Learning with Errors in the Exponent

Özgür Dagdelen* TU Darmstadt
Sebastian Gajek NEC Research Labs
Florian Göpfert TU Darmstadt

Abstract

We initiate the study of a novel class of group-theoretic intractability problems. Inspired by the theory of learning in presence of errors [Regev, STOC'05] we ask if noise in the exponent amplifies intractability. We put forth the notion of *Learning with Errors in the Exponent (LWEE)* and rather surprisingly show that various attractive properties known to exclusively hold for lattices carry over. Most notably are worst-case hardness and post-quantum resistance. In fact, LWEE's duality is due to the reducibility to two seemingly unrelated assumptions: learning with errors and the representation problem [Brands, Crypto'93] in finite groups. For suitable parameter choices LWEE superposes properties from each individual intractability problem. The argument holds in the classical and quantum model of computation.

We give the very first construction of a semantically secure public-key encryption system in the standard model. The heart of our construction is an "error recovery" technique inspired by [Joye-Libert, Eurocrypt'13] to handle critical propagations of noise terms in the exponent.

Keywords: Lattice theory, group theory, public-key encryption, existential relations, double hardness

*Part of the research was conducted while interning at NEC Research Labs

1 Introduction

Since the introduction of public-key cryptography in the ground-breaking paper of Diffie and Hellman [DH76], cryptographic systems with versatile functionality have been introduced. Deeming the system secure is a delicate task. One typically conducts a polynomial-time reduction to a computational problem conjectured to be intractable. Proofs of such nature give the strongest qualitative and quantitative arguments. On the flip side, reductions reveal the Achilles' heel of any cryptosystem. Security holds as long as no polynomial-time algorithm solves the underlying problem. Since the introduction of contemporary cryptography a central concern has been to identify computational-intractable problems and assess their hardness.

Among the most carefully scrutinized cryptographic problems are probably the discrete logarithm in finite groups and factorization. Shor's celebrated theorems [Sho94, Sho97a] curtailed for the first time the confidence of founding cryptosystems on group-theoretic assumptions. He showed the existence of polynomial-time solvers for integer factorization and discrete logarithm computation in the non-classical quantum computation model. Researchers have then begun to look for alternative computational problems. In this line of work Regev explored a lattice problem class known as learning with errors (LWE) [Reg05]. Given a distribution of noisy equations $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where e is taken from a small Gaussian error distribution, the problem comes in two flavors. The search problem is to compute $\mathbf{s}$ whereas the decisional pendant asks to distinguish $(\mathbf{a}, b)$ from random elements in $\mathbb{Z}_q^n \times \mathbb{Z}_q$. There are several convincing arguments to believe in LWE's intractability [Reg10]: First, the best known solvers run in exponential time and even quantum algorithms do not seem to help. Second, learning with errors is a generalization of learning from parity with error, which is a well-studied problem in coding theory. Any major progress in LWE will most likely cause significant impact to known lower bounds of decoding random linear codes. Lastly and most importantly, breaking certain average-case problem instances of LWE breaks all instances of certain standard lattice problems [Reg05, Pei09, LM09, BLP+13]. Against the background an armada of cryptosystems has been proposed with versatile properties [Reg05, LP11, GPV08, Gen09, ABB10a, ABB10b, CHKP10, BV11a, BGV12, Bra12].

Taking the findings from lattices in presence of errors into account we carry on the study of noise as a *non-black box* intractability amplification technique. Specifically, we ask does noise effect the intractability of group-theoretic problems as well? If so, is cryptography possible in groups where noise terms propagate in the system and may easily distort the cryptographic task? Apart from the theoretical interest, our work has concrete practical motivation. Recently, large-scale electronic surveillance data mining programs put in question the security provided by present cryptographic mechanisms. (See also the IACR statement and mission on mass surveillance.[1]) One of the problems is that many security protocols in the wild are based on a single intractability problem and we do not know the exact security. What if somebody has found a clever way to factor numbers? This already suffices to decrypt most of the TLS-protected Internet traffic and eavesdrop emails, social network activities,

[1] http://www.iacr.org/misc/statement-May2014.html

and voice calls.[2] Note, answering any of the above questions in an affirmative way advertises a novel family of computational assumptions with hardness and robustness properties in the superposition of group and lattice theory.

1.1 Our Contribution

Blending Group and Lattice Theory. As an initial step towards approaching above questions, we introduce the notion of *learning with errors in the exponent* (LWEE). The assumption reconciles the group theoretic structure of discrete-log related problems with the algebraic simplicity of lattice theory. The technical idea behind can be summarized as planting an LWE sample $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ in the exponent of a generator g of some group $\mathbb{G}$ of order q. More precisely, the distribution consists of samples $(g^{\mathbf{a}}, g^{\langle \mathbf{a}, \mathbf{s} \rangle + e}) \in \mathbb{G}^n \times \mathbb{G}$ where $\mathbf{a}$ is sampled uniformly from $\mathbb{Z}_q^n$, and $\mathbf{s} \leftarrow_R \chi_s^n$, $e \leftarrow_R \chi_e$ from some distributions χ_s, χ_e. Similar to LWE, learning with errors in the exponent comes in two versions: The search version asks to compute the secret vector $\mathbf{s}$ while in the decisional variant one is supposed to distinguish LWEE samples from a randomly sampled group elements.

