performance gain from using the uniform distribution over the discrete Gaussian distribution is cancelled out by the large parameters required to establish a sufficient security level. This in turn results in larger key and ciphertext sizes and slower encryption and decryption times as Cabarcas *et al.*'s [CGW14] experimental results show. As an additional drawback, U-LP restricts the message space to binary strings compared to an unrestricted alphabet in LP.

2.4.5 Ring-LWE

While most LWE-variants attempt to improve efficiency in one way or another, the LWE-version that is generally considered most competitive is somewhat orthogonal to all previously discussed variants: *Ring*-LWE (R-LWE). Given n random numbers, LWE cannot produce $\mathcal{O}(n)$ pseudo-random numbers "in one shot". Instead, an additional set of n random numbers is required to output a new pseudo-random number [Reg10]. Lyubashevsky *et al.* [LPR10] recognized this fact as the main inefficiency of LWE-based schemes. Hence, with R-LWE they found a way to define a vector multiplication operation such that the resulting distribution is indeed pseudo-random, namely using multiplication over the polynomial *ring*. We state the formal problem definition next[9].

Definition 10 (Ring-LWE (R-LWE) [LPR10])**.** *For a ring R of degree n over $\mathbb{Z}$, let $R_q = R/qR$ be the quotient ring with integer modulus $q \geq 2$ and let χ denote a distribution over the ring R.*
Then, $\mathcal{R}_{s,\chi}$ denotes the R-LWE distribution that is obtained by generating tuples of the form $(a, b = s \cdot a + e \mod q) \in R_q \times R_q$, where $a \in R_q$ and $s \in R_q$ are sampled uniformly at random and e is drawn according to distribution χ.

[9] Note that we follow the notation from Peikert in [Pei16] for better readability. As he points out, there is a slight difference in writing from the original definition [LPR10], where a certain fractional ideal $R^\vee$ is used instead of R, which is its dual. However, this "tweaked" version is equivalent to the original form as Peikert [Pei16] also describes.

Given some m samples from $\mathcal{R}_{q,\chi}$, search R-LWE$_{q,\chi}$ *describes the problem of recovering s (fixed for all samples).*
Given a sample of the form $(a_i, b_i) \in R_q \times R_q$, decision R-LWE$_{q,\chi}$ *describes the problem of determining whether it is distributed according to either $\mathcal{R}_{s,\chi}$ for a uniformly random $s \in R_q$ (fixed for all samples) or drawn uniformly at random from $R_q \times R_q$, respectively.*

R-LWE is considered to be analogous to LWE from a hardness perspective because it has its own reduction from worst-case lattice problems: R-LWE is known to be as hard as the SIVP problem over *ideal* lattices [LPR10]. Therefore, there is no loss in parameters due to additional reduction from the original LWE-assumption.
In what follows, the K-SIVP$_\gamma$ problem is defined analogously to the SIVP problem as in Definition 2 but works over a fractional ideal $\mathcal{I}$ in some number field K that is endowed with some geometric norm as opposed to a general lattice. Consequently, the desired output is a set of n linearly independent elements in $\mathcal{I}$ whose norms are all at most $\gamma \cdot \lambda_n(\mathcal{I})$ [LPR10, Definition 2.10].

Lemma 11 (R-LWE-assumption [LPR10], [PRSD17, Corollary 6.3])**.** *Let K be an arbitrary number field of degree n and $R = \mathcal{O}_K$. Let $\alpha = \alpha(n) \in (0, 1)$, and let $q = q(n) \geq 2$ be an integer such that $\alpha q \geq \omega(1)$. There is a polynomial-time quantum reduction from K-SIVP$_\gamma$ to (average-case,* decision*) R-LWE$_{q,\Upsilon_\alpha}$ for any*

$$\begin{aligned}\gamma &= \max\left\{\omega(\sqrt{n}/\alpha) \cdot \eta(\mathcal{I})/\lambda_n(\mathcal{I}), \sqrt{2}n/(\lambda_1(\mathcal{I}^\vee)\lambda_n(\mathcal{I}))\right\} \\ &\leq \max\left\{\omega(\sqrt{n\log n}/\alpha), \sqrt{2}n\right\},\end{aligned}$$

where a distribution sampled from Υ_α is an elliptical Gaussian[10]*.*

Note that although R-LWE is commonly instantiated over *cyclotomic* number fields, the above hardness result applies to any number field and any number setting. In fact this recent result from Peikert *et al.* [PRSD17] is especially remarkable as Lyubashevsky *et al.* [LPR10] originally only provided a quantum reduction from worst-case lattice problems to search R-LWE. An additional classical search-to-decision reduction allowed to argue worst-case hardness for decision R-LWE. However, this second reduction restricted parameters to "*prime moduli that split well*" [PRSD17] and cyclotomic number fields. Therefore, Peikert *et al.*'s [PRSD17] result (reflected in Lemma 11) was able to lift that restriction.
Since R-LWE is understood as the analogue of LWE in the ring setting and more compact, it has become common practice to propose an LWE-based scheme with an LWE-based security proof and implement it using R-LWE in order to improve efficiency and practicality. For instance Lindner and Peikert [LP11] show that for their encryption scheme LP (see Section 2.3.2) they are able to reduce the key sizes by a factor of at least 200 when moving from standard LWE to R-LWE while maintaining the same security level. In fact, with 2-5 kilobits the key sizes became comparable to modern implementations of RSA as the authors note [LP11].
Additionally, since the transformation to the ring setting is straightforward,

[10] We omit the details of the parameter setup for the error distribution here for better readability. We refer to [PRSD17, Definition 6.1] for more details.

virtually any LWE-variant can also be defined as a ring version: for example Brakerski and Vaikuntanathan [BV11b] utilize Applebaum *et al.*'s [App+09] result (Lemma 10) and base their schemes on an R-LWE version, where both the secret and the error are sampled from the error distribution. We will use this R-LWE variant due to [BV11b] in our schemes in Section 3.7. We state the problem's definition next.

Definition 11 (R-LWE variant [BV11b])**.** *For all $\kappa \in \mathbb{N}$, let $f(x) = f_\kappa(x) \in \mathbb{Z}[X]$ be a polynomial of degree $n = n(\kappa)$, let $q = q(\kappa) \in \mathbb{Z}$ be a prime integer, let $t = t(\kappa) \in \mathbb{Z}_q^*$ (thus t and q are relatively prime), let the ring $R = \mathbb{Z}[X]/\langle f(X)\rangle$ and $R_q = R/qR$, and let χ denote a distribution over the ring R.*
The decision *Ring-*LWE *problem asks to distinguish in polynomial time (in κ) between any $l = \mathsf{poly}(\kappa)$ samples $(a_i, a_i \cdot s + t \cdot e_i)_{i\in[l]}$ and l uniform random samples from $R_q \times R_q$, where s is sampled from the error distribution χ, a_i are uniform in R_q and the error polynomials e_i are sampled from the error distribution χ.*

Note that this R-LWE variant inherits the security properties of standard R-LWE (Lemma 11) [BV11b, Theorem 1].

Chapter 3

Privacy-Preserving Data Aggregation

Differential Privacy (DP) built a long research track record (see e.g. [Dwo08; DR14] for an overview) since its introduction by Dwork [Dwo06] in 2006[1]. It has recently regained interest in the wider technology community with companies such as Apple [App] and Google [EPK14; MR17] adopting DP techniques in their products, see e.g. [Gre16; Nov17].
Private Stream Aggregation (PSA) applies DP techniques to aggregation operations. We propose a novel lattice-based PSA scheme called LaPS, which extends secure sum aggregation for distributed privacy-sensitive data into the post-quantum age. We give a high-level overview of our approach (Section 3.1) and review related work (Section 3.2) before covering some background information on DP and the concept of PSA-schemes (Section 3.3). We recall Shi *et al.*'s [Shi+11] PSA scheme in particular, which was the first such proposal, as it will serve as the baseline for the evaluation of our results (Section 3.4).
Subsequently, we present our lattice-based PSA scheme LaPS. We leverage the A-LWE problem in combination with additively homomorphic encryption in order to satisfy strong security guarantees and provide increased efficiency and scalability compared to Shi *et al.*'s [Shi+11] PSA scheme.
Our definitions first come in a general flavor (Section 3.5). In particular, the choice of privacy-preserving noise is up to the application, i.e. it can be implemented in a plug-and-play manner. We formally analyze LaPS's security and privacy guarantees within this framework (Section 3.6). Additionally, we present a concrete instantiation of our scheme (Section 3.7) and its implementation, which demonstrates a performance gain in terms of decryption runtime of roughly 150 times compared to [Shi+11] for a plaintext space of 2^{16} and 1000 participants (Section 3.8). We also consider some extensions of our scheme (Section 3.9). Our findings were presented in [BGZ18].

3.1 Our Approach

PSA schemes aim to solve the problem of aggregating data from multiple users in a privacy-preserving way without placing trust in the aggregator. This is achieved by combining privacy-preserving techniques based on the addition of noise to each user's input and encryption of the resulting

[1] The term was first coined by Dwork in [Dwo06] but was actually a culmination of the following works [DN03; DN04; Blu+05; Dwo+06].

value. The aggregator can only decrypt the aggregate result, which inherently contains some noise and therefore preserves the privacy of each individual user. As mentioned previously, Shi *et al.*'s [Shi+11] scheme was the first proposition of a PSA scheme for sum aggregation. It is based on the Decisional Diffie-Hellman (DDH) assumption and the decryption routine requires solving a discrete logarithm problem. The scheme's parameters have to be chosen such that this step can be efficiently executed, which significantly limits the plaintext space (i.e. binary inputs). Resolving this issue is left as an open problem in [Shi+11].
In Section 2.4.2.1 we presented the A-LWE problem due to El Bansarkhani *et al.* [EDB15] as a variant of LWE, where the error term $\mathbf{e}$ can be utilized in order to embed a message $\mathbf{m}$ inside it. Recalling Definition 9, an A-LWE term $(\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$ is produced by first encoding message $\mathbf{m}$ into a pseudo-random vector $\mathbf{v}$ as $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m})$ before sampling $\mathbf{e}$ such that $\mathbf{e} \leftarrow D_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}),\alpha q}$. The authors show that this error distribution is indistinguishable from the standard discrete Gaussian distribution that is used in regular LWE. Using the relation $\mathbf{v} = \mathbf{G}\mathbf{e}$ one can recover the message, where $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^T$ is a gadget matrix (see Section 2.4.2.1). Subsequently, $\mathbf{v}$ is decoded to retrieve $\mathbf{m}$. Note that leveraging the error term essentially as a data container immediately allows for secure encryption. This concept is more efficient than adding a message to a freshly sampled LWE-term for each encryption, which corresponds to typical LWE-based encryption schemes such as Regev's encryption scheme (see Section 2.3.1).
Therefore, by using A-LWE for our PSA scheme we benefit from higher bandwidth-encryption and a simple decryption routine, which yields faster decryption runtime compared to previous approaches such as [Shi+11]. Note that the latter almost comes "for free" when moving away from the DDH-based system as in [Shi+11], since LWE-based decryption is much faster than solving a discrete logarithm problem, especially for a large plaintext space.

3.1.1 Naive Approach

At first, one might consider taking an A-LWE-based encryption scheme such as the generic scheme proposed by El Bansarkhani *et al.* [EDB15] and integrating it as the user's encryption operation in Shi *et al.*'s [Shi+11] PSA scheme - thereby replacing the DDH-assumption by the A-LWE-assumption and leveraging A-LWE's efficiency advantages.
However, this idea fails when aggregating the resulting ciphertexts: according to A-LWE's definition (Definition 9) each message $\mathbf{m}$ is encoded into a vector $\mathbf{v}$ in a one-time pad style. Unfortunately, the XOR-operation that is used for the encoding is not *additively homomorphic*, i.e. the sum of the ciphertexts will *not* correspond to the sum of the underlying plaintexts. Furthermore, decryption as formulated in El Bansarkhani *et al.*'s [EDB15] generic scheme requires knowledge of the secret $\mathbf{s}$. Therefore, the aggregator's decryption would necessitate having the user's secret key. However, this violates the basic requirement of PSA schemes as it would entail placing trust in the aggregator.

3.1.2 A solution that *does* work

In the naive construction it turns out that the main problem lies in the way the message is encoded in the A-LWE term. Therefore, instead of encoding as El Bansarkhani *et al.* [EDB15] define it, we generate the encoding $\mathbf{v}$ by applying *any* additively homomorphic function[2] to the message. From a security standpoint, the resulting vector $\mathbf{v}$ is only required to be indistinguishable from random. As long as this requirement is met, the hardness of the problem is preserved. In particular, A-LWE reduces from the LWE-assumption and by extension from worst-case lattice problems.
We achieve indistinguishability from random by formulating the encoding step as the encryption routine of an additively homomorphic encryption scheme with pseudorandom ciphertexts. The vector $\mathbf{v}$ then corresponds to the resulting ciphertext. Finally, correct decryption of this "inner" encryption allows the aggregator to correctly retrieve the aggregate result by decrypting the sum of the ciphertexts. Therefore, our generalized version of A-LWE maintains security while ensuring correct aggregation of the user's output ciphertexts. Note that we view this additively homomorphic encryption scheme as a building block in our general PSA scheme. As a result, our construction allows for a flexible instantiation of individual building blocks based on the requirements of the application.

Remark 1. *Note that the idea of using LWE for PSA schemes in itself is not new: Valovich [Val16] bases his PSA scheme on a variant of LWE, where the error in the LWE-term is drawn from a Skellam distribution [Ske46] and serves as privacy-preserving noise at the same time. Thus his scheme addresses both security and privacy at once. The LWE-based construction also allows him to resolve the open problem from [Shi+11] regarding the plaintext length.*
However, in order to prove hardness of his LWE variant, Valovich uses the lossy code construction due to Döttling and Müller-Quade [DM13] (see Section 2.4.2) and the resulting parameter constraints decrease the efficiency of the scheme. Although one would typically resort to implementing the scheme in the ring setting in order to improve efficiency in practice, Valovich's [Val16] LWE version does not seem to translate into a ring variant as the author remarks. Therefore, the practical performance of the scheme remains unclear.
From a formal point of view, his reduction from LWE also results in a slight loss in tightness with respect to worst-case lattice problems. In contrast, our scheme's security is based directly on the LWE-assumption and does not require increased parameters. Additionally, our construction translates into the ring setting in a straightforward way and our experimental results show that we can indeed benefit from this setting in terms of efficiency.

3.2 Related Work

In this section we review several areas of previous work that are relevant in evaluating our PSA scheme LaPS. We discuss how our work fits into the context of each field and highlight how we advance the state-of-the-art. Note that we will cover some of these topics in more technical detail later on. However, here we highlight particular lines of work that aim to solve

[2] Here, we refer to additive homomorphism in the sense that the addition of the function's outputs corresponds to the addition of the function's respective inputs.

a similar question as ours, focusing on the different solution concepts in comparison to our contributions.

PSA. After Shi *et al.*'s [Shi+11] introduction of PSA for sum aggregation in 2011 there have been a number of follow-up works extending the capabilities of the scheme. For instance, Jung *et al.* [Jun+13] generalize Shi *et al.*'s [Shi+11] scheme to evaluate a general *multi-variate polynomial function*. Note that this scheme is still based on the DDH-assumption and therefore has the same plaintext size limitations as in [Shi+11].
Chan *et al.* [CSS12] address the problem of user failures, i.e. *dynamic joins and leaves*. Their solution could actually be applied in order to extend ours, however this additional functionality comes at the cost of accuracy due to high accumulated aggregation error. Li and Cao [LC13] achieve better accuracy but incur a higher communication cost: they order the PSA users in an interleaved ring structure, where each user only adds a small amount of noise - consequently, when some users drop out, only a subset of the remaining ones have to update their keys. Our scheme, on the other hand, does not require a trade-off between accuracy and communication cost. Note that Li and Cao's scheme as well as its extension in order to compute the minimum [LCP14] are based on symmetric-key cryptography.

Local DP. The idea of local DP solutions is to protect the privacy of each sensitive data source in a distributed setting at the user's end *before* it is collected and evaluated. By applying privacy mechanisms locally, privacy is ensured in the DP sense. PSA schemes belong to solutions with that mission. However, they provide both privacy guarantees with respect to the aggregate output and security guarantees with respect to the individual user ciphertexts by combining encryption techniques with DP-mechanisms - different from other solutions that solely focus on the privacy part.
For instance, Erlingsson *et al.*'s [EPK14] RAPPOR system, which we briefly recall below, is an example of a deployed local DP solution but its security guarantees remain somewhat unclear: RAPPOR achieves the goal of privacy-preserving frequency estimation by applying a generalized version of the randomized response technique to each user's input before publishing it to the server. Concretely, the data that is collected is a set of attributes that correspond to e.g. the user's browser settings. These are represented as a binary vector, where each value is a predicate for a given attribute, and the user encodes it using Bloom filters before creating a randomized response, which is the user's DP output. After all users' responses are collected, the final result, i.e. frequencies of certain attributes, is calculated in a sophisticated decoding step based on hypotheses testing, least-squares solving and Lasso regression [Tib96].
RAPPOR is a completely user-based local DP construction and it does not require a trusted third party, which is different from our scheme as we assume a trusted setup (see Section 3.6.3). Besides the authors' in-depth analysis of DP, it is unclear what *security* guarantee is provided by the utilized Bloom filters. Although no attacks are known against the RAPPOR scheme, some attacks have been proposed against other Bloom-filter based constructions, such as a privacy-preserving record linkage system [Nie+14]

or a network protocol for packet forwarding, where the attack exploits information leaked from the utilized Bloom filters [AAS14].
Furthermore, Erlingsson *et al.*'s discussion of RAPPOR's attack model is based on the information leakage from the users' outputs and assesses its *privacy* implications [EPK14, Section 6]. In contrast, our construction eliminates this concern altogether as each user's output is *encrypted*. Therefore, we can rely on the security guarantees of our scheme, which we prove in our formal security analysis. In fact, our example instantiation allows to conjecture post-quantum security.
Finally, observe that RAPPOR is optimized for categorical input data, with the specific goal of frequency estimation. For instance the choice of randomized response over other privacy mechanisms was made with this application in mind as the authors state [EPK14]. It can be extended to numeric data by formulating the predicates as ranges up to the desired value. However, in order to achieve a fine-grained distinction, as it would be appropriate for the computation of the sum, a large expansion of the input vectors may become necessary. Our construction LaPS on the other hand, processes numeric input values in a straightforward manner as the plaintext for encryption.

Multi-party computation and fully homomorphic encryption. The general idea of multi-party computation (MPC) is to allow for the execution of computations over distributed data. Consequently, the involved parties learn the output of the computation without revealing any individual user's input. This compelling concept has become a thriving field of research with a long history [Yao82; Yao86; GMW87], see e.g. [Gol04, Chapter 7] for an overview. Observe that MPC-techniques have a very similar goal to PSA schemes, therefore it is natural to consider an application of MPC to the PSA setting. Here, we discuss qualitative and conceptual differences between some of the resulting MPC-based constructions and our LaPS scheme.
Danezis *et al.* [Dan+13] leverage secret sharing-based MPC techniques due to [Dam+06; Dam+12] in order to build a PSA scheme that allows for the computation of *non-linear aggregation functions*. They specifically design the scheme for the application to smart metering and assume a set of trusted authorities that collaboratively compute the desired function while ensuring the privacy of intermediate results. Therefore, each user distributes shares of her privacy-sensitive data to all authorities. The authorities then compute basic linear aggregation functions locally and finally combine their results in order to produce the output aggregate. However, the protocol requires interaction among the parties in order to compute a non-linear function, which with higher complexity increases the number of communication rounds. Our scheme is non-interactive and no additional set of trusted parties is required. On the other hand, we only support addition operations and we require a trusted setup.
MPC protocols generally do not account for a separate aggregating party. Therefore, it would require a tweak in order to apply them to PSA. Concretely, one may consider using *threshold fully homomorphic encryption*, e.g. Asharov *et al.*'s [Ash+12] TFHE scheme, which utilizes LWE-based encryption together with the fully homomorphic encryption schemes due to Brakerski *et al.* [BV11a; BGV12]: in this scheme all N users each possess a secret

and a public key share, which are combined into a common public encryption key after two key generation rounds. Subsequently, homomorphic operations can be executed on the generated ciphertexts. In the PSA setting, one would define a dedicated party as the aggregator, who would receive the combination, i.e. the sum, of the users' secret key shares, which then functions as his PSA decryption capability. Consequently, the aggregator could decrypt any ciphertext that corresponds to a function, e.g. the sum, of individual ciphertexts from any of the N users.
This also implies that the aggregator will be able to decrypt the sum of *any subset* of all N users' ciphertexts, including individual ones. This fact violates the core security notion of PSA schemes, which requires that the aggregator learns nothing but the final aggregate of *all* users' values. Furthermore, the TFHE scheme [Ash+12] is based on Regev's encryption scheme (see Section 2.3.1), which only allows for single-bit encryption. In contrast, our PSA scheme allows for a significantly larger plaintext space and provably satisfies the security requirements of PSA schemes.

3.3 Preliminaries

3.3.1 Differential Privacy

When surveying a group of people on potentially privacy-sensitive topics and assuming you ask yes/no questions, how do you make sure that the responses are indeed honest? If for instance the entire group responds "Yes" to the question "Have you smoked in the past?", the resulting statistic will inevitably leak information about the surveyed group although the individual responses were given anonymously. In fear of consequences, the respondents might lie.
Differential Privacy (DP) addresses exactly such scenarios where statistical queries are made on datasets containing privacy-sensitive data. The underlying idea is that useful information can be computed about a given population but no information is leaked about any particular participant. More formally, the output is considered DP iff one cannot tell whether or not a given individual was part of the surveyed population. Hence, by comparing two databases whose entries are equal except for one data record, so-called *adjacent* databases, a DP statistic will not reveal a significant difference in the output. This is achieved by applying some *privacy mechanism* $\mathcal{M}$ to the raw data in such a way that the published output fulfills the formal notion of (ϵ, δ)-DP, as defined next:

Definition 12 ((ϵ, δ)-DP[DR14]). *A randomized algorithm $\mathcal{M}$ with domain $\mathcal{D}^n$ and range $\mathcal{R}^k$ is (ϵ, δ)-DP if for all adjacent databases D_0, D_1 and for all $R \subseteq \mathcal{R}^k$:*

$$\Pr[\mathcal{M}(D_0) \in R] \leq \exp(\epsilon) \Pr[\mathcal{M}(D_1) \in R] + \delta,$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$, we say that $\mathcal{M}$ is ϵ-DP.

One of the aspects that distinguishes DP from other notions of privacy is its formal approach to quantifying a given level of privacy with concrete parameters, namely ϵ and δ. Intuitively, they give a ratio of likelihood: the

probability of retrieving a certain output from applying the privacy mechanism $\mathcal{M}$ on input D_0 versus getting the same result from applying $\mathcal{M}$ on D_1. This relation is called *privacy loss* and ϵ denotes its absolute upper bound with probability at least $1 - \delta$. Hence, typically small values are desired for ϵ and δ.
In general, a privacy mechanism produces a DP output by slightly distorting the input, e.g. by adding *noise* to the input. We present the arguably most common privacy mechanism, which utilizes *Laplace* distributed noise in order to guarantee DP. In what follows, f denotes the statistical query and Δf is its *sensitivity*, which expresses how much changing a single entry in a given database affects the outcome of the query.

Definition 13 (Laplace Mechanism ($\mathcal{M}_{Lap}$) [DR14]). *Let the Laplace distribution with scale σ and probability density function $Lap_\sigma(x) = \frac{1}{2\sigma} \exp(-|x|/\sigma)$ be denoted as Lap_σ. Given any function $f : \mathcal{D}^n \to \mathcal{R}^k$, the Laplace mechanism is defined as:*

$$\mathcal{M}_{Lap}(D, f(\cdot), \epsilon) = f(D) + (Y_1, \ldots, Y_k),$$

where Y_i are i.i.d. random variables drawn from Lap_σ with $\sigma = \Delta f/\epsilon$.

$\mathcal{M}_{Lap}$ is best suited for *numerical* queries, such as computing the sum of numerical data records, and indeed it guarantees ϵ-DP [DR14, Theorem 3.6].
Circling back to our example above, one may suggest to add noise to every given response and thereby preserve privacy of the statistical output as it will largely deviate from the count of actual responses. However, that sort of statistic is barely useful as the repeated addition of noise will lead to a significant distortion of the output. Therefore, *accuracy* is another important criterion in judging the quality and suitability of a given privacy mechanism. We restate the formal definition next.

Definition 14 ((α, β)-accuracy [UV11]). *For a query $f : \mathcal{D}^n \to \mathcal{R}^k$, the output of a mechanism $\mathcal{M}$ achieves (α, β)-accuracy if for all $D \in \mathcal{D}^n$:*

$$\Pr[|\mathcal{M}(D) - f(D)| \leq \alpha] \geq 1 - \beta.$$

The probability space is defined over the randomness of $\mathcal{M}$.