Existential Relations. In an attempt to confine learning in presence of errors in the exponent, we prove that the assumptions inherits the hardness from both theories. While striving for the existential relation to the family of group-theoretic assumptions, we infer a rather surprising connection to the (search) representation problem (ℓ-SRP), introduced by Brands [Bra93]. Given a tuple of uniformly sampled elements $g_1, \ldots, g_\ell, h$ from $\mathbb{G}$, the ℓ-SRP asks to compute $x_1, \ldots, x_\ell \leftarrow \chi$ for some distribution χ such that $\Pi_{i=1}^{\ell} g_i^{x_i} = h$. Note that the ℓ-SRP problem for $\ell = 1$ is essentially identical to the computational Diffie-Hellman problem. We give a tight reduction from ℓ-SRP to the search version of the LWEE problem.

Looking at the decisional learning with errors in the exponent problem, we first put forth the decisional pendant of the representation problem (ℓ-DRP): Given a tuple $g, g_1, \ldots, g_\ell, g^{x_1}, \ldots, g^{x_\ell}, h$ from $\mathbb{G}$, where $x_1, \ldots, x_\ell \leftarrow \chi$ are sampled from some distribution χ, ℓ-DRP asks to distinguish between $\Pi_{i=1}^{\ell} g_i^{x_i} = h$ and a randomly sampled value h in $\mathbb{G}$. Observe that ℓ-DRP coincides with the decisional Diffie-Hellman (DDH) problem for $\ell = 1$ and uniform distribution over $\mathbb{Z}_q$. In the same vain as done for the k-linear assumption [Sha07], we show (in Appendix A) that ℓ-DRP becomes progressively harder to solve in Shoup's generic group model [Sho97b]. We then show that DRP reduces to LWEE which implies that if we select a group $\mathbb{G}$ for which DDH is believed to be hard, the hardness carries over to an instantiation of LWEE in that group $\mathbb{G}$. It is worth mentioning that both of our reductions from the RP problem are tight. They hold for (potentially non-uniform) distributions χ, if the underlying RP problem is hard for representations sampled from the same distribution.

Investigating the relation to lattices, we show that an algorithm solving either the search or decisional LWEE problem efficiently can be turned into a successful attacker against the search or decisional LWE problem. Our reductions are tight and hold as well for (potentially

[2] TLS's preferred cipher suite makes use of RSA-OAEP to transport the (master) key in the key establishment process. Once the ephemeral master key for the session is known it is possible to derive session keys and decrypt all encrypted messages.

non-uniform) distribution χ if LWE is hard for secret $\mathbf{s}$ sampled from the same distribution.

A CONCRETE CRYPTOSYSTEM. We give a construction of a public-key encryption scheme. One may size the magnitude to which the RP and LWE intractability contribute to the overall security of the system. The selection of parameters (e.g., modulus, dimension) offers a flexibility to fine-tune the cryptosystem's resilience against progress in attacking the underlying RP or LWE problem or the evolution of quantum computers. Concretely, one may choose to make the scheme short, post-quantum secure, or double-hard. We discuss candidate parameter choices in Section 4.3. We remark that our construction serves the sole purpose of showcasing the possibility of designing cryptosystems based on "errors in the exponent". In practical applications, a combination of two encryption systems, say El-Gamal and Regev encryption, and each system encrypting information-theoretically a share of the message, would be given the preferred choice.

1.2 Our Techniques

The idea behind our scheme is reminiscent of Regev's public-key encryption scheme. In a nutshell, the public key is an LWEE instance $(g^{\mathbf{A}}, g^{\mathbf{As}+\mathbf{x}}) \in \mathbb{G}^{n\times n} \times \mathbb{G}^n$. Ciphertexts consist of two LWEE instances $C = (\mathbf{c_0}, c_1)$ where $\mathbf{c_0} = g^{\mathbf{Ar}+\mathbf{e}_0}$ encapsulates a random key $\mathbf{r} \in \mathbb{Z}_q^n$ and $c_1 = g^{\langle \mathbf{b},\mathbf{r}\rangle + e_1} \cdot g^{\alpha\mu}$ encrypts the message μ (we discuss the exact value of α below). The tricky part is the decryption algorithm. All known LWE-based encryption schemes require some technique to handle the noise terms. Otherwise, decryption is prone to err. Regev's technique ensures small error terms. One simply rounds $c_1 - \mathbf{c}_0\mathbf{s}$ to some reference value c_b indicating the encryption of bit b. While rounding splendidly works on integers, the technique fails in our setting. In contrast to addition and multiplication of group elements, there are no known polynomial-time algorithms for geometric operations. In fact, recovering the most significant bit—a basic operation for rounding—is conjectured to be a hard problem [FPSZ06].

Our first attempt thus was to scale the noise with some scalar t such that all error terms are *even*. The advantage of even noise terms has been demonstrated in many constructions of fully homomorphic encryption as a technique to "round to the closest" bit [BV11b, BV11a, BGV12, CCK+13]. We would then round to the closest bit "in the exponent" using the Goldwasser-Micali trick of computing the Jacobi symbol [GM82]. (Essentially, the Jacobi symbol computes the least significant bit of the exponent). The crux of the technique is that it works as long as the error in the exponent does not wrap around the order q of the group. Otherwise, error terms might become odd and decryption fails (since $2 \nmid q$). To solve the problem one might feel tempted to also choose even q. However, Brakerski and Vaikuntanathan prove that the scaled version of LWE, where samples are of the form $(\mathbf{a}_i, \mathbf{a}_i \cdot \mathbf{s} + t \cdot e_i)$ for some scalar t, is equivalent to the standard LWE assumption as long as scalar t and modulus q are coprime [BV11b]. In other words, if t is even in our construction, q must be odd. For both t and q even, unfortunately, there exists an efficient least significant bit recovery algorithm (without any trapdoor) and the whole encryption system collapses.

In our second approach we traverse a considerably different path. Instead of rounding, we synthesize the pesky error terms. To this end, we adapt the trapdoor technique of Joye

and Libert [JL13] and recover partial bits of the discrete logarithm. The main idea is to tweak the modulus in a smart way. Given composite modulus $N = pq$ with p', q', such that $p = 2^k p' + 1$ and $q = 2^k q' + 1$ are prime, there exists an efficient algorithm for recovering the k least significant bits of the discrete logarithm. We choose the parameters so that the sum of all error terms in the exponent is (with high probability) at most $2^{k-\ell}$. This leads to a "gap" between error bits and those bits covert by the discrete log instance. We plant the message in this gap by shifting it to the $2^{k-\ell}$'s bit, where ℓ is the size of the message we want to decrypt. Hence, we choose $\alpha = 2^{k-\ell}$ in our construction to shift the message bits accordingly.

1.3 Previous Work

Brickell and McCurley [BM92] to the best of our knowledge were the first to study cryptographic algorithms under hedged hardness assumptions. The authors propose a variant of Schnorr's identification scheme [Sch90] secure assuming the intractability of discrete logarithms in a group of composite order N. Their scheme is witness-hiding and sound if factoring and computing discrete logarithms are simultaneously hard. Assuming factoring is easy, their scheme degenerates to soundness under the DL assumption.

Learning with errors in the exponent has a different nature. LWEE remains intractable despite the fact that there exists an attacker breaking either of the underlying problems as long as the peer assumption remains hard in presence of the breaker. It is well known that to factor N, it suffices to be able to compute the discrete log modulo N; to compute the discrete log modulo N, it suffices to factor and compute the discrete log modulo primes. Learning with errors in the exponent builds upon two orthogonal assumptions. For appropriate parameter choices breaking one assumption will not degenerate the security of the system, unless the partner assumption is secure in presence of the breaker or significant progress is made in reconciling the representation and learning with errors problem.

We also mention the work of Gentry and Halevi [GH11]. They give a fully homomorphic encryption construction from a lattice-based somewhat homomorphic encryption and ElGamal encryption scheme. Instead of squashing the decryption function, they compress ciphertexts from the homomorphic scheme into a single ElGamal ciphertext. Similar to our work, their work attempts to build encryption schemes upon lattice and group-theoretic assumptions, but it provides no hedged security. For the proof to come through they require both the LWE and DDH assumption to simultaneously hold.

1.4 Extensions and Open Problems

While learning with errors in the exponent is an interesting concept in its own right, it requires further thorough inspection. Here we point out a few possible directions for future research:

- It would be interesting to cryptanalyze the assumption. This would help nail down concrete security parameters, in particular for the case of double-hardness where both

underlying assumptions contribute to the overall security.

- We are unaware of any existential relation between the representation and learning with errors assumption neither in the classical nor quantum model of computation. In fact, any insight would require progress in solving the hidden subgroup problem (HSP) in certain finite Abelian and non-Abelian groups. Shor's discrete-log quantum algorithm crucially relies on the HSP in Abelian groups. However, efficient quantum algorithms for the HSP in non-Abelian groups are unknown as they would give an efficient algorithm for solving the unique shortest-vector problem, being a special case of the shortest vector problem (SVP) [Reg04].

- Clearly, building further cryptosystems based on the search or decisional variant of learning with errors in the exponent is an interesting direction. A candidate to look at is the Naor-Reingold pseudorandom function which bears reminiscence to the structure of learning with errors in the exponent [NR04]. Recall, the NR pseudorandom function is defined as $f_{\mathbf{s}}(\mathbf{a}) = g^{\Pi a_i s_i}$ where g generates a group $\mathbb{G}$ of prime order q, the input to the function $\mathbf{a}$ is an integer in bit representation $a_i \in \{0,1\}$ and the seed is $\mathbf{s} \in \mathbb{Z}_q^n$. It would be interesting, if one can get a tight reduction to LWEE from a slightly modified construction $f_{\mathbf{s}}(\mathbf{a}) = g^{\sum a_i s_i}$ where $a_n = 1$ and s_n is the error term. Further, it would be interesting to investigate, if one could derive security for weak secrets, leakage-resilience, or key-dependence thanks the embedded LWE instance. Goldwasser et al. have shown that LWE bears many attractive robustness guarantees for this purpose [GKPV08].[3]

2 Preliminaries

In this section we introduce some notation and recall the representation and learning with errors problem for both the search and decision variant. No decisional pendant of the representation problem has been introduced. We give a formal definition and show that the decisional version is as least as hard as the decisional Diffie-Hellman problem.