Similar to (ϵ, δ)-DP, the parameter α denotes an absolute upper bound on the maximum deviation of $\mathcal{M}$'s output from the true result with probability at least $1 - \beta$.

Local DP. In fact, for our survey problem it is not only crucial what privacy mechanism is applied but also *how* it is applied. If the surveyed responses are centrally collected and treated using some privacy mechanism before being published, it may be acceptable to add a single Laplace distributed sample to the output. For a sufficiently large number of participants, the result would only be slightly perturbed. However, if DP is desired for each individual response as it is collected in order to compute the final output, this alternative is not an option. This additional requirement has been termed *local* differential privacy and is especially relevant when privacy-sensitive data is collected from a *distributed* set of databases, i.e. several different users.
An appropriate technique for our example problem is *randomized response*,

which was specifically developed in order to protect the privacy of survey participants and it is one of the first known privacy mechanisms [War65]. Instead of answering the question directly, the respondent flips a coin and only answers truthfully if tails comes up. In case of heads, she flips another time and subsequently responds "Yes" for heads and "No" otherwise. The idea is to provide "*plausible deniability*" [DR14] of an answer to the participant through the probabilistic construction of the mechanism. At the same time, the true fraction of "Yes" answers can be estimated accurately with overwhelming probability.
An example of this mechanism is used in practice in Google's RAPPOR technology introduced by Erlingsson *et al.* [EPK14]. As we have discussed previously, it extends randomized response from binary answers to sets of categoric responses in order to estimate frequencies of certain users' Google Chrome settings in a privacy-preserving manner. In addition to providing local DP, Erlingsson *et al.* [EPK14] show efficiency in their experimental results that are collected from millions of users.

3.3.2 Aggregation With Untrusted Aggregator

DP is traditionally achieved by applying a privacy mechanism to some given sensitive data statistic before publishing it. However, for scenarios with multiple parties, where each wishes to protect her privacy as data is collected, this process is undesirable.
Private Stream Aggregation (PSA) aims to provide a mechanism for aggregation, e.g. computing the sum of sensitive data records, where the individual users or *participants* do *not trust* the aggregator. We highlight the conceptual difference between the standard privacy model with a trusted aggregator and the PSA approach with an untrusted aggregator in Figure 3.1.
In a PSA protocol each participant U_i applies a privacy mechanism to their sensitive data D_i, resulting in a noisy version of their data X_i, before encrypting it and providing the ciphertext to the aggregator A. Subsequently, A computes the desired output by aggregating the ciphertexts and decrypting the result. This result is inherently DP and can be directly published.

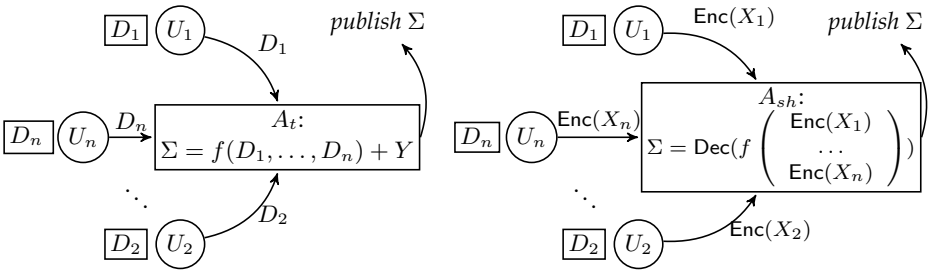

FIGURE 3.1: Comparison of privacy models
Left: Standard DP-model, *right*: PSA model [BGZ18].

Shi *et al.* [Shi+11] introduced the notion of PSA schemes in 2011. We restate their formal definition of a PSA scheme next.

Definition 15 (PSA Scheme [Shi+11]). *Let $[n] := \{1, 2, \ldots, n\}$ be the set of users participating in the aggregation each holding values from some domain $\mathcal{D}$. Let $f : \mathcal{D}^n \to \mathcal{R}$ be an aggregation function with some range $\mathcal{R}$. Let $\chi : \mathcal{D} \times \Omega \to \mathcal{D}$ denote some randomization function that adds the two input values from $\mathcal{D}$ and*

Ω, *where* Ω *is some sample space of the randomization noise. Let* T *be the set of time steps used throughout execution. A PSA scheme consists of the following PPT-algorithms:*

- $(\mathsf{param}, \{sk_i\}, sk_A) \leftarrow \mathsf{Setup}(1^\kappa)$*: Takes in a security parameter* κ *and outputs public parameters* param*, a private key* sk_i *for each participant, as well as an aggregator capability* sk_A *needed for decryption of aggregate statistics in each time step* $t \in T$*. Each participant* i *obtains the private key* sk_i *and the data aggregator obtains the capability* sk_A*.*

- $c_{i,t} \leftarrow \mathsf{NoisyEnc}_i(\mathsf{param}, sk_i, t, d, r)$*: During time step* t*, each participant calls the* $\mathsf{NoisyEnc}_i$ *algorithm to encode its data* d *with noise* r*. The result is a noisy encryption* c *of* d *randomized with the noise* r*.*

- $f(\mathbf{x}) \leftarrow \mathsf{AggrDec}(\mathsf{param}, sk_A, t, c_{1,t}, c_{2,t}, \ldots, c_{n,t})$*: The decryption algorithm takes in the public parameters* param*, a capability* sk_A*, and ciphertexts* $c_{1,t}, c_{2,t}, \ldots, c_{n,t}$ *for the same time step* t*. For each* $i \in [n]$*, let* $c_{i,t} = \mathsf{NoisyEnc}_i(sk_i, t, x_i)$*, where each* $x_i := \chi(d_i, r_i)$ *for some randomization function* χ*. Let* $\mathbf{d} := (d_1, \ldots, d_n)$ *and* $\mathbf{x} = (x_1, \ldots, x_n)$*. The decryption algorithm outputs* $f(\mathbf{x})$ *which is a noisy version of the targeted statistics* $f(\mathbf{d})$*.*

Note that we sometimes abuse notation and only require the user to input her raw value d to NoisyEnc (as opposed to both d and r as in Definition 15) when noise r is generated within the routine. Furthermore, we simplify notation by omitting the time-step t in the subscripts.

3.3.2.1 Aggregator Obliviousness

The guarantee of PSA schemes is centered around restricting the knowledge gain of the aggregator to a minimum, i.e. the aggregator does not learn *anything* but the noisy output. The fact that each participant only transmits a noisy version of their data in the first place, ensures that the output does not leak information about any individual participant's data in the DP sense. The other crucial component is encryption: each participant has an individual encryption key sk_i. The aggregator's counterpart, i.e. A's *decryption capability* consisting of his decryption key sk_A, allows A to only decrypt the final result and none of the individual users' ciphertexts on their own. In other words, A has to aggregate the input ciphertexts in order to retrieve the result.
Shi *et al.* [Shi+11] capture this guarantee with the notion of *aggregator obliviousness*, which we formally state in Definition 16. Note that while providing DP of the aggregate output as well as the individual inputs is a goal of this construction, the privacy guarantee comes from the deployed privacy mechanism. Therefore, aggregator obliviousness is a security notion[3]. Observe that PSA schemes satisfy the notion of local DP (see Section 3.3.1). However, that particular terminology is rather recent - Shi *et al.* refer to "*distributed differential privacy*" [Shi+11].

Definition 16 (Aggregator Obliviousness [Shi+11])**.** *A PSA scheme is aggregator oblivious if no PPT adversary has more than negligible advantage in* κ *in winning the following security game:*

[3]We will address both security and privacy when instantiating our PSA scheme but we refer back to Definition 12 for the privacy guarantee based on the utilized privacy mechanism.

Setup. *Challenger runs the* Setup *algorithm, returns the public parameters* param *to the adversary.*

Queries. *The adversary makes the following types of queries adaptively.*

- ***Encrypt.*** *The adversary may specify* (i, d, r) *and ask for the ciphertext. The challenger returns the ciphertext* $\mathsf{NoisyEnc}_i(\mathsf{param}, sk_i, t, d, r)$ *to the adversary.*
- ***Compromise.*** *The adversary specifies an integer* $i \in \{0, \ldots, n\}$*. If* $i = 0$*, the challenger returns the aggregator's decryption key* sk_A *to the adversary. If* $i \neq 0$*, the challenger returns* sk_i*, the secret key of the* i^{th} *participant, to the adversary.*
- ***Challenge.*** *This query can only be made once throughout the game. The adversary specifies a set of participants* U *and a time* t*, such that* $i \in U$ *has not been previously compromised.*
  For each user $i \in U$ *the adversary chooses two plaintext-noise pairs* (d_i, r_i) *and* (d'_i, r'_i) *and sends them to the challenger. The challenger flips a random bit* b*. If* $b = 0$*, the challenger computes* $\forall i \in U : c_i = \mathsf{NoisyEnc}_i(\mathsf{param}, sk_i, t, d_i, r_i)$*. If* $b = 1$*,* $\forall i \in U : c_i = \mathsf{NoisyEnc}_i(\mathsf{param}, sk_i, t, d'_i, r'_i)$ *and returns* $\{c_i\}$ *to the adversary.*

Guess. *The adversary guesses, whether* b *is 0 or 1.*

We say that the adversary wins the game if she correctly guesses b *and if she compromised the aggregator (i.e. possesses the decryption key* sk_A*,) then* $\sum_{i \in U} d_i + r_i = \sum_{i \in U} d'_i + r'_i$ *must hold.*
Note that if the aggregator colludes with a subset of the participants or is leaked some of the plaintexts[4]*, then he can inevitably learn the sum of the remaining participants' values. We require that in this case the aggregator learns no additional information about these participants' data. However, this requirement is achieved by the privacy guarantees of the scheme.*

3.3.2.2 Aggregator Unforgeability

Aggregator obliviousness denotes the basic security requirement for PSA schemes and essentially ensures that the aggregator cannot retrieve any result but the noisy aggregate of the users' inputs. However it does not protect from the aggregator manipulating that result before publishing it. Therefore, one may additionally desire *public verifiability* of the aggregator's output. This notion is referred to as *aggregator unforgeability*.
The formal definition was first introduced by Leontiadis *et al.* [Leo+15] and extends aggregator obliviousness with public verifiability. In other words, this extended version of the scheme weakens the assumption of an honest-but-curious aggregator to a dishonest aggregator and therefore provides stronger guarantees. Note that depending on the adversarial model, i.e. the assumed capabilities of the adversary, the definitions of aggregator unforgeability range from *strong* [Leo+15] to *weak* [Emu17] unforgeability. We restate strong aggregator unforgeability according to Leontiadis *et al.* [Leo+15] in Definition 17.
Concretely, each user generates a *tag* that she transmits together with her

[4] This applies analogously to the adversary who compromises all of the secret keys (including the aggregator's) but one.

ciphertext to the aggregator. The aggregator then retrieves the aggregate result and computes a *proof of correctness* by aggregating the provided tags. The verification routine VerifySum checks the validity of the aggregate output, together with the proof of correctness σ and a public verification key vk. A given PSA construction ensures aggregator unforgeability iff the aggregator can only produce a *valid* proof when using all of the provided user tags and when aggregating correctly, i.e. according to the protocol. In what follows, we denote the aggregate output by X_t and $x_{i,t}$ denotes the result of randomizing data d with the noise r following notation in [Emu17] and all other parameters are as in Definition 15.

Definition 17 (Aggregator Unforgeability [Leo+15])**.** *For any PPT adversary A and a security parameter $\lambda \in \mathbb{N}$, we define the experiment $\mathsf{Exp}_A^{AU}(\lambda)$ as follows.*

- $(\mathsf{param}, sk_A, \{sk_i\}_{i=1}^n, vk) \leftarrow \mathsf{Setup}(1^\lambda)$
- $(t^*, X_{t^*}, \sigma_{t^*}) \leftarrow A^{\mathcal{O}_{\mathsf{enc}}}(\mathsf{param}, sk_A, vk)$
- *If one of the following holds, then return* 1 *and* 0 *otherwise*

(Type I): $\mathsf{VerifySum}(\mathsf{param}, t^*, X_{t^*}, \sigma_{t^*}, vk_{t^*}) = 1$
$\wedge$ *No encryption oracle is called at* t^*

(Type II): $\mathsf{VerifySum}(\mathsf{param}, t^*, X_{t^*}, \sigma_{t^*}, vk_{t^*}) = 1$
$\wedge X_{t^*} \neq \sum_{i=1}^n x_{i,t^*} \mod M$

The encryption oracle $\mathcal{O}_{\mathsf{enc}}$ takes a tuple $(i, t, d_{i,t}, r_{i,t})$ as the input, and returns $(c_{i,t}, \sigma_{i,t}) \leftarrow \mathsf{NoisyEnc}(\mathsf{param}, t, d_{i,t}, r_{i,t}, sk_i)$.
We say that a PSA scheme is aggregator unforgeable if the advantage $\mathsf{Adv}_A^{AU}(\lambda) := \Pr[\mathsf{Exp}_A^{AU}(\lambda) = 1]$ is negligible in λ for any PPT adversary A.

Note that *public* verifiability refers to the fact that the verification can be executed by anyone, i.e. the verification routine and the verification key are public.

3.3.3 Generalized A-LWE and Gaussian Distribution

We generalize the A-LWE problem from the original definition [EDB15] as described previously, where *any* function f embeds the message m such that $\mathbf{v} = f(m)$ as long as output $\mathbf{v}$ is indistinguishable from random. Hence, whenever we refer to A-LWE in the context of our PSA scheme, we refer to our generalized version as defined below. Therefore, we may abuse notation by reusing symbols from Definition 9. Note that in the following we refer to the Gaussian parameter αq as σ for better readability. We present the resulting definition next.

Definition 18 (Generalized A-LWE Distribution (adapted from [EDB15]))**.** *Let κ, λ, q, l, x be integers, where $l = \lceil \log q \rceil$ and $\lambda = x \cdot l$. Let f be some function where the output is indistinguishable from random. Let $\mathbf{g}^T = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_q^l$ and $\mathbf{G} = \mathbf{I}_{\lambda/l} \otimes \mathbf{g}^T \in \mathbb{Z}_q^{\lambda/l \times \lambda}$. For $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^\kappa$ and $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{\kappa \times \lambda}$, define the A-LWE distribution $L_{\kappa,\lambda,\sigma}^{\mathsf{A\text{-}LWE}}(m)$ with $m \in \mathbb{Z}_q$ to be the distribution over $\mathbb{Z}_q^{\kappa \times \lambda} \times \mathbb{Z}_q^\lambda$ obtained as follows:*

- *Set $\mathbf{v} = f(m) \in \mathbb{Z}_q^{\lambda/l}$.*
- *Sample $\mathbf{e} \leftarrow D_{\Lambda_{\mathbf{v}}^\perp(\mathbf{G}),\sigma} \in \mathbb{Z}_q^\lambda$.*

- *Return* $(\mathbf{A}, \mathbf{b}^T)$ *where* $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$.

Note that we only make use of the decision variant of the A-LWE problem here. The search variants can be defined analogously to Definition 9.

Definition 19 (Generalized decision A-LWE (adapted from [EDB15])). *Let* κ, λ, q *be integers. Let* f *be some function with pseudo-random output. The* decision A-LWE$^f_{\kappa,\lambda,\sigma}$ *problem asks to distinguish in polynomial time (in* κ*) between samples* $(\mathbf{A}_i, \mathbf{b}_i^T) \leftarrow L^{\text{A-LWE}}_{\kappa,\lambda,\sigma}(m)$ *and uniform random samples from* $\mathbb{Z}_q^{\kappa\times\lambda} \times \mathbb{Z}_q^{\lambda}$ *for a secret* $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{\kappa}$ *and some* $m \in \mathbb{Z}_q$.
We say that decision A-LWE$^f_{\kappa,\lambda,\sigma}$ *is hard if all polynomial time algorithms solve the* decision A-LWE$^f_{\kappa,\lambda,\sigma}$ *problem only with negligible probability.*

In order to prove security of the resulting ciphertext we will use Lemma 13, which states that the A-LWE error term distribution is indistinguishable from the discrete Gaussian distribution under certain conditions. And in order to show that our construction fulfills that premise, we will use the bound provided by Lemma 12. Note that we restate a simplified version as formulated in [EDB14, Lemma 3].

Lemma 12 (Bound on smoothing parameter [GPV08; EDB14]). *Let* $\Lambda \subset \mathbb{R}^n$ *be a lattice with basis* $\mathbf{B} = b \cdot \mathbf{I}$ *and let* $\epsilon > 0$. *Then the smoothing parameter* $\eta_\epsilon(\Lambda)$ *is upper bounded as follows:* $\eta_\epsilon(\Lambda) \leq b \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$.

Lemma 13 ([EDB15]). *Let* $\mathbf{M} \in \mathbb{Z}_q^{a\times b}$ *be an arbitrary full-rank matrix. If the distribution of* $\mathbf{v} \in \mathbb{Z}_q^a$ *is computationally indistinguishable from the uniform distribution over* $\mathbb{Z}_q^a$, *then* $D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{M}),r}$ *is computationally indistinguishable from* $D_{\mathbb{Z}^b,r}$ *for* $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{M}))$.

3.4 Shi et al.'s PSA Scheme

We detail the first proposed PSA scheme for *sum* aggregation from Shi *et al.* [Shi+11] as it created the basis for this line of research and will be an important guideline in evaluating our PSA scheme. Note that the privacy mechanism in their scheme uses *Geometric* noise, which is essentially equivalent to a discrete version of the Laplace mechanism (see Definition 13).

Definition 20 (Shi *et al.*'s PSA Scheme [Shi+11]). *Let* $\mathbb{G}$ *be a cyclic group of prime order* p, *where Decisional Diffie-Hellman is hard. Let* $H : \mathbb{Z} \to \mathbb{G}$ *be a hash function modeled as a random oracle. Let* $\mathsf{Geom}(\sigma)$ *denote the symmetric geometric distribution over integer values with the probability mass function at* k: $\frac{\sigma-1}{\sigma+1} \cdot \sigma^{-|k|}$, *where* $\sigma > 1$. *Let* $\beta \leq 1$.

- $(\mathsf{param}, \{sk_i\}, sk_A) \leftarrow \mathsf{Setup}(1^\kappa)$: *A trusted dealer chooses a random generator* $g \in \mathbb{G}$, *and* $n+1$ *random secrets* $s_0, s_1, \ldots, s_n \in \mathbb{Z}_p$ *such that* $s_0 + s_1 + s_2 + \ldots + s_n = 0$. *The public parameters* $\mathsf{param} := g$. *The aggregator obtains the capability* $sk_A := s_0$, *and participant* U_i *obtains the secret key* $sk_i := s_i$.

- $c_{i,t} \leftarrow \mathsf{NoisyEnc}_i(\mathsf{param}, sk_i, t, d_i)$: *Each participant* i *takes her data* d_i *and adds some noise* r_i *to it, s.t.* $x_i = d_i + r_i \mod p \in \mathbb{Z}_p$, *where* r_i *is sampled as follows:*

$$r_i = \begin{cases} 0 & \text{with probability } 1-\beta \\ \mathsf{Geom}(\sigma) & \text{with probability } \beta. \end{cases}$$

She computes the following ciphertext:

$$c_{i,t} \leftarrow g^{x} \cdot H(t)^{sk_i}.$$

- $V \leftarrow \mathsf{AggrDec}(\mathsf{param}, sk_A, t, c_{1,t}, c_{2,t}, \ldots, c_{n,t})$*: Compute*

$$V \leftarrow H(t)^{sk_A} \prod_{i=1}^{n} c_{i,t}.$$

Note that Shi *et al.* [Shi+11] assume a *trusted Setup* in order to securely distribute the generated secret keys. As the authors point out, this can be achieved in practice by a trusted third party or by leveraging standard secure Multi-Party Computation techniques.

Security. Each user's noisy value x_i is encrypted by placing it in the exponent in a discrete log-style, where security is based on the hardness of the Decisional Diffie-Hellman problem as stated in Lemma 14.

Lemma 14 ([Shi+11, Theorem 1])**.** *Assuming that the Decisional Diffie-Hellman problem is hard in group $\mathbb{G}$ and that the hash function H is a random oracle, then the PSA scheme according to Definition 20 satisfies aggregator obliviousness security in the "encrypt-once" model.*

Note that Shi *et al.*'s [Shi+11] PSA scheme is set up in a time-step based manner, i.e. the users aggregate values during a certain time-step t and repeat the process if they want to share more values later on. The encryption's security depends on the current time t, since it serves as fresh randomness. The ciphertext could be compromised if a user provides multiple encryptions under the same parameters, specifically during the same time-step t. Therefore, Shi *et al.* [Shi+11] restrict their security statement to the *encrypt-once* model, which ensures that encryption only takes place once per user and time step.

Correctness. The fact that the product of the ciphertexts c_i corresponds to the sum of the values x_i in the exponent ensures correctness. Observe that the sum of the users' secret encryption keys and the aggregator's decryption key is 0, which cancels out the remaining factor $\prod_{i=1}^{n} H(t)^{sk_A + \sum_{i=1}^{n} sk_i}$ in V during decryption (see AggrDec in Definition 20).
Eventually, the aggregator is left with $g^{\sum_{i=1}^{n} x_i}$, which he decrypts by solving the discrete logarithm of V base g. Shi *et al.* [Shi+11] suggest a brute-force approach or Pollard's lambda method (see e.g. [MOV96]). Note that for this to be possible, the domain of the input values, i.e. the plaintext space, has to be significantly restricted. The authors' [Shi+11] practicality analysis in fact assumes 1-bit participant inputs.

Privacy. Finally, privacy is preserved by incorporating Geometric noise into each user's input. The probability at which each user indeed adds noise depends mainly on the fraction of assumed uncompromised participants γ:

Lemma 15 ([Shi+11, Lemma 1]). *Let $\epsilon > 0$, $0 < \delta < 1$ and $\sigma = \exp(\epsilon/\Delta)$, where Δ is the width of participant data in $\mathbb{Z}_p$. Suppose at least γ fraction of participants are uncompromised. Then, the aggregate output generated by the PSA scheme according to Definition 20 is (ϵ, δ)-DP for $\beta = \min\{\frac{1}{\gamma n} \log \frac{1}{\delta}, 1\}$.*

3.5 General LaPS Scheme

The core idea of LaPS that makes it a true PSA scheme that satisfies aggregator obliviousness, is the way we encrypt the user's plaintext into an additively homomorphic ciphertext and essentially wrap this ciphertext into an A-LWE-term. The latter is a ciphertext in itself and serves as the final output. Consequently, the aggregator's decryption capability consists of two parts. The two aggregator decryption keys can be understood as the tools to unwrap the encryption layers of the aggregate ciphertext in reverse order: using the first key the inner ciphertext is recovered from the A-LWE term and with the second key the plaintext is revealed. Observe that when the aggregator receives all user ciphertexts, the first key is designed in such a way that it can only decrypt the sum of all ciphertexts - specifically no partial sums nor individual ciphertexts.
It is essential that this process can only be executed in this order. Only the sum of all ciphertexts can be decrypted with the first key. Due to the summation the contained additively homomorphic ciphertexts - and therefore the underlying plaintexts - are summed as well. Therefore, the aggregator can only recover the sum of the plaintexts, i.e. the noisy sum of the user values.
On a technical level, a user ciphertext is of the form $(\mathbf{A}, \mathbf{b}^T)$, where $\mathbf{A}$ is a public parameter and $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$. $\mathbf{s}$ is the secret key and $\mathbf{e}$ is the error term that is sampled from distribution $D_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}),r}$, where $\mathbf{v}$ encodes the plaintext message m. Using the public gadget matrix $\mathbf{G}$ one can recover the message via $\mathbf{Ge} \equiv \mathbf{v} \mod q$.
Based on our generalized version of the A-LWE problem (see Definition 19) the encoding step from m to $\mathbf{v}$ can be instantiated with any additively homomorphic function that produces a pseudo-random output. Therefore, we view this step as the encryption routine in the cryptographic scheme $\mathsf{AHOM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. We require it to be additively homomorphic in order to ensure correctness of our PSA scheme and it has to produce pseudo-random ciphertexts such that $\mathbf{v}$ is indeed pseudo-random, as detailed in Definition 21. Note that here we require additive homomorphism in the sense that the *sum* of the ciphertexts corresponds to the sum of the plaintexts, e.g. the discrete log-style encryption as in [Shi+11] could not be plugged in here.
Altogether, each user i with input d_i adds privacy-preserving noise r_i creating a noisy version of her data $x_i = d_i + r_i \mod q$. Then she encrypts x_i with $\mathsf{AHOM.Enc}$ using the public key pk from $\mathsf{AHOM.Gen}$. The resulting "internal" ciphertext is $\mathbf{v}_i$ and is additively homomorphic. $\mathbf{v}_i$ is then utilized to sample the error $\mathbf{e}_i$ in the A-LWE term $\mathbf{c}_{i,\mathbf{A}} = \mathbf{s}_i^T\mathbf{A} + \mathbf{e}_i^T$, which corresponds to the final user ciphertext and output.
Note that $\mathbf{s}_i$ is an individual encryption key that is unique to each user and kept secret. When summing up all users' ciphertexts, the aggregate ciphertext will be an encryption of the sum of the plaintexts under the sum of the

secret keys. Therefore, in order to ensure that the aggregator can decrypt (only) the sum of the ciphertexts, his decryption key sk_{A_1} corresponds to the negative sum of the users' secret keys. By computing $sk_{A_1}^T \mathbf{A}$ and adding it to the aggregate ciphertext, the aggregator retrieves the sum of the error terms $\sum_{i=1}^{N} \mathbf{e}_i$. Using the relation between the gadget matrix $\mathbf{G}$ and $\mathbf{v}_i$ due to the construction of the error distribution, he retrieves the sum of the $\mathbf{v}_i$'s, i.e. by multiplying $\mathbf{G}$ with $\sum_{i=1}^{N} \mathbf{e}_i$. Finally, the sum of these inner ciphertexts is decrypted by invoking AHOM.Dec with sk_{A_2}, which corresponds to the secret key generated from AHOM.Gen. It is the second part of the aggregator's decryption capability.
Observe that in our case the decryption of the A-LWE term is simplified to a single addition, which essentially eliminates the $\mathbf{s}^T \mathbf{A}$ summand in the term. El Bansarkhani *et al.*'s [EDB15] definition of their A-LWE-based generic encryption scheme requires a trapdoor construction in order to recover both the secret and the error term. This step is the most computationally intensive operation of the scheme and largely determines the overall efficiency as the authors note [EDB15]. Our construction minimizes the effort to a single addition due to the definition of the aggregator's secret key.
In what follows, we first present the formal definition of our general PSA scheme LaPS before analyzing correctness, and security and privacy.