2.1 Notation

Random Sampling, Negligibility and Indistinguishability. If $\mathcal{D}$ is a probability distribution, we denote by $d \leftarrow_R \mathcal{D}$ the process of sampling a value d randomly according to $\mathcal{D}$. In case S is a set, then $s \leftarrow_R S$ means that the value s is sampled according to a uniform distribution over the set S. We write $[m]$ for the set $\{0, 1, \ldots, m-1\}$. The expression $\lceil x \rfloor$ denotes the nearest integer to $x \in \mathbb{R}$, i.e., $\lceil x \rfloor = \lceil x - 0.5 \rceil$.

A function $\varepsilon()$ is called *negligible* (in the security parameter κ) if it decreases faster than any polynomial $poly(\kappa)$ for some large enough κ. An algorithm $\mathcal{A}$ runs in probabilistic

[3] In a nutshell, this is so because LWE as "assumption" can be shown to hard despite weak secrets, i.e. keys where a fraction of bits leaked.

polynomial-time (PPT) if $\mathcal{A}$ is randomized—uses internal random coins— and for any input $x \in \{0,1\}^*$ the computation of $\mathcal{A}(x)$ terminates in at most $poly(|x|)$ steps. If the running time of an algorithm is $t' \approx t$, we mean that the distance between t' and t is negligible.

Let $X = \{X_\kappa\}_{\kappa\in\mathbb{N}}$ and $Y = \{Y_\kappa\}_{\kappa\in\mathbb{N}}$ be two distribution ensembles. We say X and Y are (t,ϵ)-computationally indistinguishable if for every PPT distinguisher $\mathcal{A}$ with running time t, there exists a function $\epsilon(\kappa)$ such that $|\Pr[\mathcal{A}(X)=1] - \Pr[\mathcal{A}(Y)=1]| \leq \epsilon(\kappa)$ (and we write $X \approx_{(t,\epsilon)} Y$). If $\mathcal{A}$ is PPT and $\epsilon(\kappa)$ is negligible, we simply say X and Y are (computationally) indistinguishable (and we write $X \approx Y$). We say a distribution ensemble $X = \{X_\kappa\}_{\kappa\in\mathbb{N}}$ has (high) min-entropy, if for all large enough κ, the largest probability of an element in X_κ is $2^{-\kappa}$. We say a distribution ensemble $X = \{X_\kappa\}_{\kappa\in\mathbb{N}}$ is well-spread, if for any polynomial $poly(\cdot)$ and all large enough κ, the largest probability of an element in X_κ is smaller than $poly(\kappa)$. (In other words, the max-entropy of distributions in X must vanish super-logarithmatically.) Under the Gaussian distribution D_σ with parameter $\sigma > 0$, the probability of sampling an integer $x \in \mathbb{Z}$ is proportional to $\exp[-x^2/(2\sigma^2)]$.

Vectors and Matrices in the Exponent. We denote vectors by bold lower case letters and matrices by bold upper case letters. The i^{th} row of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}[i]$, the j^{th} element of a vector $\mathbf{a}$ is denoted by a_j, To ease notation we sometimes write $\mathbf{a}_i$ for the i^{th} row vector, and $a_{i,j}$ for the element in the i^{th} row and j^{th} column of matrix $\mathbf{A}$. Let $\mathbb{G}$ be a group of order q, g a generator of $\mathbb{G}$, $\mathbf{a}$ a vector in $\mathbb{Z}_q^n$, and $\mathbf{A}$ a matrix in $\mathbb{Z}_q^{m\times n}$. We use the notation $g^{\mathbf{a}} \in \mathbb{G}^n$ to denote the vector $g^{\mathbf{a}} \stackrel{\text{def}}{=} (g^{a_1}, \cdots, g^{a_n})$ and $g^{\mathbf{A}} \in \mathbb{G}^{m\times n}$ to denote the matrix $g^{\mathbf{A}} \stackrel{\text{def}}{=} (g^{\mathbf{a}_1}, \cdots, g^{\mathbf{a}_m})^\top$.

Computations in the Exponent. Given $g^{\mathbf{a}}$ and $\mathbf{b}$, the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ in the exponent, denoted by $g^{\langle \mathbf{a},\mathbf{b}\rangle}$, is

$$\prod_{i=1}^{n} (g^{a_i})^{b_i} = \prod_{i=1}^{n} g^{a_i \cdot b_i} = g^{\sum_{i=1}^{n} a_i \cdot b_i} = g^{\langle \mathbf{a},\mathbf{b}\rangle} .$$

Likewise, a matrix-vector product in the exponent, given a vector $\mathbf{v}$ and $g^{\mathbf{A}}$ for a matrix $\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{pmatrix}$ can be performed by

$$\prod_{i=1}^{n} (g^{\mathbf{a}_i})^{v_i} = \prod_{i=1}^{n} g^{\mathbf{a}_i \cdot v_i} = g^{\sum_{i=1}^{n} \mathbf{a}_i \cdot v_i} = g^{\mathbf{A}\mathbf{v}} .$$

Adding (and subtracting) in the exponent is computed via element-wise multiplication (and division) of the group elements $g^{\mathbf{a}} \cdot g^{\mathbf{b}} = g^{\mathbf{a}+\mathbf{b}}$.

Quadratic Residuosity. Let $\mathbb{G}$ be a group of prime order p. The Legendre symbol verifies whether an element $a \in \mathbb{G}$ is a quadratic residue, i.e., $x^2 \equiv a \bmod p$ for some x. If $\mathbb{L}(a,p) := a^{(p-1)/2} = 1$, this is the case; otherwise $\mathbb{L}(a,p) = -1$. More generally, for $n \geq 2$, we define $\mathbb{L}(a,p)_n := a^{(p-1)/gcd(n,p-1)}$. For a group of composite order $N = p_1 \cdots p_k$ where the p_i

are odd primes, one uses its generalization, namely the Jacobi symbol, which is defined as $\mathbb{J}(a,N) = \prod_{i=1}^{k} \mathbb{L}(a,p_i)$. Note that $\mathbb{J}(a,N) = 1$ does not imply that a is a quadratic residue modulo N. However, if $\mathbb{J}(a,N) = -1$, a is certainly not. The set of quadratic residues modulo N is denoted by $\mathbb{QR}_N := \{a^2 \ : \ a \in \mathbb{Z}_N^*\}$. By $\mathbb{J}_N$ we denote the subgroup of all elements from $\mathbb{Z}_N^*$ with Jacobi symbol 1, i.e., $\mathbb{J}_N = \{a \in \mathbb{Z}_N^* \ : \ \mathbb{J}(a,N) = 1\}$. Note that $\mathbb{QR}_N$ is a subgroup of $\mathbb{J}_N$. It is widely believed that one cannot efficiently decide whether an element $a \in \mathbb{J}_N$ is a quadratic residue modulo N if the prime factors of N are unknown. (For more details, we refer to Appendix B.1.)

2.2 Standard Group-Theoretic Problems

We recall in the meanwhile standard assumptions of discrete log and Diffie-Hellman. For our proofs, we need slightly generalized versions of the problem statements to handle exponents chosen from some distribution χ with (at least) minimal entropy. Throughout the paper, let $\mathbb{G}$ be a group of order q and g be a generator of $\mathbb{G}$. We implicitly include g and q in the description of $\mathbb{G}$ when the meaning is clear from the context.

Definition 2.1 (Discrete Log)**.** *Let χ be a distribution over $\mathbb{Z}_q^*$, and let $x \leftarrow_R \chi$. The Discrete Logarithm ($\mathsf{DL}_{\mathbb{G},\chi}$) problem is (t,ϵ)-hard if any algorithm $\mathcal{A}$, running in time t, upon input g^x, outputs x with probability at most ϵ.*

We now formulate a slightly generalized version of the Diffie-Hellman problem for distributions with (minimal) entropy.

Definition 2.2 (Diffie–Hellman)**.** *Let χ be a distribution over $\mathbb{Z}_q^*$, and let $y \leftarrow_R \chi$. Further, let x, z be uniformly sampled from $\mathbb{Z}_q^*$.*

- *The Computational Diffie–Hellman ($\mathsf{CDH}_{\mathbb{G},\chi}$) problem is (t,ϵ)-hard if any algorithm $\mathcal{A}$, running in time t, upon input (g^x, g^y), outputs g^{xy} with probability at most ϵ.*
- *The Decisional Diffie–Hellman ($\mathsf{DDH}_{\mathbb{G},\chi}$) problem is (t,ϵ)-hard if*

$$(g^x, g^y, g^{xy}) \approx_{(t,\epsilon)} (g^x, g^y, g^z) \ .$$

The idea of taking into account well-spread and min-entropy distributions χ in groups $\mathbb{G}$ of prime order q is due to Canetti [Can97]. There, the assumption is an essential ingredient towards implementing the random oracle in the standard model.

We will also make use of the rank hiding assumption introduced by Naor and Segev [NS09] (and later extended by Agrawal et al. [ADVW13]). It was proven to be equivalent to the $\mathsf{DDH}_{\mathbb{G},\chi}$ assumption for groups of prime order and uniform χ [NS09].