Algorithm 1 Algorithm Sample to sample from $\Lambda^{\perp}(\mathbf{g}^T)$ [EDB15]

Require: $\mathbf{g}^T \in \mathbb{Z}_q^l, w \in \mathbb{Z}_q, r$
Ensure: $\mathbf{t} = (t_0, \ldots, t_{l-1})^T \in \Lambda_w^{\perp}(\mathbf{g}^T)$ distributed according to $D_{\Lambda_w^{\perp}(\mathbf{g}^T),r}$
  $a_0 := w$
  for $j = 0, \ldots, l-1$ **do**
    $t_j \leftarrow D_{2\mathbb{Z}+a_j,r}$
    $a_{j+1} = (a_j - t_j)/2$
  end for

Definition 21 (Lattice-based PSA (LaPS))**.** *Let κ be a security parameter, $N \in \mathbb{N}$ the number of participants and let $\beta \in (0, 1]$. Let χ be a discrete noise distribution. Let* AHOM $=$ (Gen, Enc, Dec) *be an asymmetric encryption scheme with pseudo-random ciphertexts that is additively homomorphic, such that*

$$\mathsf{AHOM.Dec}(sk, \sum_{i=1}^{N} \mathsf{AHOM.Enc}(pk, m_i)) = \mathsf{AHOM.Dec}(sk, \mathsf{AHOM.Enc}(pk, \sum_{i=1}^{N} m_i)).$$

A Lattice-based PSA scheme $LaPS =$ (Setup, NoisyEnc, AggrDec) consists of the following PPT-algorithms:

- $(\{\mathbf{A}, \mathbf{g}^T, pk\}, \{\mathbf{s}_i\}, (sk_{A_1}, sk_{A_2})) \leftarrow \mathsf{Setup}(1^{\kappa})$*: Generate the public parameters $\mathbf{A}$, $\mathbf{g}^T$ and pk as follows and distribute them to all parties.*
  - *Draw $\mathbf{A}$ uniformly at random from $\mathbb{Z}_q^{\kappa \times \lambda}$, where $l = \lceil \log q \rceil$ and $\lambda = x \cdot l$ for some positive integer x.*
  - *Set vector $\mathbf{g}^T = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_q^l$.*
  - *Generate $(pk, sk) \leftarrow$ AHOM.Gen and extract public key $pk \in (pk, sk)$.*

  For all $i \in \{1, \ldots, N\}$ draw $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^{\kappa}$ and send it to user i as her secret key. The aggregator's secret decryption key is the tuple (sk_{A_1}, sk_{A_2}), where

  - $sk_{A_1} = -\sum_{i=1}^{N} \mathbf{s}_i$ *and*
  - $sk_{A_2} = sk \in (pk, sk) \leftarrow \mathsf{AHOM.Gen}$.

- $\mathbf{c}_{i,\mathbf{A}} \leftarrow \mathsf{NoisyEnc}_i(\{\mathbf{A}, \mathbf{g}^T, pk\}, \mathbf{s}_i, d_i)$*: Each user* i *takes her data* $d_{i,\mathbf{A}} \in \mathcal{D}$ *and adds some noise* r_i *to it, such that* $x_i = d_i + r_i \mod q \in \mathbb{Z}_q$.

  - r_i *is sampled as follows:*
    $r_i = \begin{cases} 0 & \text{with probability } 1-\beta \\ Y & \text{with probability } \beta \end{cases}$, *where* $Y \leftarrow \chi$.
  - *Compute* $\mathbf{v}_i = \mathsf{AHOM.Enc}(pk, x_i) \in \mathbb{Z}_q^{\lambda/l}$.
  - *Invoke Algorithm 1 for each component of* $\mathbf{v}_i$*:*
    $\mathbf{e}_i = (\mathsf{Sample}(\mathbf{g}^T, v_{i_1}, \sigma) \ , \ldots, \ \mathsf{Sample}(\mathbf{g}^T, v_{i_{\lambda/l}}, \sigma))$. *Hence,* $\mathbf{e}_i \leftarrow D_{\Lambda^{\perp}_{\mathbf{v}_i}(\mathbf{G}),\sigma} \in \mathbb{Z}_q^{\lambda}$.

  Output the ciphertext $\mathbf{c}_{i,\mathbf{A}} = \mathbf{s}_i^T \mathbf{A} + \mathbf{e}_i^T \in \mathbb{Z}_q^{\lambda}$.

- $\sum_{i=1}^{N} x_i \leftarrow \mathsf{AggrDec}(\{\mathbf{A}, \mathbf{g}^T\}, (sk_{A_1}, sk_{A_2}), \{\mathbf{c}_{1,\mathbf{A}}, \ldots, \mathbf{c}_{N,\mathbf{A}}\})$*: Receiving the users' ciphertexts* $\{\mathbf{c}_i\}$ *the aggregator computes* $\mathbf{c} = \sum_{i=1}^{N} \mathbf{c}_{i,\mathbf{A}}$.

  - *Compute* $\mathbf{e} = \sum_{i=1}^{N} \mathbf{e}_i^T = \mathbf{c} + sk_{A_1}^T \mathbf{A}$.

  The aggregator retrieves the noisy sum of the users' values via

$$\sum_{i=1}^{N} x_i = \mathsf{AHOM.Dec}(sk_{A_2}, \mathbf{G} \cdot \mathbf{e} \mod q),$$

  where $\mathbf{G} = \mathbf{I}_{\lambda/l} \otimes \mathbf{g}^T \in \mathbb{Z}_q^{\lambda/l \times \lambda}$.

Observe that Shi *et al.*'s [Shi+11] initial notion of PSA schemes provides for a time-step based aggregation (see Section 3.4). Each time-step t conveniently also serves as fresh randomness in their scheme, hence it actually has to be re-sampled for each encryption resulting in the encrypt-once assumption of their security statement (Lemma 14). In contrast, our scheme does not require such an assumption: the public matrix $\mathbf{A}$ can be re-used across encryptions. In the case where a notion of identifiable time-steps is desired, one may adapt the scheme in a straightforward way by sampling a set of matrices $(\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{\kappa \times \lambda}, \ldots, \mathbf{A}_t \xleftarrow{\$} \mathbb{Z}_q^{\kappa \times \lambda})$ during Setup with t being the corresponding time-step. From an implementation perspective, this can be done in a memory-conserving way by solely storing a seed that is used to generate the matrix $\mathbf{A}_i$ on-the-fly. Note that all security notions can be extended, accordingly.
In the following we state correctness, and security and privacy of our general scheme LaPS. Observe that we make minimal assumptions with respect to the individual building blocks. For instance, security requires semantic security with pseudo-random ciphertexts of the embedded additively homomorphic encryption scheme AHOM. (ϵ, δ)-DP of the aggregate output is based on ϵ-DP of the used privacy mechanism.

Correctness of LaPS. AHOM is required to be additively homomorphic. Therefore, $\mathsf{AHOM.Dec}(\sum_{i=1}^{N} \mathbf{v}_i) = \sum_{i=1}^{N} x_i$. Finally, due to $\mathbf{G} \cdot \mathbf{e}_i \mod q = \mathbf{v}_i$, AggrDec indeed correctly computes

$$\mathsf{AHOM.Dec}(sk_{A_2}, \mathbf{G} \cdot \mathbf{e} \mod q = \mathbf{v}_i) = \sum_{i=1}^{N} x_i.$$

3.6 Security and Privacy of LaPS

3.6.1 Security of LaPS

We show security of our LaPS scheme by first showing semantic security of each user ciphertext that is produced by NoisyEnc (Theorem 1). We then proceed to showing aggregator obliviousness security (Theorem 2).

Theorem 1 (Semantic Security)**.** *Let the output of* AHOM.Enc *be indistinguishable from random. Then, the ciphertexts generated by* NoisyEnc *in LaPS according to Definition 21 are semantically secure for* $\sigma \geq 2\sqrt{\kappa} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$ *assuming the hardness of worst-case lattice problems*[5]*.*

Proof. First, note that due to the above assumption, $\mathbf{v}_i$ is indistinguishable from random. Furthermore, the smoothing parameter $\eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$ can be bounded from above using Lemma 12, resulting in $\eta_\epsilon(\Lambda_q^\perp(\mathbf{G})) \leq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$. Therefore, by construction $\sigma \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$ and Lemma 13 can be applied to matrix $\mathbf{G}$. Then, $D_{\Lambda_{\mathbf{v}_i}^\perp(\mathbf{G}),\sigma}$ correctly simulates the discrete Gaussian distribution $D_{\mathbb{Z}^\lambda,\sigma}$ and the ciphertexts $\mathbf{c}_i$ represent plain A-LWE$_{\kappa,\lambda,\sigma}$ samples. Note that in the remainder of this section we assume that σ and κ are set such that Lemma 13 applies and avoid restating the corresponding parameter restriction for better readability.
Therefore, the statement follows immediately from the hardness of decision A-LWE$^f_{\kappa,\lambda,\sigma}$, where $f :=$ AHOM.Enc: A-LWE samples are indistinguishable from LWE$_{\kappa,\lambda,\sigma}$ samples due to [EDB15]. Finally, LWE samples are indistinguishable from uniform samples based on the hardness of worst-case lattice problems due to [Reg05]. □

Aggregator obliviousness entails that the aggregator learns nothing but the noisy sum of all participants' values. We show that our general framework and therefore any instantiation that fulfills the stated requirements of Theorem 2 provides aggregator obliviousness.

Theorem 2 (Aggregator Obliviousness Security)**.** *Let the output of* AHOM.Enc *be indistinguishable from random and let* $\sigma \geq 2\sqrt{\kappa}$*. LaPS according to Definition 21 satisfies aggregator obliviousness security assuming the hardness of worst-case lattice problems.*

Proof. Note that this property (together with Theorem 1) targets the security of the PSA scheme as opposed to its privacy. It is independent of the used randomization procedure that adds noise to the users' values. We therefore assume that a potential adversary can choose the noise r_i as part

[5]Note that for better readability, whenever we refer to "the hardness of worst-case lattice problems" we mean the hardness of the lattice problems GapSVP and SIVP with parameters as in Lemma 2, where the lattice dimension corresponds to security parameter κ.

of the **Challenge** phase in the respective security game as specified in Definition 16. Concretely, we adopt the notation $\mathsf{NoisyEnc}_i(pk, \mathbf{g}^T, \mathbf{s}_i, \mathbf{A}, d_i, r_i)$ to set $x_i = d_i + r_i \mod q$ and encrypt x_i. This is in line with previous work, such as Shi *et al.*'s PSA scheme, which is proven to be aggregator oblivious independent of the used randomization procedure [Shi+11].
Using Theorem 1 it suffices to show that if there exists a PPT adversary $\mathcal{A}$ that wins the aggregator obliviousness security game, then there exists a PPT adversary $\mathcal{B}$ that can solve the decision $\mathsf{A\text{-}LWE}^f_{\kappa,\lambda,\sigma}$, i.e. distinguish an $\mathsf{A\text{-}LWE}_{\kappa,\lambda,\sigma}$ sample from a uniformly random sample over $\mathbb{Z}_q^{\kappa\times\lambda} \times \mathbb{Z}_q^{\lambda}$. Note that f is defined as $\mathsf{AHOM.Enc}$, where AHOM consists of the routines $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as in Definition 21.
We define the following intermediate game $\mathsf{Game1}$ similar to [Shi+11] that is indistinguishable from the aggregator obliviousness security game according to Definition 16:

- First, we treat any **Encrypt** query as a **Compromise** query from the adversary. Clearly, this makes the adversary more powerful as she obtains the secret key $\mathbf{s}_i$ and she can compute the ciphertext herself.
- Secondly, we change the **Challenge** phase to its real-or-random version, i.e. instead of having the adversary specify two sets of plaintext-randomness pairs $\{(d_i, r_i)\}$ and $\{(d_i', r_i')\}$ and have her distinguish between encryptions of either one, we let the adversary pick one set $\{(d_i, r_i)\}$ and have her distinguish between a set of valid encryptions and a set of random values in $\mathbb{Z}_q^{\lambda}$.

It is straightforward that any adversary with more than negligible advantage in winning $\mathsf{Game1}$ will also win the aggregator obliviousness security game with more than negligible advantage. Therefore, it suffices to show that with a PPT adversary $\mathcal{A}$ with more than negligible advantage in winning $\mathsf{Game1}$ we can construct an algorithm $\mathcal{B}$ that solves decision $\mathsf{A\text{-}LWE}^f_{\kappa,\lambda,\sigma}$ with more than negligible advantage.
We proceed to constructing this algorithm $\mathcal{B}$, who is supposed to win $\mathsf{Game2}$, which consists of solving decision $\mathsf{A\text{-}LWE}^f_{\kappa,\lambda,\sigma}$ with more than negligible advantage as we detail next:
Suppose $\mathcal{B}$ receives the parameters κ, λ, σ and function f and plays the standard real-or-random game with challenger $\mathcal{C}$ who tests $\mathcal{B}$'s ability of solving the decision $\mathsf{A\text{-}LWE}$ problem. Hence $\mathcal{C}$ possesses an $\mathsf{A\text{-}LWE}$ distribution $L^{\mathsf{A\text{-}LWE}}_{\kappa,\lambda,\sigma}$ and it can generate $\mathsf{A\text{-}LWE}$ samples $(\mathbf{A}, \mathbf{b}^T = \mathbf{s}^{*T}\mathbf{A} + \mathbf{e}^T)$ for some $m \in \mathbb{Z}_q$, where $\mathbf{s}^* \in \mathbb{Z}_q^{\kappa}$ is the secret, $\mathbf{A}$ is a public matrix in $\mathbb{Z}_q^{\kappa\times\lambda}$ and the error term $\mathbf{e} \in \mathbb{Z}_q^{\lambda}$ embeds the message m.

- In $\mathsf{Game2}$, $\mathcal{B}$ is allowed to make **Sample** queries, where she provides an $m \in \mathbb{Z}_q$ to $\mathcal{C}$ who generates an $\mathsf{A\text{-}LWE}$ sample from L accordingly and returns it to $\mathcal{B}$.
- In the **Distinguish** phase $\mathcal{B}$ picks a new message $m^* \in \mathbb{Z}_q$ and sends it to $\mathcal{C}$. Then $\mathcal{C}$ flips a random coin b: if $b = 0$, generate a valid $\mathsf{A\text{-}LWE}$ sample embedding m^* from L, otherwise draw $\mathbf{b}$ uniformly at random from $\mathbb{Z}_q^{\lambda}$ and send the tuple $(\mathbf{A}, \mathbf{b}^T)$ to $\mathcal{B}$.
- Finally, $\mathcal{B}$ outputs her **Guess** whether b is 0 or 1.

She wins the game and hence solves decision A-LWE$^{f}_{\kappa,\lambda,\sigma}$ if her guess of b is correct.

Setup. $\mathcal{B}$ takes $\mathbf{A}$, $\mathbf{g}^T = (1, 2, \ldots, 2^{\lambda-1})$ and $(pk, sk) \leftarrow$ AHOM.Gen that she has received from prior interaction with $\mathcal{C}$ and sends $\mathbf{A}, \mathbf{g}^T$ and pk to $\mathcal{A}$ as the public parameters. Then, $\mathcal{B}$ randomly chooses two distinct indices j and k from $\{0, \ldots, N\}$. The chance of picking the right ones is $\frac{1}{N^2}$, otherwise $\mathcal{B}$ aborts during the game. She sets

$$\mathbf{s}_k := ((\mathbf{b}^T - \mathbf{e}^T)\mathbf{A}^{-1})^T.$$

Hence, $\mathbf{s}_k$ is the secret $\mathbf{s}^*$. Note that $\mathbf{s}_k$ is *unknown* to $\mathcal{B}$. $\mathcal{B}$ samples the users' secret keys $\{\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^\kappa\}_{i\neq k}^{i\neq j}$ and sets

$$\mathbf{s}_j := -\sum_{i\neq j} \mathbf{s}_i = -(\sum_{\substack{i\neq j,\\ i\neq k}} \mathbf{s}_i + \mathbf{s}_k). \tag{3.1}$$

This comes from the inherent requirement of the PSA protocol $\sum_{i=0}^{N} \mathbf{s}_i = \mathbf{0}$. Note that $\mathbf{s}_j$ is also *unknown* to $\mathcal{B}$.
Lastly, $\mathcal{B}$ picks $sk_{A_1} \xleftarrow{\$} \{\mathbf{s}_i\}_{i\neq k}^{i\neq j}$ and sets $sk_{A_2} := sk$. Note that for the first aggregator key it indeed does not matter which $\mathbf{s}_i$ is chosen, since we ensure that $\sum_{i=0}^{N} \mathbf{s}_i = \mathbf{0}$ by choosing $\mathbf{s}_j$ according to Equation (3.1).

Compromise. On request i from $\mathcal{A}$ and if $i \neq j$, $i \neq k$, $\mathcal{B}$ sends the corresponding $\mathbf{s}_i$ to $\mathcal{A}$. If additionally $i = 0$, $\mathcal{B}$ sends (sk_{A_1}, sk_{A_2}). Otherwise, $\mathcal{B}$ aborts. Let K be the set of all compromised users $K = \{i\}$.

Challenge. $\mathcal{A}$ picks a set of uncompromised users $U \subseteq \{0, \ldots, N\}\backslash K$ and plaintext-randomness pairs $\{(d_i, r_i)\}^{i\in U}$ and transmits these to $\mathcal{B}$. Note that by construction $\{j, k\} \subseteq U$. $\mathcal{B}$ computes

$$\{\mathbf{c}_i = \mathsf{NoisyEnc}(pk, \mathbf{g}^T, \mathbf{s}_i, \mathbf{A}, d_i, r_i)\}^{i\in U\backslash\{j,k\}}.$$

Now $\mathcal{B}$ enters the **Distinguish**-phase and sends $m = d_k + r_k \mod q$ to $\mathcal{C}$ who returns the tuple $(\mathbf{A}, \mathbf{b}^T)$. $\mathcal{B}$ sets $\mathbf{c}_k := \mathbf{b}^T$. Note that $\sum_{i\in U} \mathbf{c}_i + \sum_{i\notin U} \mathbf{s}_i^T \cdot \mathbf{A} = \sum_{i\in U} \mathbf{e}_i$ and that

$$\mathsf{AHOM.Dec}(\mathbf{G} \cdot \sum_{i\in U} \mathbf{e}_i \mod q) \stackrel{!}{=} \sum_{i\in U} d_i + r_i.$$

Therefore, $\mathcal{B}$ first computes a valid encryption of $\sum_{i\in U} d_i + r_i$, i.e.

$$\mathbf{v} = \mathsf{AHOM.Enc}(pk, \sum_{i\in U} d_i + r_i) \in \mathbb{Z}_q^{\lambda/l}$$

and then sets

$$\mathbf{c}_j := \mathbf{G}^{-1} \cdot \mathbf{v} - \sum_{i\in U\backslash\{j\}} \mathbf{c}_i - \sum_{i\notin U} \mathbf{s}_i^T \cdot \mathbf{A}.$$

Note that we compute the *left* inverse of $\mathbf{G}$ such that $\mathbf{G}^{-1} \cdot \mathbf{G} = \mathbf{I}_\lambda$. Finally, $\mathcal{B}$ sends all $\{\mathbf{c}_i\}^{i\in U}$ to $\mathcal{A}$.

Guess. If $\mathcal{A}$ has more than negligible advantage in winning the aggregator obliviousness security game, she can distinguish the ciphertexts from random. Specifically, if $\mathbf{c}_k = \mathbf{b}^T$ is indeed a valid A-LWE sample it is a valid encryption of (d_k, r_k) and $\mathcal{A}$ will return 0, otherwise she returns 1. Therefore, by forwarding $\mathcal{A}$'s output to $\mathcal{C}$ as her guess, $\mathcal{B}$ wins the game: she can distinguish $\mathbf{b}^T$ from random and solve decision $\mathsf{A\text{-}LWE}^f_{\kappa,\lambda,\sigma}$. □

Remark 2. *Note that in the case where the adversary compromises all but one participant, she inevitably learns the secret key of that participant and can therefore distinguish between valid encryptions and random values. In this case the definition of aggregator obliviousness requires that she does not learn any additional information about that participant. As stated in Definition 16 this requirement translates into a privacy rather than a security guarantee. Therefore, we address it in the corresponding Theorem 3.*

3.6.2 Privacy of LaPS

We formulated the deployed privacy mechanism, i.e. the type of noise and the way it is added to the user's input, as a separate building block within our scheme LaPS. Therefore, we give a general privacy guarantee based on the DP-level of the chosen privacy mechanism. Concretely, we show (ϵ, δ)-DP of the aggregate output in terms of ϵ-DP of the privacy mechanism.

Theorem 3 (Privacy). *Let $\mathcal{M}_\chi(D, f(\cdot), \epsilon) = f(D) + (Z_1, \ldots, Z_k)$ denote a mechanism for some function $f : \mathcal{D}^N \to \mathcal{R}^k$, where Z_i are i.i.d. random variables drawn according to some distribution χ.*
If $\mathcal{M}_\chi$ achieves ϵ-DP and $f(D) = \sum_{i=1}^N d_i$ for $D = (d_1, \ldots, d_N)$ is a sum query, then the aggregate output generated by the LaPS scheme according to Definition 21 is (ϵ, δ)-DP for $\beta = \min\{\frac{1}{\gamma N} \ln \frac{1}{\delta}, 1\}$, where N denotes the number of participants and γ is the fraction of honest participants.

Proof. The proof follows from [Shi+11, Lemma 1][6] when substituting their Geometric mechanism by $\mathcal{M}_\chi$. □

3.6.3 Trusted Setup

Similar to other PSA schemes (e.g. [Shi+11; Val16]), we assume a *trusted* Setup, where the user encryption keys and the aggregator decryption capability are distributed and are subsequently assumed to be secret. This is commonly implemented using a trusted third party that executes Setup. Note that it is only required once and does not have to be repeated throughout the protocol, i.e. no more interaction. Additionally, by pre-generating a set of multiple keys one may extend this across executions of the protocol. As suggested in [Shi+11] standard secure MPC protocols can be utilized in order to execute the Setup in a distributed manner among the involved parties and thereby avoid a trusted third party.

[6] Note that in [Shi+11] the term *distributed* differential privacy is specifically coined for PSA schemes, since the privacy-preserving noise is generated in a distributed manner. For simplicity, we abuse notation and refer to "differential privacy".

3.7 LaPS Instantiation using BGV Scheme and Discrete Laplace Noise

In our example instantiation we utilize an adapted version of the BGV scheme due to Brakerski *et al.* [BGV12] as the additively homomorphic element. The discrete Laplace mechanism constitutes the privacy mechanism. We first discuss these two components separately (Section 3.7.1 and Section 3.7.2, respectively) before putting them together into an instantiation of our LaPS scheme (Section 3.7.3).
Note that we use a Ring-A-LWE-based instantiation for efficiency reasons. We present all definitions in the remainder of this section in the ring setting to be consistent with our implementation in Section 3.8.2.