Definition 2.3 (Rank Hiding)**.** *Let $\mathbb{G}$ be a group of order q with generator g, and $i, j, n, m \in \mathbb{N}$ satisfying $i, j \geq 1$. The Rank Hiding problem ($\mathsf{RH}_{\mathbb{G},i,j,m,n}$) is (t,ϵ)-hard if*

$$\{(\mathbb{G}, q, g, g^{\mathbf{M}}) : \mathbf{M} \leftarrow_R \mathsf{Rk}_i(\mathbb{Z}_q^{m\times n})\} \approx_{(t,\epsilon)} \{(\mathbb{G}, q, g, g^{\mathbf{M}}) : \mathbf{M} \leftarrow_R \mathsf{Rk}_j(\mathbb{Z}_q^{m\times n})\}$$

where $\mathsf{Rk}_k(\mathbb{Z}_q^{m\times n})$ returns an $m \times n$ matrix uniformly random from $\mathbb{Z}_q^{n\times m}$ with rank $k \leq \min(n,m)$.

2.3 Representation Problem

The representation problem in a group $\mathbb{G}$ assumes that given l random group elements $g_1, \ldots, g_l \in \mathbb{G}$ and $h \in \mathbb{G}$ it is hard to find a representation $\mathbf{x} \in \mathbb{Z}_q^\ell$ such that $h = \prod_{i=1}^{\ell} g_i^{x_i}$ holds. Brands builds an electronic cash system based on the problem. The assumption has found little application since then, until its applicability to leakage-resilient cryptosystems have been investigated [KV09, ADVW13, DV14].

We now state a more general version of the search representation problem where vector $\mathbf{x} \leftarrow_R \chi^\ell$ is sampled from a distribution χ with (at least) min-entropy and where an adversary is given $m \geq 1$ samples instead of a single one.

Definition 2.4 (Search Representation Problem)**.** *Let χ be a distribution over $\mathbb{Z}_q$, and ℓ, m be integers. Sample $\mathbf{M} \leftarrow_R \mathbb{Z}_q^{m\times\ell}$, $\mathbf{h} \leftarrow_R \mathbb{Z}_q^m$, and $\mathbf{x} \leftarrow_R \chi^\ell$. The Search Representation Problem ($\mathsf{SRP}_{\mathbb{G},\chi,\ell,m}$) is (t,ϵ)-hard if any algorithm $\mathcal{A}$, running in time t, upon input $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{Mx}})$, outputs $\mathbf{x}' \in \mathbb{Z}_q^\ell$ such that $g^{\mathbf{Mx}'} = g^{\mathbf{Mx}}$ with probability at most ϵ. If χ is the uniform distribution, we sometimes skip χ in the index and say that $\mathsf{SRP}_{\mathbb{G},\ell,m}$ is (t,ϵ)-hard.*

Brands proves the equivalence of the representation problem and the discrete logarithm problem for uniform χ and $m = 1$. It is easy to verify that the reduction holds for every distribution for which the discrete logarithm problem holds.

For establishing the relations to the learning with errors in the exponent problem (cf. Section 3.2), we need a decisional variant of the representation problem. To our surprise, the decisional version has not been defined before, although the assumption is a natural generalization of the decisional Diffie-Hellman problem to ℓ-tuples (similar in spirit as the ℓ-linear problem in $\mathbb{G}$ [Sha07]). Given ℓ random group elements $g_1, \ldots, g_\ell \in \mathbb{G}$ and $g^{x_1}, \ldots, g^{x_\ell} \in \mathbb{G}$ where $x_1, \ldots, x_\ell \leftarrow_R \mathbb{Z}_q^*$, it is hard to decide if $h = \prod_{i=1}^{\ell} g_i^{x_i}$ or h is a random group element in $\mathbb{G}$. Our definition below generalizes this problem to the case, where $m \geq 1$ samples are given to an adversary and $x_1, \ldots, x_\ell$ are sampled from any min-entropy distribution χ.

Definition 2.5 (Decisional Representation Problem)**.** *Let χ be a distribution over $\mathbb{Z}_q^*$, and ℓ, m be integers. Sample $\mathbf{M} \leftarrow_R \mathbb{Z}_q^{m\times\ell}$, $\mathbf{h} \leftarrow_R \mathbb{Z}_q^m$, and $\mathbf{x} \leftarrow_R \chi^\ell$. The Decisional Representation ($\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$) problem is (t,ϵ)-hard if*

$$(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{Mx}}) \approx_{(t,\epsilon)} (g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{h}}) \ .$$

If χ is the uniform distribution over $\mathbb{Z}_q^*$, we say $\mathsf{DRP}_{\mathbb{G},\ell,m}$ is (t,ϵ)-hard.

Note that the $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ problem can be stated in the framework of the Matrix-DDH assumption recently introduced by Escala et al. [EHK+13]. We give evidence that the family of $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ problems is a class of progressively harder problems (with increasing ℓ) and thus put another class of hardness problems to the arsenal of [EHK+13]. We defer proofs of following propositions to Appendix A and C.1.

Proposition 2.6. *If $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ is (t,ϵ)-hard, then for any $\ell, m \geq 1$ with $t' \approx t$ and distribution χ with min-entropy $\mathsf{DRP}_{\mathbb{G},\chi,\ell+1,m}$ is (t',ϵ)-hard.*

Proposition 2.7. *In the generic group model* $\mathsf{DRP}_{\mathbb{G},\chi,\ell+1,m}$ *is hard for distribution* χ *with minimal entropy, even in presence of a* $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$*-oracle.*

Note that the $\mathsf{DRP}_{\mathbb{G},\chi,1,1}$-problem with χ being the uniform distribution over $\mathbb{Z}_q$ coincides with the decisional Diffie-Hellman (DDH) problem. Hence, we obtain the corollary that for uniform distributions χ, the decisional Diffie-Hellman problem implies the representation problem $\mathsf{DRP}_{\mathbb{G},\chi,\ell,1}$ for $\ell \geq 1$. In fact, Proposition 2.6 suggests a stronger argument. Assuming the decisional Diffie-Hellman problem holds for well-spread and min-entropy distributions χ, then the $\mathsf{DRP}_{\mathbb{G},\chi,\ell,1}$ holds for χ and $\ell \geq 1$.

While Propositions 2.6 and 2.7 show that the hardness of the DRP problem progressively increases with ℓ, the following proposition states that the problem remains hard with increasing number of samples m. More precisely, we show that $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ is hard as long as $\mathsf{DRP}_{\mathbb{G},\chi,\ell,1}$ and the rank hiding problem (cf. Definition 2.3) is hard. The proof can be found in Appendix C.

Proposition 2.8. *If* $\mathsf{RH}_{\mathbb{G},m,m+1,m+1,2\ell+1}$ *is* (t,ϵ)*-hard and* $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ *is* (t',ϵ')*-hard in a cyclic group* $\mathbb{G}$ *of order* q*, then for any distribution* χ_e *and any* $m > 0$ *with* $t' \approx t$ *and* $\epsilon'' \leq (1-\epsilon)^{-1}\epsilon'$ $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m+1}$ *is* (t,ϵ'')*-hard.*

2.4 Learning with Errors

The learning with errors assumption comes as a search and decision lattice problem. Given a system of m linear equations with random coefficients $\mathbf{a}_i \in \mathbb{Z}_q^n$ in the n indeterminates $\mathbf{s}$ sampled from some distribution χ_s and biased with some error e_i from the error distribution χ_e, it is hard to compute vector $\mathbf{s}$ or distinguish the solution $b_i = \sum_i^n \mathbf{a}_i\mathbf{s} + e_i$ from a uniform element in $\mathbb{Z}_q$.

Definition 2.9 (Learning with Errors)**.** *Let* n, m, q *be integers and* χ_e, χ_s *be distributions over* $\mathbb{Z}$*. For* $\mathbf{s} \leftarrow_R \chi_s$*, define the* LWE *distribution* $L^{\mathsf{LWE}}_{n,q,\chi_e}$ *to be the distribution over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$ *obtained such that one first draws* $\mathbf{a} \leftarrow_R \mathbb{Z}_q^n$ *uniformly,* $e \leftarrow_R \chi_e$ *and returns* $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ *with* $b = \langle \mathbf{a}, \mathbf{s} \rangle + e$*. Let* $(\mathbf{a}_i, b_i)$ *be samples from* $L^{\mathsf{LWE}}_{n,q,\chi_e}$ *for* $0 \leq i < m = poly(\kappa)$*.*

- *The Search Learning With Errors* ($\mathsf{SLWE}_{n,m,q,\chi_e}(\chi_s)$) *problem is* (t,ϵ)*-hard if any algorithm* $\mathcal{A}$*, running in time* t*, upon input* $(\mathbf{a}_i, b_i)_{i\in[m]}$*, outputs* $\mathbf{s}$ *with probability at most* ϵ*.*

- *The Decisional Learning with Error* ($\mathsf{DLWE}_{n,m,q,\chi_e}(\chi_s)$) *problem is* (t,ϵ)*-hard if*

$$(\mathbf{a}_i, b_i)_{i\in[m]} \approx_{(t,\epsilon)} (\mathbf{a}_i, c_i)_{i\in[m]}$$

  for a random secret $\mathbf{s} \leftarrow_R \chi_s$*.*

If χ_s *is the uniform distribution over* $\mathbb{Z}_q$*, we simply write* $\mathsf{LWE}_{n,m,q,\chi_e}$*.*

A typical distribution for the error is a discrete Gaussian distribution with an appropriate standard deviation. There are several proposals for the distribution of the secret. While the uniform distribution is the most standard one, it is shown that setting $\chi_s = \chi_e$, known as the "normal form", retains the hardness of LWE [Mic01, ACPS09]. We also note that the learning with errors problem where the error is scaled by a constant α relatively prime to q is as hard as the original definition [BV11b]. The "scaled" LWE distribution then returns $(\mathbf{a}, b)$ with $\mathbf{a} \leftarrow_R \mathbb{Z}_q^n$ and $b = \langle \mathbf{a}, \mathbf{s} \rangle + \alpha e$.

2.5 Public-Key Encryption

In a public-key encryption, the encryptor holds a public key and encrypts a message such that the holder of the corresponding secret key reconstructs the message plaintext.