3.7.1 Adapted BGV Scheme

The BGV scheme was first introduced by Brakerski *et al.* [BV11a; BV11b; BGV12]. This fully homomorphic encryption scheme is LWE-based and it has seen broad application in many cryptographic constructions [Dam+13; GHS12; MP12; SV14]. Damgård *et al.* [Dam+13] for instance leverage a version of the original BGV scheme from [BGV12] for their MPC protocol, where they extend the plaintext space beyond binary bits. We adapt their notation and thereby also take advantage of their proof of correctness that we adapt in a straightforward way.
Note that we make two modifications to the scheme as it is defined by Damgård *et al.* [Dam+13]. Firstly, since our PSA scheme targets sum aggregation, we do not require the multiplication operation of the BGV scheme. Therefore, we actually reduce it to a *somewhat* homomorphic scheme, which also allows for significant efficiency gains, since the scheme is greatly simplified - for instance by eliminating key-switching and multiplication which is the most computationally intensive operation. Secondly, Damgård *et al.* [Dam+13] define their key generation in a distributed manner due to the MPC context, which we do not require. Consequently, the distributions and magnitudes of the resulting values are different, as we detail in Section 3.7.1.1. Finally, we apply the result from Applebaum *et al.* [App+09] (see Section 2.4.4) and securely sample the secret in the R-LWE term from the error distribution, which lets us optimize the parameter magnitudes.
We utilize the following subroutines due to [Dam+13]:

- $\mathcal{ZO}(0.5, n)$: Generate a vector of length n with elements chosen at random from $\{-1, 0, 1\}$ such that the probabilities for each coefficient are $p_{-1} = \frac{1}{4}$, $p_0 = \frac{1}{2}$, $p_1 = \frac{1}{4}$.

- $\mathcal{DG}(\sigma'^2, n)$: Generate a vector of length n with elements chosen according to the discrete Gaussian distribution with variance σ'^2.

- $\mathcal{RC}(0.5, \sigma'^2, n)$: Generate (v, e_0, e_1) where $v \leftarrow \mathcal{ZO}(0.5, n)$ and $e_0, e_1 \leftarrow \mathcal{DG}(\sigma'^2, n)$.

- $\mathcal{U}(q, n)$: Generate a vector of length n with elements generated uniformly at random modulo q.

Note that in order to control the error that is generated from arithmetic operations on the ciphertexts, the BGV scheme uses *modulus switching*. In

other words, for an input ciphertext that is defined over the modulus q and a target modulus q', the SwitchModulus routine outputs a ciphertext that is defined over the modulus q' but encrypts the same plaintext as the input ciphertext.
Different from the original definition, we do not estimate the incurred error before reducing the modulus - instead, our SwitchModulus routine is equivalent to the function Scale as in [GHS12, Appendix B.2]. It takes an element $x \in R_q$, modulus q and target modulus q', and returns an element $y \in R_{q'}$. In coefficient representation it holds that $y \equiv x \mod p$ and y is the closest element to $(q'/q) \cdot x$ that satisfies this mod-p condition, where p is the plaintext modulus (adapted from [GHS12]). We refer to [GHS12, Appendix D] for details on the evaluation representation. Next, we define our adapted version of the BGV scheme.

Definition 22 (Adapted BGV [BGV12; Dam+13])**.** *Let $R = \mathbb{Z}[X]/\Phi_m(X)$ and $R_q = (\mathbb{Z}/q\mathbb{Z})[X]/\Phi_m(X)$ for some cyclotomic polynomial $\Phi_m(X)$ and integer q, where $\phi(m)$ is the degree of R over $\mathbb{Z}$. Let σ' be the Gaussian standard deviation. The plaintext space is R_p for some prime p and ciphertexts are tuples in $R_{q_1} \times R_{q_1}$, which get reduced (in the decryption process) to tuples in $R_{q_0} \times R_{q_0}$ for the two moduli q_0 and q_1.*
Set q_0, q_1 such that $q_0 = p_0$ and $q_1 = p_0 \cdot p_1$ for some primes p_0, p_1, where $q_0, q_1 > p$.

- BGV.Gen*: Generate $a \leftarrow \mathcal{U}(q_1, \phi(m))$. Draw $s, \epsilon \leftarrow \mathcal{DG}(\sigma'^2, \phi(m))$. Compute $b = a \cdot s + p \cdot \epsilon$ and output (a, b) as the public key and s as the secret key.*

- BGV.Enc$(pk, \mu \in R_p)$*: Using modulus q_1, choose a "small" polynomial, i.e. with $0, \pm 1$ coefficients, and two polynomials with Gaussian coefficients $(v, e_0, e_1) \leftarrow \mathcal{RC}(0.5, \sigma'^2, \phi(m))$. Then set $c_0 = b \cdot v + p \cdot e_0 + \mu$, $c_1 = a \cdot v + p \cdot e_1$ and output ciphertext $c = (c_0, c_1) \in R_{q_1} \times R_{q_1}$.*

- BGV.Dec(sk, c)*: For input ciphertext c defined modulo q_1, invoke* SwitchModulus(c, q_1, q_0)*, which produces a new ciphertext $c' = (c'_0, c'_1)$ defined modulo q_0 such that*

$$\Big((c'_0 - s \cdot c'_1) \mod q_0 \equiv (c_0 - s \cdot c_1) \mod q_1\Big) \mod p.$$

  Decryption of c' is performed by setting $\mu' = (c'_0 - s \cdot c'_1) \mod q_0$ and outputting $\mu' \mod p$.

Remark 3. *We fix the following parameter setting following [Dam+13]: with m as a power of 2, we have that $\phi(m) = m/2$. Select $R = \mathbb{Z}[X]/(X^{m/2} + 1)$ and $p = 1 \mod m$, i.e. $R_p \simeq \mathbb{F}_p^{m/2}$ and ring constant $c_m = 1$.*

As mentioned previously, we can take advantage of the proofs of security and correctness due to Damgård *et al.* [Dam+13] for their BGV scheme after applying our modifications. We state semantic security and correctness of our Adapted BGV scheme below.

Theorem 4 (BGV: Semantic Security)**.** *The BGV scheme according to Definition 22 is semantically secure with pseudo-random ciphertexts assuming the hardness of* decision *Ring*-LWE.

Proof. The proof follows from [Dam+13, Theorem 2] except that in our case the secret key is not generated in a distributed manner. Therefore, we do not require the circular security assumption. Note that the ciphertexts are simply Ring-LWE samples as in Definition 11 and are thus indistinguishable from uniform samples as long as decision Ring-LWE is hard to solve. □

3.7.1.1 Correctness of Adapted BGV

Following the parameter analysis in [GHS12] and [Dam+13], we first analyze the expected magnitudes of values sampled from the distributions listed above Definition 22 before estimating the noise generated in each part of the scheme. This analysis allows us to formulate the correctness requirement of the Adapted BGV scheme (i.e. Inequality 3.5).
For our particular ring setting, where $R = \mathbb{Z}[X]/\Phi_m(X)$ and m is set as a power of 2, we bound the p-norm of a ring element $x \in R$ using its canonical embedding $\mathsf{can}(x) : R \to \mathbb{C}^{\phi(m)}$, i.e. $||x||_\infty \leq ||x||_\infty^{\mathsf{can}} \leq ||x||_1$, where $||x||_\infty^{\mathsf{can}} = ||\mathsf{can}(x)||_\infty$. can maps a ring element x to a $\phi(m)$-vector, where each coefficient is an evaluation of x on the complex primitive m-th root of unity ζ_m^i over all $i \in (\mathbb{Z}/m\mathbb{Z})^*$.
Sampling $x \in R$ from $\mathcal{ZO}(0.5, \phi(m))$ generates a random variable with variance $\mathsf{Var}_Z = \frac{1}{2}\phi(m)$. With distribution $\mathcal{DG}(\sigma'^2, \phi(m))$ we get $\mathsf{Var}_G = \sigma'^2 \cdot \phi(m)$ and $\mathcal{U}(q, \phi(m))$ yields $\mathsf{Var}_U = \frac{q^2}{12} \cdot \phi(m)$.
By the law of large numbers, $||x||_\infty^{\mathsf{can}}$ is bounded by $6 \cdot \sqrt{\mathsf{Var}_i}$ w.h.p., since $\mathsf{erfc}(6) \approx 2^{-55}$, where $i \in \{Z, G, U\}$ depending on which distribution x is sampled from. For two such elements $x, y \in R$ with variances $\mathsf{Var}(\mathsf{can}(x))$ and $\mathsf{Var}(\mathsf{can}(y))$ respectively, we bound the product $||x \cdot y||_\infty^{\mathsf{can}}$ by

$$16\sqrt{\mathsf{Var}(\mathsf{can}(x))} \cdot \sqrt{\mathsf{Var}(\mathsf{can}(y))},$$

since $\mathsf{erfc}(4)^2 \approx 2^{-50}$. Consequently, we get the following bounds on the secret key s and the public key components a and ϵ from the key generation routine BGV.Gen according to Definition 22:

$$\begin{aligned}
\mathsf{Var}(\mathsf{can}(a)) &= \mathsf{Var}_U = \frac{q_1^2}{12} \cdot \phi(m), \\
\mathsf{Var}(\mathsf{can}(s)) &= \mathsf{Var}_G = \sigma'^2 \cdot \phi(m), \\
\mathsf{Var}(\mathsf{can}(\epsilon)) &= \mathsf{Var}_G = \sigma'^2 \cdot \phi(m).
\end{aligned}$$

As in [Dam+13] we define the noise of a ciphertext $c = (c_0, c_1)$ as an upper bound on $||c_0 - s \cdot c_1||_\infty^{\mathsf{can}}$.

In the following we look at the noise from "fresh" ciphertexts, i.e. those generated by BGV.Enc, and the noise in reduced ciphertexts, i.e. outputs of SwitchModulus that is invoked during decryption in BGV.Dec bounded according to Definition 22.

Fresh ciphertexts. If $c = (c_0, c_1) = \mathsf{BGV.Enc}(pk, \mu)$, then the noise in c is

bounded by

$$\begin{aligned}
&||c_0 - s \cdot c_1||_\infty \leq ||c_0 - s \cdot c_1||_\infty^{\mathsf{can}} \\
&= ||(a \cdot s + p \cdot \epsilon) \cdot v + p \cdot e_0 + \mu - s \cdot (a \cdot v + p \cdot e_1)||_\infty^{\mathsf{can}} \\
&= ||\mu + p \cdot (\epsilon \cdot v + e_0 - e_1 \cdot s)||_\infty^{\mathsf{can}} \\
&\leq ||\mu||_\infty^{\mathsf{can}} + p \cdot (||\epsilon \cdot v||_\infty^{\mathsf{can}} + ||e_0||_\infty^{\mathsf{can}} + ||e_1 \cdot s||_\infty^{\mathsf{can}}) \\
&\leq \phi(m) \cdot (p-1) + p \cdot \Big(16 \cdot \sqrt{\sigma'^2 \cdot \phi(m) \cdot \frac{1}{2} \cdot \phi(m)} + \\
&\quad 6 \cdot \sqrt{\sigma'^2 \cdot \phi(m)} + 16 \cdot \sqrt{\sigma'^2 \cdot \phi(m) \cdot \sigma'^2 \cdot \phi(m)}\Big) \\
&= \phi(m) \cdot (p-1) + 2p\sigma' \cdot ((8 + 4\sqrt{2}) \cdot \phi(m) + 3 \cdot \sqrt{\phi(m)}) \\
&\quad = B_{clean}.
\end{aligned}$$

Reduced ciphertexts. If input ciphertext c has noise ν then output ciphertext $c' = \mathsf{SwitchModulus}(c_0, c_1)$ has noise ν', where $\nu' = \frac{q_0}{q_1} \cdot \nu + B_{scale} = \frac{\nu}{p_1} + B_{scale}$. Recall that $q_0 = p_0 \cdot p_1$.
B_{scale} captures overhead noise from the rounding error caused by reducing to modulus $q_1 = p_1$. Let $\tau = (\tau_0, \tau_1)$ be the rounding error, i.e. $(\tau_0, \tau_1) = (c'_0, c'_1) - \frac{q_0}{q_1}(c_0, c_1)$. Then, $\mathsf{can}(\tau_i)$ is roughly distributed according to a complex Gaussian with variance $\frac{p^2}{12} \cdot \phi(m)$. Therefore,

$$||\tau_0 + \tau_1 \cdot s||_\infty^{\mathsf{can}} \quad \leq \quad \frac{1}{\sqrt{3}} \cdot p \cdot (3 \cdot \sqrt{\phi(m)} + \sigma' \cdot \phi(m)) = B_{scale}.$$

Sum of ciphertexts. Summing ciphertexts $c_1, \ldots, c_N$ with noises $\nu_1, \ldots, \nu_N$ respectively, results in the total noise $\nu = \sum_{i=1}^{N} \nu_i$.
BGV.Dec takes a sum of some N ciphertexts $c = \sum_{i=1}^{N} c_i$ as input, where each $c_i \leftarrow \mathsf{BGV.Enc}$ is a fresh ciphertext. Then c is reduced to c' using $\mathsf{SwitchModulus}(c, q_1, q_0)$ and the plaintext is retrieved via $(c'_0 - s \cdot c'_1 \mod q_0) \mod p$. Therefore, in order to decrypt correctly

$$\nu' < \frac{q_0}{2} = \frac{p_0}{2}, \tag{3.2}$$

where ν' is the noise associated to c'. We can bound ν' using the bounds given above:

$$\begin{aligned}
\nu' &\leq \frac{N \cdot B_{clean}}{p_1} + B_{scale} && (3.3) \\
&= \frac{N \cdot \Big(\phi(m) \cdot (p-1) + 2p\sigma' \cdot ((8 + 4\sqrt{2}) \cdot \phi(m) + 3 \cdot \sqrt{\phi(m)})\Big)}{p_1} \\
&\quad + \frac{1}{\sqrt{3}} \cdot p \cdot (3 \cdot \sqrt{\phi(m)} + \sigma' \cdot \phi(m)). && (3.4)
\end{aligned}$$

3.7.1.2 Parametrization for Correctness and Security

Combining Inequalities (3.2) and (3.3) from our analysis above yields the following correctness requirement:

$$\frac{N \cdot B_{clean}}{p_1} + B_{scale} < \frac{q_0}{2} = \frac{p_0}{2}, \tag{3.5}$$

where B_{clean} and B_{scale} are defined based on parameters $\phi(m)$, p and σ' as in Equation (3.4).
Following Lindner and Peikert's [LP11] bit-security estimations we get the following security requirement:

$$\phi(m) \geq \frac{(k+110)\cdot \ln(q_1/\sigma')}{7.2} \tag{3.6}$$

for bit-security level k, where $\phi(m)$ is the ring degree, q_1 is the modulus and σ' is the Gaussian parameter.

3.7.2 Discrete Laplace Mechanism

The privacy mechanism within our LaPS scheme is also formulated as an individual building block. For our instantiation here, we use the discrete Laplace mechanism, which is essentially the discrete version of the Laplace mechanism $\mathcal{M}_{Lap}$ as in Definition 13. As noted previously, the Laplace mechanism is a standard differential privacy mechanism, which we discretize in order to make it suitable for our cryptographic application as is common practice. We restate the definition of the underlying discrete distribution next, before defining the mechanism and stating its ϵ-DP.

Definition 23 (Discrete Laplace (DLap) [IK06])**.** *The discrete Laplace distribution with scale $\varsigma > 1$ and parameter $p = \exp(-1/\varsigma) \in (0,1)$ is the distribution supported on $\mathbb{Z}$ with probability mass function*

$$DLap_\varsigma(x) = \frac{1-p}{1+p}p^{|x|} = \frac{1-\exp\left(-\frac{1}{\varsigma}\right)}{1+\exp\left(-\frac{1}{\varsigma}\right)}\exp\Big(-\frac{|x|}{\varsigma}\Big).$$

This distribution is denoted by $DLap_\varsigma$.

Definition 24 (DLap-Mechanism $\mathcal{M}_{DLap}$ [GRS09])**.** *Given any function $f : \mathcal{D}^n \to \mathcal{R}^k$, the discrete Laplace mechanism is defined as:*

$$\mathcal{M}_{DLap}(D, f(\cdot), \epsilon) = f(D) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables drawn from $DLap_\varsigma$ as in Definition 23 with $\varsigma = \Delta f/\epsilon$.

Observe that $\mathcal{M}_{DLap}$ is equivalent to the Geometric mechanism [GRS09] that Shi *et al.* [Shi+11] use in their PSA scheme. While they do not explicitly prove ϵ-DP, it is reflected in [Shi+11, Fact 1]. We state ϵ-DP of $\mathcal{M}_{DLap}$ for completeness.

Lemma 16 (DLap-Mechanism: ϵ-DP)**.** *The discrete Laplace mechanism $\mathcal{M}_{DLap}$ preserves ϵ-DP.*

Proof. The proof follows a standard structure that is common in differential privacy literature, see e.g. [DR14, Theorem 3.6]. In fact, it is analogous to the widely known continuous version of the Laplace mechanism, i.e. $\mathcal{M}_{Lap}$ as in Definition 13, with the exception of having a discrete function and distribution range.
Let $D_0 \in \mathcal{D}^n$ and $D_1 \in \mathcal{D}^n$ be adjacent databases, let $f(\cdot)$ be some function $f : \mathcal{D}^n \to \mathcal{R}^k$ and let the l_1-norm of a database D be denoted $\|D\|_1$,

where $\|D\|_1 = \sum_{i=1}^{|\mathcal{D}|} |D_i|$ and $\mathcal{D}$ denotes the universe of records. Let $\Delta f = \max_{D_0, D_1 \text{ adjacent}} \|f(D_0) - f(D_1)\|_1$ according to [DR14].
Comparison at some arbitrary point $z \in \mathcal{R}^k$ yields

$$\begin{aligned}
\frac{\Pr[\mathcal{M}_{DLap}(D_0, f, \epsilon) = z]}{\Pr[\mathcal{M}_{DLap}(D_1, f, \epsilon) = z]} &= \prod_{i=1}^{k} \left(\frac{\exp(-\frac{\epsilon|z_i - f(D_0)_i|}{\Delta f})}{\exp(-\frac{\epsilon|z_i - f(D_1)_i|}{\Delta f})} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\epsilon(|z_i - f(D_1)_i| - |z_i - f(D_0)_i|)}{\Delta f} \right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\epsilon|f(D_0)_i - f(D_1)_i|}{\Delta f} \right) \\
&= \exp\left(\frac{\epsilon\|f(D_0) - f(D_1)\|_1}{\Delta f} \right) \leq \exp(\epsilon)
\end{aligned}$$

Similarly, $\frac{\Pr[\mathcal{M}_{DLap}(D_1, f, \epsilon) = z]}{\Pr[\mathcal{M}_{DLap}(D_0, f, \epsilon) = z]} \geq \exp(-\epsilon)$ by symmetry. □

3.7.3 Putting It Together

By leveraging the Adapted BGV scheme from Section 3.7.1 as the additively homomorphic element and utilizing the Discrete Laplace Mechanism as discussed in Section 3.7.2 as the scheme's privacy mechanism and plugging these components into Definition 21, we can now assemble a complete instantiation of our LaPS scheme. In the remainder of this section, we cover some preliminary definitions of the scheme's underlying algebra - particularly pertaining to the fact that our definitions are in the ring setting - before stating the full definition of our resulting instantiation.
Due to our "layered" approach, particularly in the encryption, we use several moduli throughout the scheme: prime p is the plaintext modulus and q_1 is the final ciphertext modulus, i.e. corresponding to the users' outputs. When the aggregated ciphertext is decrypted by the aggregator, BGV.Dec internally reduces the modulus to q_0. Moduli $q_0, q_1 > p$ are set such that $q_0 = p_0$ and $q_1 = p_0 \cdot p_1$ for some primes p_0, p_1.
The BGV routines produce and process internal ciphertexts and the final user outputs and aggregator inputs of the PSA scheme are external ciphertexts. We define the resulting rings as: the plaintext space $R_p = (\mathbb{Z}/p\mathbb{Z})[X]/\Phi_{m'}(X)$, the internal key and ciphertext space $R_{\text{int}} = (\mathbb{Z}/q_1\mathbb{Z})[X]/\Phi_{m'}(X)$ and the external key and ciphertext space $R_{\text{ext}} = (\mathbb{Z}/q_1\mathbb{Z})[X]/\Phi_m(X)$ for some cyclotomic polynomials $\Phi_m(X)$ and $\Phi_{m'}(X)$.
Setting m' to be a power of two and p such that $p \mod m' \equiv 1$, yields $\phi(m') = \frac{m'}{2}$ as the degree of R_p and R_{int}. Note that the only difference between R_{int} and R_{ext} is the dimension: namely choose[7] $\phi(m)$ s.t. $\phi(m) = 2 \cdot \phi(m') \cdot l$, where $l = \lceil \log q_1 \rceil$.
In order to transform from R_{ext} to $\mathbb{Z}_{q_1}$, particularly for the utilization of Algorithm 1, we define the following mappings:

[7] Note that $\phi(m)$ can also be made larger than $2 \cdot \phi(m') \cdot l$ by slightly tweaking the defined mappings Z2R and R2Z: simply fill up the coefficient representation with 0's to pad up to the desired length $\phi(m)$ in order to get a ring element in R_{ext} and remove the same number of 0's when transforming that ring element back to $\mathbb{Z}_{q_1}^{2 \cdot \phi(m') \cdot l}$.

- $\mathsf{z2R}_{q,m} : \mathbb{Z}_q \to R_q$: takes a scalar x over the q-ary field and produces a vector $y = (x, 0, \ldots, 0)$ of dimension $\phi(m)$, where y is coefficient representation for the output ring element.
- $\mathsf{R2z}_{q,m} : R_q \to \mathbb{Z}_q$: takes a ring element and outputs the first element of its coefficient representation.
- $\mathsf{R2Z}_{q,m} : R_q \to \mathbb{Z}_q^{\phi(m)}$: outputs a vector of size $\phi(m)$ by copying the entries of the coefficient representation of the input ring element.
- $\mathsf{Z2R}_{q,m} : \mathbb{Z}_q^{\phi(m)} \to R_q$: interprets the input vector as the coefficient representation of a polynomial in R_q and outputs the corresponding ring element.

We state the complete definition of the example instantiation of our LaPS scheme next. Note that addition and multiplication operations of ring elements are performed component-wise and that parameters of BGV-routines and internal ciphertexts are denoted with bars above the variable names for better readability.

Definition 25 (LaPS using BGV and $\mathcal{M}_{DLap}$)**.** *Let κ be a security parameter, $N \in \mathbb{N}$ the number of participants, γ the fraction of uncompromised participants and let $\varsigma > 1$. Fix the rings R_p, R_{int} and R_{ext} with the corresponding parameters $p, p_0, p_1, q_0, q_1, m, m', l$ as described above. Let $\kappa = \phi(m)/l$.*

- $(\{a, \mathbf{g}^T, pk\}, \{s_i\}, (sk_{A_1}, sk_{A_2})) \leftarrow \mathsf{Setup}(1^\kappa)$*: Generate the public parameters a, $\mathbf{g}^T$ and pk as follows and distribute them to all parties.*
  1. *Draw a uniformly at random from R_{ext}.*
  2. *Set vector $\mathbf{g}^T = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_{q_1}^l$.*
  3. *Generate $((\bar{a}, \bar{b}), \bar{s}) \leftarrow \mathsf{BGV.Gen}$ and extract public key pk as $pk = (\bar{a}, \bar{b}) \in R_{\mathsf{int}} \times R_{\mathsf{int}}$.*
  4. *For all $i \in \{1, \ldots, N\}$ draw $s_i \leftarrow R_{\mathsf{ext}}$ and send it to user i as her secret key.*
  5. *The aggregator's secret decryption key is the tuple (sk_{A_1}, sk_{A_2}), where*
     - *$sk_{A_1} = -\sum_{i=1}^N s_i$ and*
     - *$sk_{A_2} = \bar{s} \in ((\bar{a}, \bar{b}), \bar{s}) \leftarrow \mathsf{BGV.Gen} \in R_{\mathsf{int}}$.*
- $c_{i,a} \leftarrow \mathsf{NoisyEnc}_i(\{a, \mathbf{g}^T, pk\}, s_i, d_i)$*: Each user i takes her data $d_{i,a} \in \mathcal{D}$ and adds some noise r_i to it, such that $x_i = d_i + r_i \mod p \in \mathbb{Z}_p$.*
  1. *r_i is sampled as follows:*
     $$r_i = \begin{cases} 0 & \text{with probability } 1-\beta \\ Y & \text{with probability } \beta \end{cases},$$
     where $Y \leftarrow DLap_\varsigma$ and $\beta = \frac{1}{\gamma N} \log(\frac{1}{\delta})$.
  2. *Set $\bar{x}_i = \mathsf{z2R}_{p,m'}(x_i) \in R_p$ and compute $\bar{c} = (\bar{c}_0, \bar{c}_1) \leftarrow \mathsf{BGV.Enc}(pk, \bar{x}_i)$.*
  3. *Set $\mathbf{v}_i = (\mathsf{R2Z}_{q_1,m'}(\bar{c}_0) || \mathsf{R2Z}_{q_1,m'}(\bar{c}_1)) \in \mathbb{Z}_{q_1}^{2\cdot\phi(m')}$, where $||$ denotes concatenation.*
  4. *Invoke Algorithm 1 for each component of $\mathbf{v}_i$: $\mathbf{e}_i = (\mathsf{Sample}(\mathbf{g}^T, v_{i_1}, \sigma), \ldots, \mathsf{Sample}(\mathbf{g}^T, v_{i_{2\phi(m')}}, \sigma))$. Hence, $\mathbf{e}_i \leftarrow D_{\Lambda^\perp_{\mathbf{v}_i}(\mathbf{G}),\sigma} \in \mathbb{Z}_{q_1}^{2\cdot\phi(m')\cdot l}$*

5. *Transform to the ring by* $e_i = \mathsf{Z2R}_{q_1,m}(\mathbf{e}_i)$. *Note that since* $\phi(m) = 2 \cdot \phi(m') \cdot l$, $e_i \in R_{\mathsf{ext}}$.
6. *Output the ciphertext* $c_{i,a} = a \cdot s_i + e_i \in R_{\mathsf{ext}}$.

- $\sum_{i=1}^{N} x_i \leftarrow \mathsf{AggrDec}(\{a, \mathbf{g}^T\}, (sk_{A_1}, sk_{A_2}), \{c_{1,a}, \ldots, c_{N,a}\})$: *Receiving the users' ciphertexts* $\{c_{i,a}\}$ *the aggregator computes* $c = \sum_{i=1}^{N} c_{i,a}$.
  1. *Compute* $e = \sum_{i=1}^{N} e_i = c + a \cdot sk_{A_1}$.
  2. *Set* $\mathbf{e} = \mathsf{R2Z}_{q_1,m}(e) \in \mathbb{Z}_{q_1}^{\phi(m)}$.
  3. *Compute* $\mathbf{v} = \mathbf{G} \cdot \mathbf{e} \mod q_1 \in \mathbb{Z}_{q_1}^{2\cdot\phi(m')}$, *where* $\mathbf{G} = \mathbf{I}_{\phi(m)/l} \otimes \mathbf{g}^T \in \mathbb{Z}_{q_1}^{\frac{\phi(m)}{l} \times \phi(m)}$. *Again, note that* $\phi(m)/l = 2 \cdot \phi(m')$ *and that* $\mathbf{v}$ *is the sum of the individual* $\mathbf{v}_i$*'s from* $\mathsf{NoisyEnc}_i$.
  4. *Parse* $\mathbf{v}$ *as a tuple of vectors* $\mathbf{v} = (\mathbf{v}', \mathbf{v}'') \in \mathbb{Z}_{q_1}^{\phi(m')} \times \mathbb{Z}_{q_1}^{\phi(m')}$.
  5. *Decrypt:* $\bar{x} = \mathsf{BGV.Dec}(sk_{A_2}, (\mathsf{Z2R}(\mathbf{v}'), \mathsf{Z2R}(\mathbf{v}''))) \in R_p$. *Note that* $\bar{x}$ *is the sum of the individual* $\bar{x}_i$*'s from* $\mathsf{NoisyEnc}_i$.
  6. *The aggregator retrieves the noisy sum of the users' values with* $\sum_{i=1}^{N} x_i = \mathsf{R2z}_{p,m'}(\bar{x}) \in \mathbb{Z}_p$.

As an instantiation of our general framework, above scheme inherits the security and privacy guarantees of the general LaPS scheme (see Section 3.6.1 and 3.6.2, respectively). This means that we essentially only need to show that the requirements of Inequality (3.5) for correctness, Theorem 2 for aggregator obliviousness and Theorem 3 for privacy are satisfied.

3.7.3.1 Correctness of LaPS using BGV and $\mathcal{M}_{DLap}$

The overall PSA scheme is correct as long as the internal BGV Scheme is correct, since the users' noisy sum is recovered using BGV.Dec. Therefore, correct decryption through the AggrDec routine is ensured iff Inequality (3.5) is met with respect to R_{int}, i.e. fixing $\sigma' = 3.2$ as in [GHS12; Dam+13][8]

$$\frac{N \cdot B_{clean}}{p_1} + B_{scale} < \frac{q_0}{2} = \frac{p_0}{2}, \tag{3.7}$$

where $B_{clean} = \phi(m') \cdot (p-1) + 6.4p \cdot ((8 + 4\sqrt{2}) \cdot \phi(m') + 3 \cdot \sqrt{\phi(m')})$ and $B_{scale} = \frac{1}{\sqrt{3}} \cdot p \cdot (3 \cdot \sqrt{\phi(m')} + 3.2 \cdot \phi(m'))$.

3.7.3.2 Security of LaPS using BGV and $\mathcal{M}_{DLap}$

Theorem 5 (Semantic Security). *Let* $\sigma' \geq \omega(1)$, $\epsilon = \mathsf{negl}(\kappa)$. *The ciphertexts generated by* NoisyEnc *in the PSA scheme according to Definition 25 are semantically secure for* $\sigma \geq \omega(\sqrt{\log(\kappa)}) \cdot (\kappa N / \log(\kappa N))^{\frac{1}{4}}$ *assuming the hardness of worst-case lattice problems.*

Proof. Suppose, $\mathbf{v}_i$ that is generated in NoisyEnc in Step (3), is indistinguishable from random. Then, by Lemma 13 $D_{\Lambda^{\perp}_{\mathbf{v}_i}(\mathbf{G}),\sigma}$ correctly simulates the discrete Gaussian distribution $D_{\mathbb{Z}^{\phi(m)},\sigma}$ for $\sigma \geq \eta_\epsilon(\mathbf{G})$. Note that

[8] Gentry *et al.* [GHS12] originally choose this value according to the parameter analysis from Micciancio and Regev [MR09].

$\mathbf{G} = \mathbf{I}_{\phi(m)/l} \otimes \mathbf{g}^T$. Adopting El Bansarkhani *et al.*'s [EDB15] argumentation, $\mathbf{G}$ induces the lattice $\Lambda_{q_1}^{\perp}(\mathbf{G}) = \{\mathbf{x} \in \mathbb{Z}^{\phi(m)} | \mathbf{G}\mathbf{x} \equiv 0 \mod q_1\}$ with generator matrix $\mathbf{S} = \mathbf{I}_{\phi(m)/l} \otimes \mathbf{S}_l \in \mathbb{Z}_{q_1}^{\phi(m)/l \times l}$, where

$$\mathbf{S}_l = \begin{bmatrix} 2 & & & 0 \\ -1 & 2 & & \\ & \ddots & \ddots & \\ 0 & & -1 & 2 \end{bmatrix} \in \mathbb{Z}_{q_1}^{l \times l}.$$

Using Lemma 12, which states an upper bound on the smoothing parameter $\eta_\epsilon(\Lambda)$ of a given lattice Λ and its basis, the smoothing parameter $\eta_\epsilon(\Lambda_{q_1}^{\perp})$ is bounded from above by $||\mathbf{S}|| \cdot \sqrt{\ln(2\frac{\phi(m)}{l}(1+\frac{1}{\epsilon}))/\pi} \leq 2 \cdot \sqrt{\phi(m)/l}$. By definition $\kappa = \phi(m)/l$ and hence $\eta_\epsilon(\Lambda_{q_1}^{\perp}) \leq 2 \cdot \sqrt{\kappa}$.
Consequently, Lemma 13 applies and the ciphertexts $c_{i,a}$ represent plain Ring-A-LWE samples. Observe that the hardness of decision A-LWE carries over to the ring setting in a straightforward manner[9] as shown in [EDB14]. Therefore, the statement follows immediately from the hardness of decision Ring-A-LWE: Ring-A-LWE samples are indistinguishable from R-LWE samples due to Lemma 7 when applied to the ring setting. decision R-LWE is as hard as worst-case lattice problems [PRSD17]. The latter holds as long as $\sigma \geq \omega(\sqrt{\log(\kappa)}) \cdot (\kappa N / \log(\kappa N))^{\frac{1}{4}}$, which is given by assumption. Observe that this parameter constraint arises from [PRSD17, Corollary 7.3], which states the hardness of solving decision Ring-LWE with spherical error. Note that the main theorem in [PRSD17] is not applicable here as the error terms resulting from the sampling routine Sample are spherical [SD17].
Finally, note that $\mathbf{v}_i$ in Step (3) is the ring-transform of internal ciphertext $\bar{c}$ from Step (2). Semantic security and pseudo-randomness of internal ciphertexts follows from Theorem 4 assuming the hardness of Ring-LWE. Since $\sigma \geq \omega(1)$, hardness of decision Ring-LWE is satisfied according to Theorem 6.2 in [PRSD17] and $\mathbf{v}_i$ is indeed indistinguishable from random and subsequently the claim follows. □

Aggregator obliviousness is also inherited from aggregator obliviousness of the general LaPS scheme (see Section 3.6.1). The corresponding Theorem 2 requires pseudo-randomness of the internal ciphertexts. This property comes with the semantic security of the NoisyEnc routine in our LaPS instance, which we indeed show in Theorem 5. Therefore, aggregator obliviousness of our example instantiation follows immediately. We state the resulting theorem below.

Theorem 6 (Aggregator Obliviousness Security)**.** *Let parameters be as in Theorem 5. Then, the PSA scheme according to Definition 25 satisfies aggregator obliviousness security assuming the hardness of worst-case lattice problems.*

Proof. This follows directly from aggregator obliviousness of LaPS due to Theorem 2 when applied to the ring setting and semantic security of the NoisyEnc routine in our LaPS instance, which follows from Theorem 5. □

[9] See [EDB14, Section 4.5] for an example of a BGV-based encryption scheme that is reduced to the ring variant of A-LWE.

3.7.3.3 Privacy and Accuracy of Aggregate Output

Following Shi *et al.*'s [Shi+11] argument, suppose all user values from the data domain $\mathcal{D}$ are inside an interval of width Δ in $\mathbb{Z}_p$. If γN users each add noise of magnitude $\Theta(\frac{\Delta}{\epsilon})$, where γ is the fraction of uncompromised users, then the total accumulated noise in the aggregate output has magnitude roughly $O(\frac{\Delta}{\epsilon}\sqrt{N})$. We state the resulting privacy and accuracy guarantees of the scheme next.

Theorem 7 (Privacy & Accuracy). *Let $\epsilon > 0$, $0 < \delta < 1$, $\Delta \geq \frac{\epsilon}{3}$, $\gamma \geq \frac{1}{N}\ln(\frac{1}{\delta})$ and $\varsigma = \Delta/\epsilon$. The output of* AggrDec *as in Definition 25 is* (ϵ, δ)-DP.
Moreover, the aggregate achieves $(\frac{4\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}\ln(\frac{1}{\delta})\ln(\frac{2}{\eta})}, \eta)$*-accuracy for all* η *such that* $\ln(\frac{2}{\eta}) \leq \frac{1}{\gamma}\ln(\frac{1}{\delta})$.

Proof. With the chosen parameters it is straightforward to recognize the randomization procedure in NoisyEnc according to Definition 25 as a randomized Discrete Laplace mechanism $\mathcal{M}_{DLap}$ as in Definition 24, where the function f is the sum function. (ϵ, δ)-DP of the aggregate thus follows immediately from ϵ-DP of $\mathcal{M}_{DLap}$ due to Theorem 16 and DP of LaPS due to Theorem 3. Lastly, (α, β)-accuracy follows from utility of Shi *et al.*'s [Shi+11] randomization procedure. □

3.8 Experimental Results

In this section we evaluate several parameter sets that satisfy the security and correctness requirements of Inequality (3.7), Theorem 5 and Inequality (3.8). Furthermore, we present experimental results from implementing the example instantiation of our LaPS scheme as in Definition 25. We also put these results into context by discussing how our scheme performs in comparison to previous schemes due to Shi *et al.* [Shi+11] and Valovich [Val16], which are the most closely related constructions from previous work.

3.8.1 Example Parameters

The BGV parameters corresponding to a certain bit-security level (see Section 3.7.1.2) apply to any (Ring-)LWE-based construction. Therefore, from Inequality (3.6) we get the following requirement for the parameters of our example instantiation (i.e. R_{ext} and Gaussian parameter σ):

$$\phi(m) \geq \frac{(k + 110) \cdot \ln(q_1/\sigma)}{7.2}, \tag{3.8}$$

where k is the bit-security level and all other parameters are as in Definition 25. Finally, Inequality (3.7), Theorem 5 and Inequality (3.8) yield the set of parameter constraints for correctness, semantic security and bit-security, respectively. We give an overview of possible valid parameter sets for different bit-security levels k, plaintext modulus p and number of participants N.
Note that the previous constraints result in some circular dependencies of

the parameters. Hence, we fix $\sigma = 0.1q_1$ and first choose p_1 before picking[10] $\phi(m')$. Finally, we plug these parameters into Inequality (3.7), which yields p_0 and consequently q_1 since $q_1 = p_0 \cdot p_1$. Our results are shown in Tables 3.1, 3.2 and 3.3.

N	$\log(p_0 = q_0)$	$\log(q_1 = p_0 \cdot p_1)$
100	31	36
1000	34	39
10000	38	43

TABLE 3.1: Parameters for plaintext modulus $p \approx 2^{16}$, bit-security level $k = 80$, ring degrees $\phi(m') = 32$ and $\phi(m) \approx 2^{11}$, where N is the number of participants and $p_1 \approx 2^5$ [BGZ18].

Comparing these parameters to Damgård *et al.*'s [Dam+13] instantiation of the BGV scheme, our instantiation allows for much smaller moduli, i.e. our q_1 has magnitude 2^{63} for 100 users compared to 2^{252} in [Dam+13, Appendix G.4] with the same parameters k, p and $\phi(m')$, see Table 3.2. Observe that this improvement stems from our much less restrictive correctness requirement (i.e. Inequality (3.7)) due to the fact that we do not require correct evaluation of homomorphic multiplication.

N	$\log(p_0 = q_0)$	$\log(q_1 = p_0 \cdot p_1)$
100	48	63
1000	49	64
10000	52	67

TABLE 3.2: Parameters for plaintext modulus $p \approx 2^{32}$, bit-security level $k = 128$, ring degrees $\phi(m') = 8192$ and $\phi(m) \approx 2^{20}$, where N is the number of participants and $p_1 \approx 2^{15}$ [BGZ18].

These parameters also show the scalability of our construction in terms of number of participants: for large enough moduli p and q_1 the number of users can grow to a large extent without significantly affecting the other parameters, see Table 3.3.

N	$\log(p_0 = q_0)$	$\log(q_1 = p_0 \cdot p_1)$
10000	146	196
10^{15}	151	201
10^{21}	171	221

TABLE 3.3: Parameters for plaintext modulus $p \approx 2^{128}$, bit-security level $k = 80$, ring degrees $\phi(m') = 32768$ and $\phi(m) \approx 2^{24}$, where N is the number of participants and $p_1 \approx 2^{50}$ [BGZ18].

3.8.2 Implementation

We present our experimental results from implementing our PSA scheme as in Definition 25. We conducted our experiments on a MacBook running

[10] Observe that the first two constraints are dependent on q_1 but are easily satisfied as long as $q_1 \gg \phi(m')$.

macOS Sierra with a single 2.5 GHz Intel Core i7 and 16GB memory using part of the HElib C++ library[11].

	p	NoisyEnc (ms)	AggrDec (ms)
(I)	$5 \approx 2^2$	3.57646	1.86864
	$37 \approx 2^5$	3.61646	1.882
	$65537 \approx 2^{16}$	3.72438	1.96416
(II)	$65537 \approx 2^{16}$	77.3304	67.6243

TABLE 3.4: Results for $N = 1000$,
(I) $k = 80$, $\phi(m') = 32$, $\phi(m) = 2048$,
(II) $k = 128$, $\phi(m') = 512$, $\phi(m) \approx 2^{15}$, $q_1 \approx 2^{44}$ [BGZ18].

With setting (I) we select the parameter set satisfying correctness and security requirements with bit-security level $k = 80$ according to Table 3.1 and different plaintext spaces $p \lesssim 2^{16}$. For completeness, we also select another parameter setting (II) for bit-security level $k = 128$ and plaintext space $p \approx 2^{16}$.
We list the results in Table 3.4, where we measured the average encryption and decryption runtime over 1000 runs each for $N = 1000$ participants in milliseconds. For example, for 1000 people under 65 years old[12] this can be used to compute their average age.
For target DP-parameters $\epsilon = 1$, $\delta = 0.1$, we have $\gamma \geq 0.0023$, i.e. at least 3 out of 1000 participants should be uncompromised. Subsequently, we achieve $(400 \cdot (p-1), 2/\exp(10))$-accuracy due to Theorem 7 when choosing $\eta = 2/\exp(10)$, where the accumulated sum does not exceed $1000 \cdot (p-1)$.

3.8.3 Evaluation

In order to put our experimental results into context, we discuss performance results from previous work. It is noteworthy that to the best of our knowledge no other comparable[13] PSA scheme is equipped with an implementation. We focus on Shi *et al.*'s [Shi+11] as well as Valovich's [Val16] works, as they specify high-level guidelines in evaluating the accuracy and performance of their schemes. We will use their findings as a benchmark for our results.

Accuracy. Valovich and Alda [VA15] analyze how different noise distributions perform in terms of accuracy when used in privacy mechanisms. In their comparison the accuracies of the Geometric mechanism, i.e. $\mathcal{M}_{DLap}$ as in Definition 24, and of the Skellam mechanism, which Valovich utilizes in his PSA scheme [Val16], are about the same. In contrast, the Binomial mechanism performs much worse.

Runtime. Shi *et al.* [Shi+11] estimate the NoisyEncrypt routine of their PSA scheme to take 6 ms for a classic Diffie-Hellman group modular a 1024-bit

[11] https://github.com/shaih/HElib
[12] According to [Bur15] the majority of Americans, i.e. 85.9 %, is under 65 years of age.
[13] One may consider Li and Cao's [LC13] work an exception as the authors indeed provide runtime results. However, their scheme is not comparable to ours as it accounts for dynamic user leaves and joins, which our scheme does not currently support.

prime. According to the authors, this decreases to about 0.6 ms when using high-speed elliptic curves. Our scheme does outperform the former (see setting (I) in Table 3.4) but comes short of the latter by a factor of roughly 6. For the decryption operation, the authors [Shi+11] estimate that a brute-force approach-based implementation of their scheme would take around 0.3 s for 1000 participants. Our decryption is more than 150 times faster. Furthermore, Shi *et al.*'s [Shi+11] results assume that the plaintext space is limited to a single bit - in contrast, our example scheme implementation allows for the encryption of up to 16 bits, i.e. the plaintext space is much larger. On a final note, though we cannot compare these results to any previous work, our measurements for the 128-bit security level also indicate practical runtimes. Note that due to the limited availability of PSA scheme implementations - in fact, this is the first implementation of a lattice-based PSA scheme - these results can only give an idea of the possible performance of our schemes.

3.9 Extensions

Shi *et al.* [Shi+11] consider certain modifications of their PSA scheme that extend beyond summation. For instance, evaluating the user *data distribution*, allowing for *public access* of the aggregate result, which would eliminate the need for an explicit aggregating party, or aggregating subsets of the user data that can be accessed through hierarchical *access control*. We expect the authors' suggestions to be also applicable to our construction.
Our scheme currently does not support *user failures*, i.e. we require each user to submit a ciphertext, otherwise the protocol has to start over, which is also the case for Shi *et al.*'s [Shi+11] PSA scheme. Chan *et al.* [CSS12] propose a solution, where the users are essentially grouped into sub-groups along a binary tree. Consequently, their construction tolerates some dynamic joins and leaves of users at the cost of decreased accuracy of the aggregate output and added communication overhead. We expect that this approach also applies to our construction in a straightforward manner.

Chapter 4

Privacy-Preserving Social Media Advertising Architecture (SOMAR)

In this chapter we look at an application of the LaPS scheme. We construct an architecture, which allows for social media advertising while preserving the privacy of the users. The guarantees of privacy and verifiability are derived from the underlying PSA scheme.
We proceed by first explaining the problem context within social media advertising, i.e. the challenges of *affiliate marketing* in its current form, and why existing solutions from the field of *Online Behavioral Advertising* are not applicable or insufficient. Then, we introduce the details of our architecture SOMAR. Note that SOMAR is formulated in terms of generic building blocks and it can be thus instantiated based on the application's specific requirements. Finally, we present an instantiation of the architecture using a LaPS-instance and discuss experimental results from our implementation, which show evidence of the practicality of our construction. The content of this chapter was presented in [BGZ17a; BGZ17b].

4.1 Affiliate Marketing Model

Social media has created a new way for businesses and brands to advertise their products on the Internet. A study in 2011 revealed that 74% of consumers based their buying decision on social media content [Ben11] - it has become a multi-billion dollar business. At the same time, the amount and sensitivity of personal data that is being collected along the way has grown substantially and increased the associated privacy risks for end users.
Influencers are individuals with a significant following on their social media profiles [TBB10; Lan14; Hit15; Bow09], e.g. 3.4 million people "like" *Pamela Reif*'s posts on Instagram [Rei] and *Estee Lalonde* has 1.2 million subscribers[1], who watch her videos on YouTube [Lal]. Advertisers seek to leverage the popularity of such influencers in order to promote products to their followers, who are typically between 15 and 30 years old[2] and are also referred to as *millennials*. This particular group of people is an especially attractive target group as they constitute "*about 80 million people who spend by some accounts over one trillion dollars per year*" [Rad15]. Therefore, product merchants and their marketing teams have increasingly focused on social

[1] Numbers as of Feb 2018.

[2] There exist several notions that put the age limits in a slightly different range - this is the common intersection [Rap14; RH15; Deu16; Gol17].

media advertising as a means to reach this target group.
While there are numerous ways of turning likes and follows into profit [PMF12; KM12; Ma15], they are generally kept away from the public eye, i.e. the involved processes and agreements remain unclear or hidden. However, considering that they commonly involve the processing of privacy-sensitive user data, this lack of transparency directly translates into privacy risks for the end user. On the other hand, their data is highly valuable to advertisers [Tuc14] as it allows for precise characterization of the target group, therefore the data hunger of the advertisement industry will only continue growing. The revelations around how Facebook data from 50 million users was utilized for "*psychographic targeting*" [Val18] by the firm Cambridge Analytica is a recent example of this development [RCC18; CGH18; Val18].
We are concerned with a particularly interesting marketing model, called *affiliate marketing*. Here, influencers advertise certain merchants' products to their social media followers in return for receiving a share of the generated revenue [KM12; Cab17; FC01]. We discuss the associated privacy risks and how to solve the resulting problems.
The remainder of this chapter is organized as follows. In Section 4.2 we give an overview of existing solutions for traditional digital advertising. To the best of our knowledge, this is the first solution for privacy-preserving advertising in the particular context of social media marketing. We discuss the privacy risks related to the affiliate marketing model as well as existing trust assumptions between influencers and merchants as our problem statement. Subsequently, we propose the SOMAR architecture as a solution and discuss its building blocks (Section 4.3). Finally, we analyze the efficiency of our solution (Section 4.4).

4.2 Related Work

Most existing solutions address privacy concerns around digital advertising on a *general* level, i.e. to the best of our knowledge there are no other propositions specifically designed for social media advertising. In fact, a significant body of related work focuses on *Online Behavioral Advertising* (OBA) or *targeted* advertising [BRT11; Liu+13; LS16; Men+16], where through consistent tracking and evaluation of user's online browsing behavior, certain advertisements can be selected for an individual user that are deemed personally interesting or relevant to her.
In practice, tools like persistent cookies save the activities of a user across different visited websites (see [Aca+14] for an overview of tracking methods) and the collected information is reported back to the *broker*, who is the entity selecting the set of ads that are displayed on certain websites based on the extracted preferences of their audiences. The *publisher* is essentially the party representing the website where ads are placed, and the *advertiser* is the brand[3] or company providing ads for their products. Note that these instances can be far more nuanced in practice, e.g. a creative agency creating an ad would be distinguished from the marketing team that determines its placement, however for the purposes of OBA analysis as discussed above, both would be considered part of the advertiser party.

[3]In the following we use the terms "brand" and "merchant" interchangeably. In the traditional advertising model, they would correspond to the "advertiser".

The goal of privacy-preserving OBA solutions is to conceal every individual user's preferences while still providing targeted ads. One may consider applying these concepts to social media advertising by defining the influencer as a combination of the broker and the publisher, since he selects and displays ads on his own site. Finally, the brand would be viewed as the advertiser. The overall objective then translates into hiding any individual user's personal information, e.g. age and gender, while still allowing the influencer to compute statistics over his followers' attributes. The latter is essential to the influencer in order to shape a social media marketing strategy that is suited to his audience.
In general, the literature around privacy-preserving OBA and targeted advertising can be divided into *client-side* solutions, solutions that are deployed on an additional layer that is best described as a "*middleware*" layer, and finally *server-side* solutions. In the remainder of this section we discuss a selection of representative solutions from each category and their limitations when deployed in the social media advertising model.

Client-side OBA solutions. Solutions that aim to keep all sensitive data at the user's end, where each user locally executes the algorithms that select relevant ads, are considered client-side solutions [Tou+10; HHB10; GCF11; NAB11; DFL14]. Regardless of how user privacy is concretely achieved, the fact that the advertiser is supposed to gain no information about the users at all, is what makes these solutions (see [BRT11] for an overview of client-side solutions) problematic in the social media setting: the sacrifice in functionality would significantly affect the influencer who relies on user data as his main source of information - social media advertising in general heavily relies on the availability of user information. We discuss concrete examples next.
In *Privad* [GCF11] each user runs software that locally determines relevant ads based on the user's profile and an additional party, called the *dealer*, serves as a proxy for all user data that is anonymized before being sent to the broker. In this step, the authors distinguish between user *interests* and *demographics* with sensitive and non-sensitive attributes, e.g. age is sensitive while gender and location is considered non-sensitive. The dealer thus removes any sensitive information from the communicated user data and subsequently subscribes to ads from the broker that correspond to the set of non-sensitive attributes, which combined with broad demographic data determine a set of interest groups. This grouping is also essential in allowing reasonable efficiency, since each determined interest attribute results in a separate communication channel between the broker and the respective users. Hence, the additional demographic filter reduces the ad load and improves bandwidth efficiency.
In the social media setting, full user anonymity would lead to hiding any information that is collected during a user's purchase on the merchant's website. Furthermore, user groups would essentially have to subscribe to a subset of ads on the influencer's site that correspond to certain product categories based on user interests. The latter would be achieved through the user's computation of the type of ads that she would like to see and the dealer would send (anonymized) requests to the influencer with the resulting user interests.

Although anonymization preserves user privacy, this approach may be limited from a practical point of view: for instance, advertising products that a given user is currently unaware of or not yet interested in would become impossible as in the Privad model a user can only express interest in product types that she is already interested in. This fact inhibits the sale of new products which the customer finds through social media advertising in the first place[4]. Additionally, the anonymization of user information prevents the influencer from learning the attributes of his "average follower", as noted before. In contrast, our architecture does not have these shortcomings since we provide user privacy by using privacy-preserving aggregation rather than anonymization. Furthermore, we consider *all* individual user information to be sensitive. In short, our solution does not create a trade-off between privacy and functionality but provides both independently.
On the other hand, Privad enables client-side fraud detection by utilizing the dealer as a monitor of the users' click and view behavior, who reports suspicious activities to the broker. This concept could be implemented in our setting by adding a trusted third party, which monitors the user's behavior on the merchant's website and communicates to the influencer, where applicable. SOMAR however provides cryptographic proofs of correctness without the need for a trusted third party. Note that these proofs allow for the detection of both fraudulent merchants as well as fraudulent influencers while in above model only client-side fraud is detected.
In the *Adnostic* [Tou+10] system a set of random advertisements is preloaded, the user selects relevant ads locally and the user's choice, i.e. her views and clicks, are not revealed to the broker or the advertiser. Somewhat similar to our architecture, cryptographic tools are utilized in order to allow the broker to charge publishers and advertisers correctly: during defined billing cycles billing-relevant data is encrypted into homomorphic ciphertexts, equipped with a zero-knowledge proof of accuracy, and sent to a trusted third party, which at the end of a period decrypts the aggregated result and relays it to the broker. Green *et al.* [GLM16] additionally improve the scheme by utilizing cryptographic voting schemes, which results in higher scalability. This type of system could be used in our problem setting in order to eliminate trust assumptions between merchant and influencer, however this would require an additional trusted third party compared to our architecture.

Middleware OBA solution. An example for a middleware kind solution is *P3* [NAB11] that sits between users and advertisers and concentrates on recommendation-based services by locally serving targeted ads to users without revealing the users' behavior to the advertisers. In our context, one may formulate an aggregating party as a middleware layer between users and the influencer, which preserves user privacy while serving aggregate data to the influencer. In contrast, our solution achieves the same goal *without* the need for an additional middleware layer.

[4] A prominent example for such products is the so-called "fidget spinner", whose popularity is largely attributed to social media, e.g. YouTube videos [Bul17].

Server-side OBA solution. *ObliviAd* [Bac+12] is a server-side solution, which places a trusted hardware device at the broker's end. The authors call this device a *secure coprocessor* (SC). It processes encrypted user preferences in order to retrieve ads from the broker using oblivious RAM techniques, which ensures that the users' preferences remain hidden. Similar to Adnostic, encrypted tokens are accumulated by the SC, which decrypts and mixes the results for privacy-preserving billing. One may adopt this solution by using a trusted hardware device like the SC at the influencer's end in order to allow for privacy-preserving aggregation. SOMAR yields the same outcome without an additional hardware device.

The *LSBS* schemes [Her+14] solve the problem of sharing the location of users with other groups of users, i.e. their friends, without revealing user location information to the service provider. The schemes that use identity-based encryption also provide privacy-preserving aggregate statistics to the service provider, i.e. where he can compute aggregate statistics over user locations in a privacy-preserving manner. While this problem appears similar to our aggregation task within social media advertising, LSBS provides aggregation of *individual* user information, e.g. the number of times a particular user appeared at a certain location, as opposed to aggregation of data across *all* users as is required in our setting. Therefore, we solve a conceptually different problem. Moreover, our architecture appears[5] to be several orders of magnitude more efficient in terms of runtime and bandwidth efficiency.

In summary, although some of the discussed solutions can be applied in order to provide privacy-preserving social media advertising, they all come at the cost of functionality detrimental to the advertisers' interests. Additionally, existing solutions around OBA typically do not target a formal privacy guarantee such as DP. Therefore, their application to our problem would yield aggregated data that is susceptible to DP-attacks. Furthermore, none of the solutions were able to eliminate the trust assumption between merchants and influencers without the need for a trusted third party.
Looking at related work in the more general scope of computing statistics over distributed privacy-sensitive user data, i.e. not necessarily only when applied to digital advertisements, we also find a number of works [Che+12; CAF13] that rely on trusted third parties. As pointed out previously, this is disadvantageous compared to our solution as it would require introducing an additional party between users, merchant and influencer. The same issue arises with solutions like *Prio* [CGB17], where a set of dedicated servers obliviously compute aggregates of user data, which guarantees privacy as long as at least one server is honest. Note that this system does not provide privacy on a DP level in its initial form but can be extended to such guarantee [CGB17, Section 7].
Bilogrevic *et al.* [Bil+14] on the other hand, introduce an aggregation model that does not rely on a trusted third party and achieves DP of the end user data. Concretely, the data *aggregator* is defined as an external party that executes the aggregation but is not trusted by the users. Their model also

[5] The authors [Her+14] use a different platform, therefore a direct comparison is not possible. However, broadcast encryption is known to be a less efficient primitive than the combination of single public-key encryption and aggregation operations.

includes a *customer* party, which does not communicate to the users and instead formulates queries to the aggregator in order to retrieve certain aggregates. Indeed this system could be applied to social media advertising by defining the merchant as the aggregator and viewing the influencer as the customer. However, their system does not account for verifiability, i.e. the influencer still has to place trust in the merchant. In contrast, our solution provides verifiability through the use of proofs of correctness. Observe that due to the modular design of our construction, this functionality can be removed if it is not needed for a given application. In that case, one may deploy an aggregation model like Bilogrevic *et al.*'s [Bil+14] within our architecture. Hence, one may view SOMAR as a generic framework for aggregation models.

4.3 Providing Privacy and Proofs of Correctness: SOMAR Architecture

Figure 4.1 illustrates the following scenario: I is an Instagram influencer with a large group of followers $\{U_i\}_{i=1}^{F}$. Note that in practice also metrics other than number of followers are considered when defining an influencer, e.g. number of likes (see e.g. [Gou14]). Suppose, his posts are around food and cooking, which attracts the interest of brand S, who is a meal kit subscription service. Therefore, S sends I a free meal kit per month and in return I showcases S's product on his account together with a link where users can become new subscribers. When a certain subset of I's followers $\{U_i\}_{i=1}^{N}$, where $N \leq F$, are indeed influenced into subscribing, they click on the provided link and complete the purchase on S's website. Up to this point, the purchasing process is rather intuitive and does arguably not impose any additional privacy risks compared to a regular purchase on S's website.

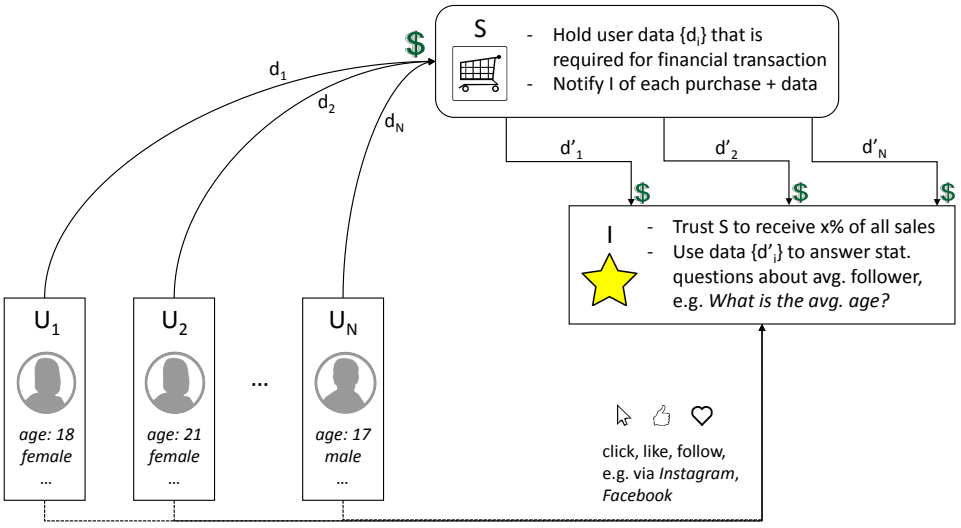

FIGURE 4.1: Current social media marketing model,where S sponsors I to present products to social media users $\{U_i\}$ [BGZ17a; BGZ17b].

However, in the affiliate marketing model, I additionally receives a certain percentage of the sales revenue from S, which was generated from the product's advertisement on I's page. More specifically, S forwards *each and every* purchase confirmation i - containing both the price and all personal data provided by U_i. Therefore, U_i's data is not only collected by S in order to facilitate the financial transaction but is also left with I by extension and likely without the knowledge of the user.
For the user, this results in increased privacy concerns, especially with respect to risk of data breaches as her data is stored on both S's and I's servers. In fact, taking into account that social media users typically follow more than one influencer and considering their respective agreements with brands like S, this makes the user's data more vulnerable.
At the same time, I and S are faced with a different set of problems in this situation: on the one hand, I cannot be sure that the data provided by S is complete and accurate, i.e. that indeed $d'_i = d_i$ for all $i \in \{1, \dots, N\}$ in Figure 4.1. On the other hand, S has no way of proving his honest behavior. Forwarding individual purchase confirmations gives I a rough idea of the generated revenue but it is by no means a guarantee as this information could be modified by S. Observe that ultimately, I is interested in *aggregate* data, e.g. the total sales revenue, average age of his followers, etc. He is looking to understand the attributes of his *average follower*, which allow him to shape his content and marketing strategy accordingly. Therefore, the data $\{d'_i\}$ is provided in a somewhat inappropriate form, as it gives more details than I needs while not giving any accuracy guarantees.

4.3.1 Building Blocks

SOMAR aims to solve the previously detailed problems and it provides the means for privacy-preserving social media advertising with verifiable data aggregation. Our architecture is based on aggregator unforgeable PSA schemes (see Section 3.3.2.2) in order to ensure end user data privacy and to compute proofs of correctness for the generated data aggregates.
Figure 4.2 shows how the building blocks of SOMAR are combined in our architecture. Concretely, each user U_i implements some DP-mechanism $\mathcal{M}$, which transforms the user's raw data d_i into a noisy version x_i, i.e. $x_i := \mathcal{M}(d_i)$. She also implements an encryption module Enc_i that encrypts x_i into a ciphertext c_i, which constitutes part of the user's final output (c_i, σ_i). The combination of privacy mechanism and encryption can be thought of as the components of the NoisyEnc routine in a PSA scheme as in Definition 15. The user's final output is completed by an authentication[6] *tag* σ_i that is computed by the authentication module Auth_i and allows for the verification of the overall aggregate result.
An aggregator unforgeable PSA scheme (see Section 3.3.2.2) inherently provides exactly the above set of functionalities and therefore it serves as the underlying structure for our architecture. On a high level, the influencer takes the role of the aggregator: he receives the noisy user data in the form of ciphertexts, aggregates it and retrieves the noisy aggregate x_{agg} using a decryption routine Dec. Finally, this result can be verified together with a proof of correctness σ_{agg} and a public verification key vk using the public

[6]Note that we refer to authentication as a means for data integrity and not in the sense of authentication of identities.

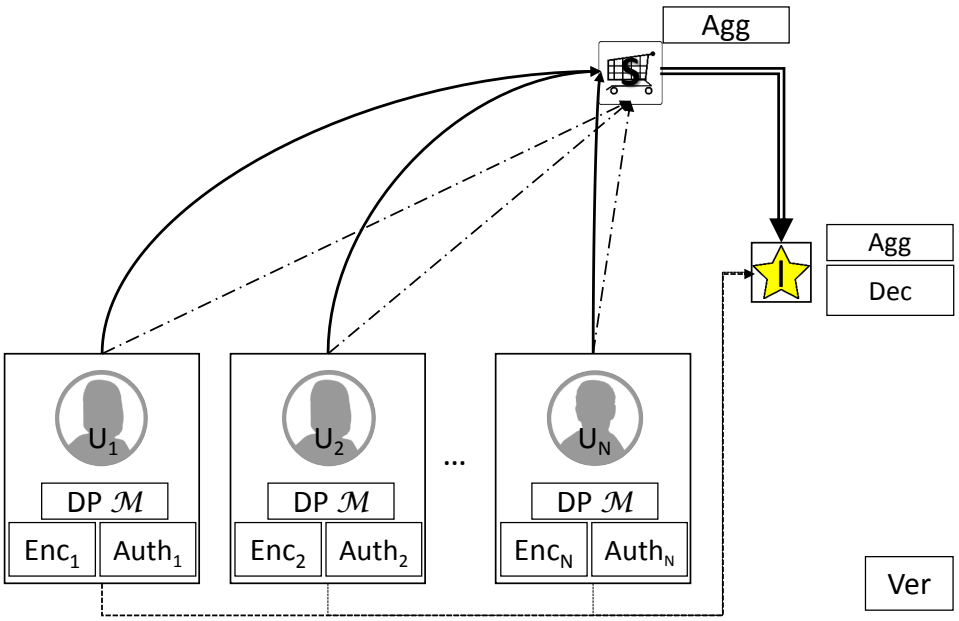

FIGURE 4.2: Structure of SOMAR [BGZ17a; BGZ17b].

verification module Ver, where the proof of correctness is computed by aggregating the individual users' authentication tags σ_i.
Therefore, each user implements privacy mechanism $\mathcal{M}$, encryption module Enc_i with an individual secret encryption key and the authentication module Auth_i with an individual secret authentication key. Merchant S implements the aggregation routine Agg, which essentially constitutes the execution of the desired aggregation function f_{agg}. Finally, influencer I also aggregates using Agg and decrypts using Dec and his secret decryption key.
In SOMAR we treat each module as an individual building block (see Figure 4.2). This allows for the instantiation of each module based on the individual needs of the application, e.g. the plug-and-play deployment of a certain privacy mechanism that provides a desired privacy-accuracy trade-off without affecting the other building blocks.
In contrast to the standard model (see Figure 4.1), privacy-sensitive user data is only sent from users $\{U_i\}_{i=1}^N$ to merchant S (straight-line edges) and S only shares DP-aggregate data with I (double-edged arrow). Assuming that each user has access to some trusted device, e.g. on a computer with a trusted browser extension, that invokes the user operations and stores secret keys, the performed actions are transparent to the human user (for simplification, we still refer to the "user" executing that action). These user operations only communicate encrypted data, i.e. ciphertexts and authentication tags, to other parties (dashed edges).
We detail the individual computations of each party with respect to the defined building blocks next. Figure 4.3 shows an overview.

User. After user U_i has clicked the link in I's post, she is directed to S's website, where she makes a purchase - this is where our solution kicks in:

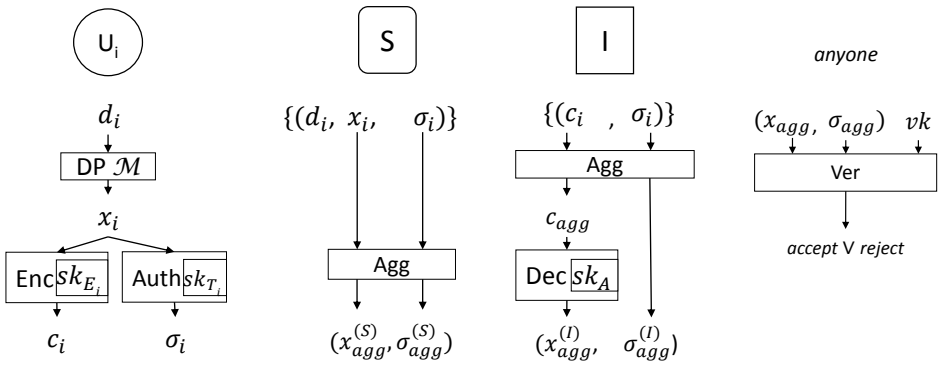

FIGURE 4.3: Detailed computations
(a) *User*, (b) *Merchant*, (c) *Influencer*, (d) *Verification* [BGZ17a].

U_i applies privacy mechanism $\mathcal{M}$ to her data d_i, e.g. age, gender[7], which generates the noisy value x_i. Encrypting x_i using Enc yields ciphertext c_i as $c_i := \mathsf{Enc}(sk_{E_i}, x_i)$, where sk_{E_i} is U_i's secret encryption key. She uses the authentication routine Auth with her authentication key sk_{T_i} in order to generate the authentication tag σ_i as $\sigma_i := \mathsf{Auth}(sk_{T_i}, x_i)$. Finally, each user holds the tuple $(d_i, x_i, sk_{E_i}, sk_{T_i}, c_i, \sigma_i)$ and shares (d_i, x_i, σ_i) with S and (c_i, σ_i) with I, respectively.

Both S and I receive N sets of user data, which they simultaneously accumulate. S's input is the users' raw and noisy data and authentication tags $\{(d_i, x_i, \sigma_i)\}_{i=1}^N$ and I receives encrypted noisy user data and authentication tags $\{(c_i, \sigma_i)\}_{i=1}^N$.

Merchant. S performs the desired aggregation function f_{agg} on the users' noisy values $x_{agg}^{(S)} = f_{agg}(\{x_i\}_{i=1}^N)$. He also aggregates the individual authentication tags σ_i in order to compute his proof of correctness $\sigma_{agg}^{(S)} = f_{agg}(\{\sigma_i\}_{i=1}^N)$. Eventually, S holds $(x_{agg}^{(S)}, \sigma_{agg}^{(S)})$.

Influencer. I receives $\{(c_i, \sigma_i)\}_{i=1}^N$ as input and aggregates the ciphertexts, which yields the aggregated ciphertext c_{agg}. He retrieves the aggregate result by decrypting c_{agg} using his decryption capability sk_A, i.e. $x_{agg} = \mathsf{Dec}(sk_A, c_{agg})$. Finally, he computes his proof of correctness $\sigma_{agg}^{(I)}$ as $\sigma_{agg}^{(I)} = f_{agg}(\{\sigma_i\}_{i=1}^N)$. Note that as long as the underlying PSA scheme is aggregator oblivious and since I is the aggregator in this setting, he can only decrypt the aggregated ciphertext c_{agg}. More concretely, he cannot decrypt any *individual* user ciphertext and no information about any individual user is leaked to I. He now holds $(x_{agg}^{(I)}, \sigma_{agg}^{(I)})$, i.e. the noisy aggregate output and the proof of correctness.

Verification. When S provides I with $x_{agg}^{(S)}$ and claims that this value corresponds to the noisy aggregate of user data, I can now check whether $x_{agg}^{(S)} \stackrel{?}{=} x_{agg}^{(I)}$. If either S or I were dishonest, these values will be different.

[7] Information like age and gender is typically optional when filling out a form for online purchases. Generally, one may think of any privacy-sensitive information that is collected during the purchasing process.

In that case, the public routine Ver can be used for verification: providing x_{agg} together with the proof of correctness σ_{agg}, Ver will only return accept for the correct input and reject, otherwise. It will always fail for incorrect inputs based on the aggregator unforgeability guarantee of the underlying PSA scheme.

Remark 4. *Note that the verification key vk has to be published by the party that generates the input authentication tags, i.e. the user. Consequently, the party that attempts to get their value verified, e.g. S, cannot provide a fake verification key to the verifier, e.g. I, and thereby falsely obtain acceptance. In practice, this could be achieved by establishing a public register within the architecture, where any new verification key is published once it is generated.*

Influencer I can also use the generated proofs of correctness in order to prove accuracy of certain aggregate data to *other* merchants. For instance, merchants that look to promote their products to a target group with certain statistical attributes can verify that I's followers indeed have these attributes. I would simply present the aggregate data together with a proof of correctness and the interested merchants can verify it using the public verification routine.

Remark 5. *Compared to the social media marketing model as shown in Figure 4.1, our architecture requires each party to implement the modules as specified previously - in particular, S has to set up the* Agg *operation and I has to additionally implement the decryption module* Dec. *While providing end user privacy, this naturally incurs additional cost compared to before. However, note that this also serves S's and I's individual interests: on the one hand, merchant S benefits from their clients' increased privacy as added brand value and recognition and it minimizes the risk of future data compromises. On the other hand, the influencer I can now rely on correct aggregate data.*

4.3.2 SOMAR Instantiation

In order to achieve the two goals of our architecture of *user privacy* and *data reliability* in practice, we combine DP tools and encryption with proofs of correctness - namely through the use of aggregator unforgeable PSA schemes. Note that each of these components corresponds to individual steps in the underlying PSA scheme as in Definition 15 and constitutes a separate module in our architecture. The combination of DP-mechanism $\mathcal{M}$ and encryption module Enc naturally corresponds to NoisyEnc; the aggregation module Agg is the aggregation function f and together with the decryption module Dec, they constitute AggrDec.

Tags/Signature In our architecture users generate authentication tags that can be aggregated into a proof of correctness for the overall aggregate result. In practice, this can be implemented using a *homomorphic aggregate signature scheme* [TDB16]. Here, each authentication tag σ_i is produced as a signature of the noisy user value x_i - due to the homomorphic nature, it can be aggregated and the final output σ_{agg} corresponds to a valid signature of the underlying aggregated messages x_{agg}. Hence, the aggregate signature serves as the proof of correctness and our verification module Ver corresponds to the verification routine in the signature scheme.

Bundling an aggregator oblivious PSA scheme with proofs of correctness, for instance by using a homomorphic aggregate signature scheme, yields an aggregator unforgeable PSA scheme. By satisfying this notion the scheme provides all desired functionalities: the aggregate output, i.e. the noisy aggregate user data provided to influencer I, does not leak any information about any individual user U_i and can be verified using an aggregate proof of correctness. Hence, I is not forced to place trust in S nor vice-versa.

Using LaPS. Existing proposals for aggregator unforgeable PSA schemes [Leo+15; Emu17] are based on the DDH-assumption and consequently have the same limitations as Shi *et al.*'s [Shi+11] scheme (see Section 3.4). In contrast, our LaPS scheme can be extended to aggregator unforgeability in a straightforward way by combining it with a homomorphic aggregate signature scheme, as detailed above.
Jing's [Jin14] HAS scheme is a particularly good fit as it is a lattice-based homomorphic aggregate signature scheme and therefore maintains conjectured post-quantum security. It also performs best among other homomorphic aggregate signature schemes in terms of efficiency [TDB16] and thus aligns well with our LaPS scheme with respect to performance. Hence, taking LaPS as the underlying aggregator oblivious PSA scheme and for instance using the example instantiation that we have presented in Section 3.7, lets us define module $\mathcal{M}$ as the discrete Laplace mechanism, module Enc as a combination of A-LWE- and BGV-based encryption, and Dec as the corresponding decryption routine. Note that for this LaPS instance Agg corresponds to the sum function. Finally, adding the HAS scheme yields aggregator unforgeability and completes the SOMAR instantiation, where the HAS signature routine constitutes the Auth module and the HAS verification routine represents the Ver module.

4.4 Experimental Results

We base our runtime estimations for all parties, i.e. user U_i, merchant S, and influencer I, on an instantiation of SOMAR with our PSA scheme LaPS when combined with the homomorphic aggregate signature scheme HAS due to Jing [Jin14]. In fact, we use the exact form of LaPS that we have presented in Section 3.7, where a reduced version of the BGV scheme is the additively homomorphic element and the discrete Laplace mechanism is used as the privacy mechanism, i.e. the LaPS instance according to Definition 25. Consequently, the desired aggregation is the sum.

	$\mathcal{M}$	Enc	Agg	Dec
User	3.72		-	-
Merchant	-	-	0.02	-
Influencer	-	-	1.96	

TABLE 4.1: Runtime results in ms for 1000 users, 80-bit security level and $\approx 2^{16}$ plaintext space [BGZ17a].

Note that by combining certain building blocks in our architecture we get the subroutines of the LaPS scheme. Therefore, we can reuse the runtime measurements of our LaPS scheme (see Section 3.8.2) for the runtimes of

privacy mechanism and encryption and aggregation and decryption in this SOMAR instantiation. We estimate the runtimes for the remaining building blocks of authentication tagging and verification based on results from El Bansarkhani and Buchmann [EB13].

	Auth	Ver
User	15.4	3.1
Merchant	-	
Influencer	-	

TABLE 4.2: Estimate runtimes in ms for 1000 users, 80-bit security level and $\approx 2^{16}$ plaintext space based on [EB13].

Our results are summarized in Tables 4.1 and 4.2 and they are shown per operation and executing party. Recall that the verification routine is accessible to anyone. Note that we separate our runtime results from the estimations based on El Bansarkhani and Buchmann's [EB13] results as their implementation was conducted on a different platform (see [EB13, Section 5]) than ours[8].

We assume $N = 1000$ participants and target a bit-security level of 80 bits with a plaintext space of roughly 2^{16} (as in Section 3.8.2). In the following, we discuss the basis for our estimations and subsequently detail the derivation of our results for the individual building blocks of our architecture and their combination.

According to our results in Table 3.4, the execution of the NoisyEnc routine within our LaPS scheme takes 3.72 ms. This includes both invoking the privacy mechanism, i.e. the discrete Laplace mechanism as $\mathcal{M}$, and encrypting, i.e. executing Enc, hence:

$$\mathcal{M} + \mathsf{Enc} \Rightarrow 3.72\ ms. \tag{4.1}$$

The user generates an authentication tag σ_i for their noisy value x_i by *signing* it using the Sign routine of the HAS scheme [Jin14]. The latter is defined as computing a homomorphic hash value of x_i and running the pre-image sampling algorithm SamplePre, which outputs the signature.

The signature routine in Gentry *et al.*'s [GPV08] GPV signature scheme has the same structure except that a general hash function is used instead of a homomorphic one. El Bansarkhani and Buchmann [EB13] improve over the original GPV scheme and present a particularly efficient implementation of this signature scheme using Micciancio and Peikert's [MP12] trapdoor construction. Since the signature routines are thus comparable, we use El Bansarkhani and Buchmann's [EB13] experimental results as the basis for our runtime estimation of the tag generation in the Auth routine.

The homomorphic hash according to [Jin14] is computed by first determining a set of common vectors $\{\alpha_j\}$, which are (regular) hash values of some public values $\{j\}$, and then multiplying each vector $\{\alpha_j\}$ with the input, say message $\mathbf{m}$, i.e. $\{\langle \alpha_j, \mathbf{m} \rangle\}$. The final homomorphic hash of $\mathbf{m}$ is the combination of these results into a column vector. Regarding the computational effort, there is one additional operation compared to the GPV signature scheme, where a set of scalar products between two vectors is computed.

[8] Note that our platform details are as detailed in Section 3.8.2.

El Bansarkhani and Buchmann [EB13] break down the computation time of the signature routine into its individual parts: their results indicate that the computation of such scalar products occupies about 10% of the total computation time. Therefore, we estimate that the execution of our Auth module amounts to roughly 110% of the runtime of the GPV signature routine as implemented in [EB13]. Their runtime result is 14 ms for parameters $n = 256$ and $k = 27$, which yields 80-bit security, hence

$$\mathsf{Auth} \Rightarrow 14\ ms \cdot 1.1 = 15.4\ ms. \tag{4.2}$$

Note that the implementation in [EB13] is set in the ring setting, which is in line with our LaPS instance.
The targeted aggregation operation here is the sum, therefore Agg computes the sum of the input values. Our runtime measurements indicate for number of participants and plaintext space as before:

$$\mathsf{Agg} \Rightarrow 0.02\ ms. \tag{4.3}$$

Observe that this result is also applicable to the summation of ciphertexts and authentication tags as the individual additions of vector elements can be entirely parallelized. AggrDec in the LaPS scheme contains both the aggregation of the input ciphertexts, i.e. Agg, and their decryption, i.e. Dec. Therefore, aggregation and decryption take

$$\mathsf{Agg} + \mathsf{Dec} \Rightarrow 1.96\ ms. \tag{4.4}$$

With these results at hand, we now estimate the overall runtime for each involved party:

User. The user applies the privacy mechanism $\mathcal{M}$, which results in the noisy value x_i, encrypts using Enc, and authenticates using Auth. Assuming that x_i can be accessed from the authentication routine (which considering that it is all part of one "user routine" is a reasonable assumption), the user's overall runtime corresponds to the sum of the runtimes for these individual building blocks, which based on results (4.1) and (4.2) is practical.

Merchant. The aggregation using Agg is the only task of the merchant, which takes 0.02 ms according to (4.3). Note that the aggregation of ciphertexts and authentication tags can be optimized into one operation using parallelization.

Influencer. The influencer aggregates both input ciphertexts and authentication tags, which as before can be accounted for as one aggregation operation, and subsequently decrypts the aggregate ciphertext. Therefore, his total runtime corresponds to result (4.4), i.e. 1.96 ms.

Verification. Analogous to the authentication routine, the verification Ver corresponds to the verification method in the HAS scheme [Jin14]. This algorithm is in fact identical to the Verify routine in the GPV signature scheme [GPV08] except for the hashing step as detailed before. Therefore,

we again take El Bansarkhani and Buchmann's [EB13] runtime result but this time for the verification, which equals 1.7 ms, and add 1.4 ms for the additional operation due to the computation of a homomorphic hash instead of a general one. The latter corresponds to the estimate for computing scalar products based on the runtime of the signing routine in the GPV scheme. Hence,

$$\mathsf{Verify} \Rightarrow 1.7\ ms + 1.4\ ms = 3.1\ ms. \tag{4.5}$$

As stated previously, the verification routine can be invoked by any party that wants to verify some aggregate data.

Chapter 5

Summary

In this chapter we summarize our findings and discuss potential future research directions.

5.1 Conclusions

We have applied LWE-based encryption to the problem of privacy-friendly data aggregation and our results have shown that this application indeed leads to improvements in performance. We have also provided a solution for privacy-preserving social media marketing and thereby resolved a problem that has not been considered before. We summarize our contributions in the following.

Lattice-based PSA. We introduced a new lattice-based PSA scheme called LaPS, which improves over previous schemes in several ways: we resolved the main limitation from Shi *et al.*'s [Shi+11] scheme with regards to the plaintext size restriction based on the high bandwidth efficiency of the underlying LWE-based construction. We were also able to remove previously required assumptions, i.e. the encrypt-once model, and provide an overall stronger security guarantee due to conjectured post-quantum hardness of worst-case lattice problems, which constitutes our underlying assumption. Moreover, we achieved significant performance improvements in terms of both plaintext space (roughly 66000 times larger) and runtime (about 150 times faster decryption) compared to Shi *et al.*'s [Shi+11] PSA scheme.

Privacy-preserving social media marketing. We also presented SOMAR, which is a privacy-friendly social media marketing architecture that benefits from the DP- and security guarantees of PSA schemes. In fact, we are the first to consider this problem. We showed how instantiating SOMAR with our LaPS scheme allows for an efficient solution that provides both end user data privacy and public verifiability of the produced data aggregates. Again, based on the underlying assumption we may conjecture post-quantum security and as our experimental results showed, our solution is practical.

Summarizing, with our schemes and constructions we were able to show that lattice-based and particularly LWE-based systems have intriguing properties that allow for efficient implementation in practice. In particular, privacy-preserving data aggregation, which has been the main focus of this work, can indeed be realized in an efficient manner when using an LWE-based construction.

It is our hope that our contributions to advancing the state-of-the-art in applying LWE-based schemes will inspire a more differentiated view of efficiency in the context of certain applications and will reinforce the consideration of LWE as a hardness assumption beyond the need for post-quantum hardness.

5.2 Future Work

The effort to establish compelling evidence for the unique suitability of LWE for certain (pre-quantum era) applications is a significant one and our contributions can only be understood as steps in that direction. Nevertheless, we can now formulate a set of potential future research directions that will help continue this effort. We start by listing a number of interesting follow-up questions that arose directly from our work on our PSA-scheme LaPS (Chapter 3) and our SOMAR architecture (Chapter 4) and end on a more general note around (A-)LWE-based systems.

- **Support richer statistics.** In our discussion of LaPS and PSA schemes in general we focused our view on the most basic case of aggregation, i.e. the sum. It would be valuable to support other operations using a lattice-based scheme, such as computing the aggregate product.

- **User failures.** Our construction currently assumes that all N participants submit an input. If only one user ciphertext is missing, the overall aggregation process has to start over. Therefore, allowing for dynamic joins and leaves would tolerate such user failures. We have indicated potential ways of realizing these in Section 3.9.

- **Aggregator unforgeability.** The notion of aggregator unforgeability (see Section 3.3.2.2) extends aggregator obliviousness in that it provides public verifiability of the aggregate result. Hence, when the aggregator is not trusted to accurately publish the computed result, this notion is appropriate. For our scheme LaPS, this could be achieved by combining it with a homomorphic aggregate signature scheme, e.g. the HAS scheme [Jin14] as detailed in the instantiation of our architecture SOMAR (see Section 4.3.2).

- **Guarantees for malicious users.** SOMAR is clearly a user-centric architecture. We assume that it is in the user's interest to ensure her data privacy and therefore she will not only accept the task of encrypting her data and communicating it to the other parties but also follow the protocol honestly. Therefore, an interesting problem would be to design schemes where the user is not trusted. For instance, one may consider a variation of SOMAR, where parties other than the user are (perhaps jointly) responsible for ensuring end user data privacy.

- **Avoid user-merchant collusion.** Similarly to the case of malicious users, when a merchant secretly agrees with users to provide false data, our architecture in its current form does not provide correctness guarantees to the influencer. This may be of interest in practice, especially when the merchant offers a financial incentive such as a discount.

- **Formal security model for SOMAR instantiation.** Our development of the SOMAR architecture is mainly based on the security notion of the underlying PSA scheme, i.e. aggregator unforgeability. In a more general treatment of the problem of privacy-preserving social media marketing, it would be worth considering a dedicated formal security (and privacy) model for this problem statement. Besides allowing for a rigorous evaluation of solutions, this may also expand the potential bases for appropriate instantiations beyond aggregator unforgeable PSA schemes.

- **Further applications of A-LWE.** Based on our contributions, it would be intriguing to explore more applications that can benefit from A-LWE as a hardness assumption on a general level, both from a security as well as an efficiency perspective. For example, realizing a computational fuzzy extractor based on A-LWE as opposed to traditional LWE (see e.g. [FMR13; Hut+17]).

Bibliography

[AAS14] Markku Antikainen, Tuomas Aura, and Mikko Särelä. "Denial-of-Service Attacks in Bloom-Filter-Based Forwarding". In: *IEEE/ ACM Trans. Netw.* 22.5 (2014), pp. 1463–1476. DOI: 10.1109/TNET.2013.2281614. URL: https://doi.org/10.1109/TNET.2013.2281614.

[Aca+14] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juárez, Arvind Narayanan, and Claudia Díaz. "The Web Never Forgets: Persistent Tracking Mechanisms in the Wild". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*. 2014, pp. 674–689. DOI: 10.1145/2660267.2660347. URL: http://doi.acm.org/10.1145/2660267.2660347.

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence". In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*. 1997, pp. 284–293. DOI: 10.1145/258533.258604. URL: http://doi.acm.org/10.1145/258533.258604.

[AG11] Sanjeev Arora and Rong Ge. "New Algorithms for Learning in Presence of Errors". In: *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*. 2011, pp. 403–415. DOI: 10.1007/978-3-642-22006-7_34. URL: https://doi.org/10.1007/978-3-642-22006-7_34.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. "Simultaneous Hardcore Bits and Cryptography against Memory Attacks". In: *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*. 2009, pp. 474–495. DOI: 10.1007/978-3-642-00457-5_28. URL: https://doi.org/10.1007/978-3-642-00457-5_28.

[Ajt05] Miklós Ajtai. "Representing hard lattices with O(n log n) bits". In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*. 2005, pp. 94–103. DOI: 10.1145/1060590.1060604. URL: http://doi.acm.org/10.1145/1060590.1060604.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. 1996, pp. 99–108. DOI: 10.

1145/237814.237838. URL: http://doi.acm.org/10.1145/237814.237838.

[Alb+14] Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. "Lazy Modulus Switching for the BKW Algorithm on LWE". In: *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*. 2014, pp. 429–445. DOI: 10.1007/978-3-642-54631-0_25. URL: https://doi.org/10.1007/978-3-642-54631-0_25.

[App] *Apple Differential Privacy Technical Overview*. Tech. rep. Apple Inc.

[App+09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35. URL: https://doi.org/10.1007/978-3-642-03356-8_35.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. "On the concrete hardness of Learning with Errors". In: *J. Math. Cryptol.* 9.3 (2015), pp. 169–203. DOI: 10.1515/jmc-2015-0016. URL: https://eprint.iacr.org/2015/046.pdf.

[Ash+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE". In: *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. 2012, pp. 483–501. DOI: 10.1007/978-3-642-29011-4_29. URL: https://doi.org/10.1007/978-3-642-29011-4_29.

[Ass+16] Julian Assange, Jacob Appelbaum, Andy Muller-Maguhn, and Jrmie Zimmermann. *Cypherpunks: Freedom and the Future of the Internet*. OR books, 2016.

[Bab85] László Babai. "On Lovász' Lattice Reduction and the Nearest Lattice Point Problem (Shortened Version)". In: *STACS 85, 2nd Symposium of Theoretical Aspects of Computer Science, Saarbrücken, Germany, January 3-5, 1985, Proceedings*. 1985, pp. 13–20. DOI: 10.1007/BFb0023990. URL: https://doi.org/10.1007/BFb0023990.

[Bac+12] Michael Backes, Aniket Kate, Matteo Maffei, and Kim Pecina. "ObliviAd: Provably Secure and Practical Online Behavioral Advertising". In: *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*. 2012, pp. 257–271. DOI: 10.1109/SP.2012.25. URL: https://doi.org/10.1109/SP.2012.25.

[Bau+16] Bela Bauer, Dave Wecker, Andrew J. Millis, Matthew B. Hastings, and Matthias Troyer. "Hybrid Quantum-Classical Approach to Correlated Materials". In: *Phys. Rev. X* 6 (3 2016), p. 031045. DOI: 10.1103/PhysRevX.6.031045. URL: https://link.aps.org/doi/10.1103/PhysRevX.6.031045.

[Ben11] Shea Bennett. *The Business Of Social Media*. http://www.adweek.com/digital/business-social-media/?red=at. 2011, *Retrieved Aug 9 2017*.

[Ber09] Daniel J. Bernstein. "Introduction to post-quantum cryptography". In: *Post-quantum cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Springer Berlin Heidelberg, 2009, pp. 1–14. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*. 2012, pp. 309–325. DOI: 10.1145/2090236.2090262. URL: http://doi.acm.org/10.1145/2090236.2090262.

[BGZ17a] Daniela Becker, Jorge Guajardo, and Karl-Heinz Zimmermann. "SOMAR: Privacy-Preserving SOcial Media Advertising ARchitecture". In: *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society, Dallas, TX, USA, October 30 - November 3, 2017*. 2017, pp. 21–30. DOI: 10.1145/3139550.3139563. URL: http://doi.acm.org/10.1145/3139550.3139563.

[BGZ17b] Daniela Becker, Jorge Guajardo, and Karl-Heinz Zimmermann. "Towards a new privacy-preserving social media advertising architecture (invited position paper)". In: *2017 IEEE Conference on Communications and Network Security, CNS 2017, Las Vegas, NV, USA, October 9-11, 2017*. 2017, pp. 45–457. DOI: 10.1109/CNS.2017.8228712. URL: https://doi.org/10.1109/CNS.2017.8228712.

[BGZ18] Daniela Becker, Jorge Guajardo, and Karl-Heinz Zimmermann. "Revisiting Private Stream Aggregation: Lattice-Based PSA". In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, 18th February - 21st February 2018*. 2018.

[Bil+14] Igor Bilogrevic, Julien Freudiger, Emiliano De Cristofaro, and Ersin Uzun. "What's the Gist? Privacy-Preserving Aggregation of User Profiles". In: *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II*. 2014, pp. 128–145. DOI: 10.1007/978-3-319-11212-1_8. URL: https://doi.org/10.1007/978-3-319-11212-1_8.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *J. ACM* 50.4 (2003), pp. 506–519. DOI: 10.1145/792538.792543. URL: http://doi.acm.org/10.1145/792538.792543.

[BL16] Daniel J. Bernstein and Tanja Lange. *Post-quantum cryptography for long-term security, PQCRYPTO ICT-645622 - presentation*. ISO 27 meeting, Tampa, USA, 2016. URL: http://pqcrypto.eu.org/slides/iso-liaison.pdf.

[Blu+05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. "Practical privacy: the SuLQ framework". In: *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*. 2005, pp. 128–138. DOI: 10.1145/1065167.1065184. URL: http://doi.acm.org/10.1145/1065167.1065184.

[Bow09] Chris Bowler. *Can Social Ads Do Better Than Display Ads?* http://faculty.cbpp.uaa.alaska.edu/afef/SIM-Razorfish.pdf. 2009, *Retrieved Aug 9 2017*.

[BR18] Abigail Beall and Matt Reynolds. *What are quantum computers and how do they work? WIRED explains*. http://www.wired.co.uk/article/quantum-computing-explained. 2018, *Retrieved Mar 20 2018*.

[Bra+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. "Classical hardness of learning with errors". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 2013, pp. 575–584. DOI: 10.1145/2488608.2488680. URL: http://doi.acm.org/10.1145/2488608.2488680.

[BRT11] Mikhail Bilenko, Matthew Richardson, and J Tsai. "Targeted, Not Tracked: Client-side Solutions for Privacy-Friendly Behavioral Advertising". In: *Fourth Hot Topics in Privacy Enhancing Technologies Symposium (HotPETS 2011)* (2011), pp. 1–20. URL: https://petsymposium.org/2011/papers/hotpets11-final3Bilenko.pdf.

[Bul17] Paul Bullock. *How is Digital Aiding The Rise of Fidget Spinners?* https://fastweb.media/articles/blog/how-is-digital-aiding-the-rise-of-the-fidget-spinner. 2017, *Retrieved Mar 20 2018*.

[Bur15] U.S. Census Bureau. *American FactFinder*. Tech. rep. 2015.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. "Efficient Fully Homomorphic Encryption from (Standard) LWE". In: *Electronic Colloquium on Computational Complexity (ECCC)* 18 (2011), p. 109. URL: http://eccc.hpi-web.de/report/2011/109.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages". In: *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*. 2011, pp. 505–524. DOI: 10.1007/978-3-642-22792-9_29. URL: https://doi.org/10.1007/978-3-642-22792-9_29.

[Cab17] Pauline Cabrera. *5 Creative Ways To Earn Money On Instagram*. http://www.twelveskip.com/guide/making-money/1228/ways-to-earn-money-on-instagram. 2017, *Retrieved Aug 9 2017*.

[CAF13] Ruichuan Chen, Istemi Ekin Akkus, and Paul Francis. "SplitX: High-Performance Private Analytics". In: *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*. 2013, pp. 315–326. ISBN: 9781450320566. DOI: 10.1145/2486001.2486013.

[CGB17] Henry Corrigan-Gibbs and Dan Boneh. "Prio: Private, Robust, and Scalable Computation of Aggregate Statistics." In: *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*. 2017, pp. 259–282.

[CGH18] Carole Cadwalladr and Emma Graham-Harrison. "Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach". In: *The Guardian* (2018, *Retrieved Mar 26 2018*).

[CGW14] Daniel Cabarcas, Florian Göpfert, and Patrick Weiden. "Provably secure LWE encryption with smallish uniform noise and secret". In: *ASIAPKC'14, Proceedings of the 2nd ACM Workshop on ASIA Public-Key Cryptography, June 3, 2014, Kyoto, Japan*. 2014, pp. 33–42. DOI: 10.1145/2600694.2600695. URL: http://doi.acm.org/10.1145/2600694.2600695.

[Che+12] Ruichuan Chen, Alexey Reznichenko, Paul Francis, and Johannes Gehrke. "Towards Statistical Queries over Distributed Private User Data". In: *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*. 2012, pp. 169–182.

[CN11] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*. 2011, pp. 1–20. DOI: 10.1007/978-3-642-25385-0_1. URL: https://doi.org/10.1007/978-3-642-25385-0_1.

[CSS12] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. "Privacy-Preserving Stream Aggregation with Fault Tolerance". In: *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, February 27-March 2, 2012, Revised Selected Papers*. 2012, pp. 200–214. DOI: 10.1007/978-3-642-32946-3_15. URL: https://doi.org/10.1007/978-3-642-32946-3_15.

[Dam+06] Ivan Damgård, Matthias Fitzi, Eike Kiltz, Jesper Buus Nielsen, and Tomas Toft. "Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*. 2006, pp. 285–304. DOI: 10.1007/11681878_15. URL: https://doi.org/10.1007/11681878_15.

[Dam+12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. "Multiparty Computation from Somewhat Homomorphic Encryption". In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. 2012, pp. 643–662. DOI: 10.1007/978-3-642-32009-5_38. URL: https://doi.org/10.1007/978-3-642-32009-5_38.

[Dam+13] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. "Practical Covertly Secure MPC for Dishonest Majority - Or: Breaking the SPDZ Limits". In: *Computer Security - ESORICS 2013 - 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings*. 2013, pp. 1–18. DOI: 10.1007/978-3-642-40203-6_1. URL: https://doi.org/10.1007/978-3-642-40203-6_1.

[Dan+13] George Danezis, Cédric Fournet, Markulf Kohlweiss, and Santiago Zanella Béguelin. "Smart meter aggregation via secret-sharing". In: *SEGS'13, Proceedings of the 2013 ACM Workshop on Smart Energy Grid Security, Co-located with CCS 2013, November 8, 2013, Berlin, Germany*. 2013, pp. 75–80. DOI: 10.1145/2516930.2516944. URL: http://doi.acm.org/10.1145/2516930.2516944.

[Deu16] Julia Deutsch. *Millennials Are Changing The World Of Advertising*. https://www.digitaldoughnut.com/articles/2016/may/millennials-are-changing-the-world-of-advertising. 2016, *Retrieved Aug 9 2017*.

[DFL14] Drew Davidson, Matt Fredrikson, and Benjamin Livshits. "MoRePriv: mobile OS support for application personalization and privacy". In: *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014*. 2014, pp. 236–245. DOI: 10.1145/2664243.2664266. URL: http://doi.acm.org/10.1145/2664243.2664266.

[DM13] Nico Döttling and Jörn Müller-Quade. "Lossy Codes and a New Variant of the Learning-With-Errors Problem". In: *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*. 2013, pp. 18–34. DOI: 10.1007/978-3-642-38348-9_2. URL: https://doi.org/10.1007/978-3-642-38348-9_2.

[DN03] Irit Dinur and Kobbi Nissim. "Revealing information while preserving privacy". In: *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*. 2003, pp. 202–210. DOI: 10.1145/773153.773173. URL: http://doi.acm.org/10.1145/773153.773173.

[DN04] Cynthia Dwork and Kobbi Nissim. "Privacy-Preserving Datamining on Vertically Partitioned Databases". In: *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*. 2004, pp. 528–544. DOI: 10.1007/978-3-540-28628-8_32. URL: https://doi.org/10.1007/978-3-540-28628-8_32.

[DR14] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407. DOI: 10.1561/0400000042. URL: https://doi.org/10.1561/0400000042.

[Dwo+06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*. 2006, pp. 265–284. DOI: 10.1007/11681878_14. URL: https://doi.org/10.1007/11681878_14.

[Dwo06] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*. 2006, pp. 1–12. DOI: 10.1007/11787006_1. URL: https://doi.org/10.1007/11787006_1.

[Dwo08] Cynthia Dwork. "Differential Privacy: A Survey of Results". In: *Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings*. 2008, pp. 1–19. DOI: 10.1007/978-3-540-79228-4_1. URL: https://doi.org/10.1007/978-3-540-79228-4_1.

[EB13] Rachid El Bansarkhani and Johannes A. Buchmann. "Improvement and Efficient Implementation of a Lattice-Based Signature Scheme". In: *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*. 2013, pp. 48–67. DOI: 10.1007/978-3-662-43414-7_3. URL: https://doi.org/10.1007/978-3-662-43414-7_3.

[EDB14] Rachid El Bansarkhani, Özgür Dagdelen, and Johannes A. Buchmann. "Augmented Learning with Errors: The Untapped Potential of the Error Term". In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 733. URL: http://eprint.iacr.org/2014/733.

[EDB15] Rachid El Bansarkhani, Özgür Dagdelen, and Johannes A. Buchmann. "Augmented Learning with Errors: The Untapped Potential of the Error Term". In: *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*. 2015, pp. 333–352. DOI: 10.1007/978-3-662-47854-7_20. URL: https://doi.org/10.1007/978-3-662-47854-7_20.

[Emu17] Keita Emura. "Privacy-Preserving Aggregation of Time-Series Data with Public Verifiability from Simple Assumptions". In: *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part II*. 2017, pp. 193–213. DOI: 10.1007/978-3-319-59870-3_11. URL: https://doi.org/10.1007/978-3-319-59870-3_11.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*. 2014, pp. 1054–1067. DOI: 10.1145/2660267.2660348. URL: http://doi.acm.org/10.1145/2660267.2660348.

[FC01] Frank Fiore and Shawn Collins. *Successful Affiliate Marketing for Merchants*. QUE. Indianapolis, IN, USA: Pearson Education, 2001. ISBN: 0789725258.

[FMR13] Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. "Computational Fuzzy Extractors". In: *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*. 2013, pp. 174–193. DOI: 10.1007/978-3-642-42033-7_10. URL: https://doi.org/10.1007/978-3-642-42033-7_10.

[Fol14] Janos Follath. "Gaussian sampling in lattice based cryptography". In: *De Gruyter* 60 (2014), pp. 1–23. DOI: 10.2478/tmmp-2014-0022.

[Gal13] Steven D. Galbraith. *Space-efficient variants of cryptosystems based on learning with errors*. 2013, *Retrieved Mar 20 2018*. URL: https://www.math.auckland.ac.nz/~sgal018/compact-LWE.pdf.

[GCF11] Saikat Guha, Bin Cheng, and Paul Francis. "Privad: Practical Privacy in Online Advertising". In: *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2011, Boston, MA, USA, March 30 - April 1, 2011*. 2011, pp. 169–182.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. "Public-Key Cryptosystems from Lattice Reduction Problems". In: *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*. 1997, pp. 112–131. DOI: 10.

1007/BFb0052231. URL: https://doi.org/10.1007/BFb0052231.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. "Homomorphic Evaluation of the AES Circuit". In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. 2012, pp. 850–867. DOI: 10.1007/978-3-642-32009-5_49. URL: https://doi.org/10.1007/978-3-642-32009-5_49.

[GLM16] Matthew Green, Watson Ladd, and Ian Miers. "A Protocol for Privately Reporting Ad Impressions at Scale". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 2016, pp. 1591–1601. DOI: 10.1145/2976749.2978407. URL: http://doi.acm.org/10.1145/2976749.2978407.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority". In: *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*. 1987, pp. 218–229. DOI: 10.1145/28395.28420. URL: http://doi.acm.org/10.1145/28395.28420.

[GN08] Nicolas Gama and Phong Q. Nguyen. "Predicting Lattice Reduction". In: *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*. 2008, pp. 31–51. DOI: 10.1007/978-3-540-78967-3_3. URL: https://doi.org/10.1007/978-3-540-78967-3_3.

[GNR10] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. "Lattice Enumeration Using Extreme Pruning". In: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*. 2010, pp. 257–278. DOI: 10.1007/978-3-642-13190-5_13. URL: https://doi.org/10.1007/978-3-642-13190-5_13.

[Gol+10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. "Robustness of the Learning with Errors Assumption". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. 2010, pp. 230–240. URL: http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/19.html.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004. ISBN: 0-521-83084-2.

[Gou14] Lauren Gould. *10 Important Measurements in Influencer Marketing: Instagram*. https://business.experticity.com/10-important-measurements-in-influencer-marketing-instagram/. 2014, *Retrieved Aug 9 2017*.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. 2008, pp. 197–206. DOI: 10.1145/1374376.1374407. URL: http://doi.acm.org/10.1145/1374376.1374407.

[Gre16] Andy Greenberg. *Apple's 'Differential Privacy' is about collecting your data – but not your data*. https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/. 2016, *Retrieved Mar 26 2018*.

[GRS09] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. "Universally utility-maximizing privacy mechanisms". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. 2009, pp. 351–360. DOI: 10.1145/1536414.1536464. URL: http://doi.acm.org/10.1145/1536414.1536464.

[Her+14] Michael Herrmann, Alfredo Rial, Claudia Díaz, and Bart Preneel. "Practical privacy-preserving location-sharing based services with aggregate statistics". In: *7th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec'14, Oxford, United Kingdom, July 23-25, 2014*. 2014, pp. 87–98. DOI: 10.1145/2627393.2627414. URL: http://doi.acm.org/10.1145/2627393.2627414.

[HHB10] Hamed Haddadi, Pan Hui, and Ian Brown. "MobiAd: Private and Scalable Mobile Advertising". In: *Proceedings of the Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture*. MobiArch '10. Chicago, Illinois, USA: ACM, 2010, pp. 33–38. ISBN: 978-1-4503-0143-5. DOI: 10.1145/1859983.1859993. URL: http://doi.acm.org/10.1145/1859983.1859993.

[Hit15] Lucy Hitz. *2015 Influencer Marketing Guide*. Tech. rep. SimplyMeasured, 2015.

[HKM18] Gottfried Herold, Elena Kirshanova, and Alexander May. "On the asymptotic complexity of solving LWE". In: *Des. Codes Cryptography* 86.1 (2018), pp. 55–83. DOI: 10.1007/s10623-016-0326-0. URL: https://doi.org/10.1007/s10623-016-0326-0.

[HM17] Gottfried Herold and Alexander May. "LP Solutions of Vectorial Integer Subset Sums - Cryptanalysis of Galbraith's Binary Matrix LWE". In: *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*. 2017, pp. 3–15. DOI: 10.1007/978-3-662-54365-8_1. URL: https://doi.org/10.1007/978-3-662-54365-8_1.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem". In: *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland,*

Oregon, USA, June 21-25, 1998, Proceedings. 1998, pp. 267–288. DOI: 10.1007/BFb0054868. URL: https://doi.org/10.1007/BFb0054868.

[Hut+17] Christopher Huth, Daniela Becker, Jorge Guajardo, Paul Duplys, and Tim Güneysu. "Securing Systems With Indispensable Entropy: LWE-Based Lossless Computational Fuzzy Extractor for the Internet of Things". In: *IEEE Access* 5 (2017), pp. 11909–11926. DOI: 10.1109/ACCESS.2017.2713835. URL: https://doi.org/10.1109/ACCESS.2017.2713835.

[Ibm] *What is quantum computing?* Tech. rep. IBM Q, IBM Corporation.

[IK06] Seidu Inusah and Tomasz J. Kozubowski. "A discrete analogue of the Laplace distribution". In: *J. Stat. Plan. Inference* 136 (2006), pp. 1090–1102. DOI: 10.1016/j.jspi.2004.08.014.

[Jin14] Zhengjun Jing. "An efficient homomorphic aggregate signature scheme based on lattice". In: *Mathematical Problems in Engineering* 2014 (2014).

[Jun+13] Taeho Jung, XuFei Mao, Xiang-Yang Li, Shaojie Tang, Wei Gong, and Lan Zhang. "Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation". In: *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*. 2013, pp. 2634–2642. DOI: 10.1109/INFCOM.2013.6567071. URL: https://doi.org/10.1109/INFCOM.2013.6567071.

[KF15] Paul Kirchner and Pierre-Alain Fouque. "An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices". In: *CRYPTO '15* 9215 (2015), pp. 43–62.

[KM12] V. Kumar and Rohan Mirchandani. "Increasing the ROI of Social Media Marketing". In: *MIT Sloan Management Review* Fall (2012), pp. 1–21.

[Lal] Estee Lalonde. *essiebutton*. YouTube. URL: https://www.youtube.com/user/essiebutton.

[Lan14] Brian M. Landry. *Influence at Scale*. Tech. rep. 2014, pp. 735–744. DOI: 10.1145/2783258.2783334.

[LC13] Qinghua Li and Guohong Cao. "Efficient Privacy-Preserving Stream Aggregation in Mobile Sensing with Low Aggregation Error". In: *Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings*. 2013, pp. 60–81. DOI: 10.1007/978-3-642-39077-7_4. URL: https://doi.org/10.1007/978-3-642-39077-7_4.

[LCP14] Qinghua Li, Guohong Cao, and Thomas F. La Porta. "Efficient and Privacy-Aware Data Aggregation in Mobile Sensing". In: *IEEE Trans. Dependable Sec. Comput.* 11.2 (2014), pp. 115–129. DOI: 10.1109/TDSC.2013.31. URL: https://doi.org/10.1109/TDSC.2013.31.

[Leo+15] Iraklis Leontiadis, Kaoutar Elkhiyaoui, Melek Önen, and Refik Molva. "PUDA - Privacy and Unforgeability for Data Aggregation". In: *Cryptology and Network Security - 14th International Conference, CANS 2015, Marrakesh, Morocco, December 10-12, 2015, Proceedings*. 2015, pp. 3–18. DOI: 10.1007/978-3-319-26823-1_1. URL: https://doi.org/10.1007/978-3-319-26823-1_1.

[Liu+13] Bin Liu, Anmol Sheth, Udi Weinsberg, Jaideep Chandrashekar, and Ramesh Govindan. "AdReveal: improving transparency into online targeted advertising". In: *Twelfth ACM Workshop on Hot Topics in Networks, HotNets-XII, College Park, MD, USA, November 21-22, 2013*. 2013, 12:1–12:7. DOI: 10.1145/2535771.2535783. URL: http://doi.acm.org/10.1145/2535771.2535783.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische Annalen* 261.4 (1982), pp. 515–534.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. "Generalized Compact Knapsacks Are Collision Resistant". In: *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*. 2006, pp. 144–155. DOI: 10.1007/11787006_13. URL: https://doi.org/10.1007/11787006_13.

[LN13] Mingjie Liu and Phong Q. Nguyen. "Solving BDD by Enumeration: An Update". In: *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco, CA, USA, February 25-March 1, 2013. Proceedings*. 2013, pp. 293–309. DOI: 10.1007/978-3-642-36095-4_19. URL: https://doi.org/10.1007/978-3-642-36095-4_19.

[LP11] Richard Lindner and Chris Peikert. "Better Key Sizes (and Attacks) for LWE-Based Encryption". In: *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*. 2011, pp. 319–339. DOI: 10.1007/978-3-642-19074-2_21. URL: https://doi.org/10.1007/978-3-642-19074-2_21.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*. 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1. URL: https://doi.org/10.1007/978-3-642-13190-5_1.

[LS16] Yang Liu and Andrew Simpson. "Privacy-preserving targeted mobile advertising: requirements, design and a prototype implementation". In: *Softw., Pract. Exper.* 46.12 (2016), pp. 1657–1684. DOI: 10.1002/spe.2403. URL: https://doi.org/10.1002/spe.2403.

[Ma15] Alexandra Ma. "How To Make Money On Instagram". In: *Huffington Post* (2015).

[Men+16] Wei Meng, Ren Ding, Simon P. Chung, Steven Han, and Wenke Lee. "The Price of Free: Privacy Leakage in Personalized Mobile In-Apps Ads". In: *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. 2016.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*. Vol. 671. The Kluwer International Series in Engineering and Computer Science. Boston, Massachusetts: Kluwer Academic Publishers, 2002.

[Mic02] Daniele Micciancio. "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions". In: *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*. 2002, pp. 356–365. DOI: 10.1109/SFCS.2002.1181960. URL: https://doi.org/10.1109/SFCS.2002.1181960.

[Mic18] Daniele Micciancio. *On the Hardness of Learning With Errors with Binary Secrets*. 2018. URL: http://cseweb.ucsd.edu/~daniele/papers/BinLWE.pdf.

[MM11] Daniele Micciancio and Petros Mol. "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions". In: *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*. 2011, pp. 465–484. DOI: 10.1007/978-3-642-22792-9_26. URL: https://doi.org/10.1007/978-3-642-22792-9_26.

[MOV96] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN: 0-8493-8523-7.

[MP12] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. 2012, pp. 700–718. DOI: 10.1007/978-3-642-29011-4_41. URL: https://doi.org/10.1007/978-3-642-29011-4_41.

[MP13] Daniele Micciancio and Chris Peikert. "Hardness of SIS and LWE with Small Parameters". In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*. 2013, pp. 21–39. DOI: 10.1007/978-3-642-40041-4_2. URL: https://doi.org/10.1007/978-3-642-40041-4_2.

[MR04] Daniele Micciancio and Oded Regev. "Worst-Case to Average-Case Reductions Based on Gaussian Measures". In: *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*. 2004, pp. 372–381. DOI:

10.1109/FOCS.2004.72. URL: https://doi.org/10.1109/FOCS.2004.72.

[MR09] Daniele Micciancio and Oded Regev. "Lattice-based Cryptography". In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2009, pp. 147–191. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7_5. URL: https://doi.org/10.1007/978-3-540-88702-7_5.

[MR17] Brendan McMahan and Daniel Ramage. *Federated Learning: Collaborative Machine Learning without Centralized Training Data*. Tech. rep. Google LLC, 2017.

[NAB11] Animesh Nandi, Armen Aghasaryan, and Makram Bouzid. "P3: A privacy preserving personalization middleware for recommendation-based services". In: *Fourth Hot Topics in Privacy Enhancing Technologies Symposium (HotPETS 2011)*. 2011.

[Ngu99] Phong Q. Nguyen. "Cryptanalysis of the Goldreich - Goldwasser - Halevi Cryptosystem from Crypto '97". In: *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. 1999, pp. 288–304. DOI: 10.1007/3-540-48405-1_18. URL: https://doi.org/10.1007/3-540-48405-1_18.

[Nie+14] Frank Niedermeyer, Simone Steinmetzer, Martin Kroll, and Rainer Schnell. "Cryptanalysis of Basic Bloom Filters used for Privacy Preserving Record Linkage". In: *J. Priv. Confidentiality* 6.2 (2014), pp. 59–79.

[Nov17] Jordan Novet. *Following Apple, Google is exploring differential privacy in Gboard for Android*. https://venturebeat.com/2017/04/06/following-apple-google-tests-differential-privacy-in-gboard-for-android/. 2017.

[NR06] Phong Q. Nguyen and Oded Regev. "Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures". In: *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*. 2006, pp. 271–288. DOI: 10.1007/11761679_17. URL: https://doi.org/10.1007/11761679_17.

[Pei10] Chris Peikert. "An Efficient and Parallel Gaussian Sampler for Lattices". In: *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*. 2010, pp. 80–97. DOI: 10.1007/978-3-642-14623-7_5. URL: https://doi.org/10.1007/978-3-642-14623-7_5.

[Pei16] Chris Peikert. "A Decade of Lattice Cryptography". In: *Foundations and Trends in Theoretical Computer Science* 10.4 (2016), pp. 283–424. DOI: 10.1561/0400000074. URL: https://doi.org/10.1561/0400000074.

[Pie12] Krzysztof Pietrzak. "Cryptography from Learning Parity with Noise". In: *SOFSEM 2012: Theory and Practice of Computer Science - 38th Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 21-27, 2012. Proceedings*. 2012, pp. 99–114. DOI: 10.1007/978-3-642-27660-6_9. URL: https://doi.org/10.1007/978-3-642-27660-6_9.

[PMF12] Irena Pletikosa Cvijikj, Florian Michahelles, and Elgar Fleisch. "Social Media Integration into the GS1 Framework". In: *Auto-ID Labs White Paper* January (2012), pp. 1–21.

[PR06] Chris Peikert and Alon Rosen. "Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*. 2006, pp. 145–166. DOI: 10.1007/11681878_8. URL: https://doi.org/10.1007/11681878_8.

[PRSD17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. "Pseudorandomness of Ring-LWE for Any Ring and Modulus". In: *Proc. 49th Annu. ACM SIGACT Symp. Theory Comput. STOC 2017*. 2017, pp. 461–473. URL: https://eprint.iacr.org/2017/258.pdf.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. "A Framework for Efficient and Composable Oblivious Transfer". In: *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*. 2008, pp. 554–571. DOI: 10.1007/978-3-540-85174-5_31. URL: https://doi.org/10.1007/978-3-540-85174-5_31.

[Rad15] National Public Radio. *How Do You Market To Millennials? - new boom*. http://www.npr.org/series/352990765/new-boom. 2015, *Retrieved Aug 9 2017*.

[Rap14] Samantha Raphelson. *Amid The Stereotypes, Some Facts About Millennials*. https://www.northcountrypublicradio.org/news/npr/354196302/amid-the-stereotypes-some-facts-about-millennials. 2014, *Retrieved Aug 9 2017*.

[RCC18] Matthew Rosenberg, Nicholas Confessore, and Carole Cadwalladr. "How Trump Consultants Exploited the Facebook Data of Millions". In: *The New York Times* (2018, *Retrieved Mar 26 2018*).

[Reg03] Oded Regev. "New lattice based cryptographic constructions". In: *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*. 2003, pp. 407–416. DOI: 10.1145/780542.780603. URL: http://doi.acm.org/10.1145/780542.780603.

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*. 2005, pp. 84–93. DOI: 10.1145/1060590.1060603. URL: http://doi.acm.org/10.1145/1060590.1060603.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56.6 (2009), 34:1–34:40. DOI: 10.1145/1568318.1568324. URL: http://doi.acm.org/10.1145/1568318.1568324.

[Reg10] Oded Regev. *The Learning with Errors Problem - presentation*. Cambridge, 2010. URL: http://slideplayer.com/slide/4519318/.

[Rei] Pamela Reif. *pamela_rf*. Instagram. URL: https://www.instagram.com/pamela_rf/?hl=en.

[RH15] Margaret Rouse and Matthew Haughn. *Millennials (Millennial generation)*. http://whatis.techtarget.com/definition/millennials-millennial-generation. 2015, *Retrieved Mar 23 2018*.

[RS10] Markus Rückert and Michael Schneider. "Estimating the Security of Lattice-based Cryptosystems". In: *IACR Cryptology ePrint Archive* 2010 (2010), p. 137. URL: http://eprint.iacr.org/2010/137.

[Saa15] Markku-Juhani O. Saarinen. "Gaussian Sampling Precision in Lattice Cryptography". In: *Eprint* (2015), pp. 4–12. URL: https://eprint.iacr.org/2015/953.pdf.

[Sch03] Claus-Peter Schnorr. "Lattice Reduction by Random Sampling and Birthday Methods". In: *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*. 2003, pp. 145–156. DOI: 10.1007/3-540-36494-3_14. URL: https://doi.org/10.1007/3-540-36494-3_14.

[SD17] Noah Stephens-Davidowitz. Personal email communication. Nov. 16, 2017.

[SE94] Claus-Peter Schnorr and M. Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems". In: *Math. Program.* 66 (1994), pp. 181–199. DOI: 10.1007/BF01581144. URL: https://doi.org/10.1007/BF01581144.

[Shi+11] Elaine Shi, T.-H. Hubert Chan, Eleanor G. Rieffel, Richard Chow, and Dawn Song. "Privacy-Preserving Aggregation of Time-Series Data". In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*. 2011. URL: http://www.isoc.org/isoc/conferences/ndss/11/pdf/9_3.pdf.

[Sho] Victor Shoup. *Number theory library 5.5.2 (NTL) for C++*. URL: http://www.shoup.net/ntl/.

[Sho97] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172. URL: https://doi.org/10.1137/S0097539795293172.

[Ske46] John G Skellam. "The frequency distribution of the difference between two Poisson variates belonging to different populations". In: *Journal of the Royal Statistical Society: Series A* 109 (1946), p. 296.

[SS11] Damien Stehlé and Ron Steinfeld. "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices". In: *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*. 2011, pp. 27–47. DOI: 10.1007/978-3-642-20465-4_4. URL: https://doi.org/10.1007/978-3-642-20465-4_4.

[SV14] Nigel P. Smart and Frederik Vercauteren. "Fully homomorphic SIMD operations". In: *Des. Codes Cryptography* 71.1 (2014), pp. 57–81. DOI: 10.1007/s10623-012-9720-4. URL: https://doi.org/10.1007/s10623-012-9720-4.

[TBB10] Michael Trusov, Anand V. Bodapati, and Randolph E. Bucklin. "Determining influential users in internet social networks". In: *J. Mark. Res.* XLVII.4 (2010), pp. 643–658. ISSN: 0022-2437. DOI: 10.1509/jmkr.47.4.643.

[TDB16] Giulia Traverso, Denise Demirel, and Johannes A. Buchmann. *Homomorphic Signature Schemes - A Survey*. Springer Briefs in Computer Science. Springer, 2016. ISBN: 978-3-319-32114-1. DOI: 10.1007/978-3-319-32115-8. URL: https://doi.org/10.1007/978-3-319-32115-8.

[Tib96] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1996), pp. 267–288.

[Tou+10] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. "Adnostic: Privacy Preserving Targeted Advertising". In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2010, San Diego, California, USA, 28th February - 3rd March 2010*. 2010. URL: http://www.isoc.org/isoc/conferences/ndss/10/pdf/05.pdf.

[Tuc14] Catherine E. Tucker. "Social networks, personalized advertising and privacy controls". In: *J. Mark. Res.* LI.October (2014), pp. 546–562. ISSN: 00222437. DOI: 10.1021/ic961434r.

[UV11] Jonathan Ullman and Salil P. Vadhan. "PCPs and the Hardness of Generating Private Synthetic Data". In: *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*. 2011, pp. 400–416. DOI: 10.1007/978-3-642-19571-6_24. URL: https://doi.org/10.1007/978-3-642-19571-6_24.

[VA15] Filipp Valovich and Francesco Aldà. "Private Stream Aggregation Revisited". In: *CoRR* abs/1507.08071 (2015). arXiv: 1507.08071. URL: http://arxiv.org/abs/1507.08071.

[Val16] Filipp Valovich. "On the hardness of the Learning with Errors problem with a discrete reproducible error distribution". In: *CoRR* abs/1605.02051 (2016). arXiv: 1605.02051. URL: http://arxiv.org/abs/1605.02051.

[Val18] Andrea Valdez. "Everything you need to know about Facebook and Cambridge Analytica". In: *Wired* (2018, *Retrieved Mar 26 2018*).

[Wag02] David A. Wagner. "A Generalized Birthday Problem". In: *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*. 2002, pp. 288–303. DOI: 10.1007/3-540-45708-9_19. URL: https://doi.org/10.1007/3-540-45708-9_19.

[War65] Stanley L Warner. "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.

[Wil11] Colin P. Williams. *Explorations in Quantum Computing, Second Edition*. Texts in Computer Science. Springer, 2011. ISBN: 978-1-84628-886-9. DOI: 10.1007/978-1-84628-887-6. URL: https://doi.org/10.1007/978-1-84628-887-6.

[Yao82] Andrew Chi-Chih Yao. "Protocols for Secure Computations (Extended Abstract)". In: *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*. 1982, pp. 160–164. DOI: 10.1109/SFCS.1982.38. URL: https://doi.org/10.1109/SFCS.1982.38.

[Yao86] Andrew Chi-Chih Yao. "How to Generate and Exchange Secrets (Extended Abstract)". In: *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25. URL: https://doi.org/10.1109/SFCS.1986.25.

[Gol17] Goldman Sachs Global Investment Research. *How the millennial generation will transform the economy*. Tech. rep. 2017.

Appendix A

Curriculum vitae

Name	Daniela Becker
Citizenship	German
Date of birth	June 28 1995
Place of birth	Quakenbrück, Germany

Education

Oct 2010 - Sep 2013	Bachelor of Science, Computational Informatics *Hamburg University of Technology, Hamburg, Germany*
Jan 2013 - May 2013	Exchange Semester, Computer Science *National University of Singapore, Singapore*
Oct 2013 - Sep 2015	Master of Science, Computational Informatics *Hamburg University of Technology, Hamburg, Germany*
Sep 2014 - Jan 2015	Exchange Semester, Computer Science *Institut National des Sciences Appliquées, Toulouse, France*
Oct 2015 - Jul 2018	Doctoral Studies (PhD) *Hamburg University of Technology, Hamburg, Germany*
Jan 2016 - May 2016	Visiting Student, Computer Science *Carnegie Mellon University, Pittsburgh, USA*

Work Experience

Sep 2011 - Oct 2011	Intern - Software Development Dept. *Kuehne + Nagel Road SAS (Alloin Transports SAS), Villefranche-sur-Sâone, France*
Oct 2011 - Dec 2012	Software Developer *Reporta Controllingsysteme AG, Hamburg, Germany*
Oct 2011 - Dec 2012	Teaching Assistant - STS Institute 1^{st} and 2^{nd} year B.Sc. Computer Science courses *Hamburg University of Technology, Hamburg, Germany*
Jun 2013 - Sep 2014	Working Student IT Dept.: Electronic Trading; Securities Trading & Taxes *Joh. Berenberg, Gossler & Co. KG, Hamburg, Germany*
Feb 2015 - Sep 2015	Master's Student - ETI Department *Bosch Sicherheitssysteme GmbH, Grasbrunn, Germany*
Nov 2015 - Jul 2018	Research Scientist - Security & Privacy *Robert Bosch LLC, RTC North America, Pittsburgh, USA*