Definition 2.10. *A public-key encryption scheme* (PKE) *is a tuple of three algorithms* PKE = (Setup, Encrypt, Decrypt) *such that:*

- *The key-generation algorithm* KeyGen *takes as input a security parameter* 1^κ*. It outputs a public key* pk *and a secret key* sk*.*
- *The encryption algorithm* Encrypt *takes as input the public key* pk *and a message* $m \in \mathcal{M}$*. It outputs a ciphertext* c*.*
- *The decryption algorithm Decrypt takes as input the secret key* sk *and a ciphertext* c*. It outputs a message* m*.*

We require that for all security parameters κ*, all tuples* $(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathsf{KeyGen}(1^\kappa)$*, all messages* $m \in \mathcal{M}$*, we have* $m = \mathsf{Decrypt}(\mathsf{sk}, \mathsf{Encrypt}(\mathsf{pk}, m))$ *with probability negligibly close to* 1*.*

Semantic security of a public-key encryption scheme against chosen-plaintext attacks is defined as an experiment between the challenger and adversary as follows:

Experiment $\mathbf{Exp}^{\text{IND-CPA-b}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$:

Setup: The challenger runs Setup on input 1^κ. $\mathcal{A}$ is given pk.

Challenge: At some point, $\mathcal{A}$ comes up with two messages m_0, m_1 subject to the restriction that $|m_0| = |m_1|$. $\mathcal{A}$ is given $\mathsf{Encrypt}(\mathsf{pk}, m_b)$.

Guess: $\mathcal{A}$ comes up with a guess b'. The output of the experiment is defined as b'.

The advantage of adversary $\mathcal{A}$ in violating plaintext privacy of the PKE scheme is the absolute value of the difference between the experiment for $b = 0$ and the experiment for $b = 1$.

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = |\Pr\left[\mathbf{Exp}^{\text{IND-CPA-0}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-CPA-1}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1\right]|\,.$$

Definition 2.11 (IND-CPA Security)**.** *A public-key encryption system* PKE = *(Setup, Encrypt, Decrypt) is indistinguishable under chosen-plaintext attacks or simply plaintext private, if for all polynomial-time adversaries* $\mathcal{A}$ *we have that* $\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$ *is a negligible function in* κ*.*

3 Learning with Errors in the Exponent

In this section we define learning with errors in the exponent and explore relations to known intractability problems.

3.1 Definition

For self-containment, the assumption is stated both as a search and decision problem over a group $\mathbb{G}$ of order q, and exponents sampled from distributions χ_e, χ_s over $\mathbb{Z}$. We demonstrate the versatility and general utility of the assumption in Section 4.

Definition 3.1 (Learning with Errors in the Exponent)**.** *Let $\mathbb{G}$ be a group of order q where g is a generator of $\mathbb{G}$. Let n, m, q be integers and χ_e, χ_s be distributions over $\mathbb{Z}$. For any fixed vector $\mathbf{s} \in \mathbb{Z}_q^n$, define the* LWEE *distribution $L^{\mathsf{LWEE}}_{\mathbb{G},n,q,\chi_e}$ to be the distribution over $\mathbb{G}^n \times \mathbb{G}$ obtained such that one first draws vector $\mathbf{a} \leftarrow_R \mathbb{Z}_q^n$ uniformly, $e \leftarrow_R \chi_e$ and returns $(g^{\mathbf{a}}, g^b) \in \mathbb{G}^n \times \mathbb{G}$ with $b = \langle \mathbf{a}, \mathbf{s} \rangle + e$. Let $(g^{\mathbf{a}_i}, g^{b_i})$ be samples from $L^{\mathsf{LWEE}}_{\mathbb{G},n,q,\chi_e}$ and c_i be uniformly sampled from $\mathbb{Z}_q^*$ for $0 \leq i < m = poly(\kappa)$.*

- *The Search Learning With Errors in the Exponent ($\mathsf{SLWEE}_{\mathbb{G},n,m,q,\chi_e}(\chi_s)$) problem is (t, ϵ)-hard if any algorithm $\mathcal{A}$, running in time t, upon input $(g^{\mathbf{a}_i}, g^{b_i})_{i \in [m]}$, outputs $\mathbf{s}$ with probability at most ϵ.*
- *The Decision Learning With Errors in the Exponent ($\mathsf{DLWEE}_{\mathbb{G},n,m,q,\chi_e}(\chi_s)$) problem is (t, ϵ)-hard if*

$$(g^{\mathbf{a}_i}, g^{b_i})_{i \in [m]} \approx_{(t,\epsilon)} (g^{\mathbf{a}_i}, g^{c_i})_{i \in [m]}$$

  for a random secret $\mathbf{s} \leftarrow_R \chi_s^n$. If χ_s is the uniform distribution over $\mathbb{Z}_q$, we write $\mathsf{DLWEE}_{\mathbb{G},n,m,q,\chi_e}$.

We let $\mathsf{Adv}^{\mathsf{DLWEE/SLWEE}}_{\mathbb{G},n,m,q,\chi_e,\chi_s}(t)$ denote a bound on the value ϵ for which the decisional/search LWEE *problem is (t, ϵ)-hard.*

One may interpret learning with errors in the exponent in two ways. One way is to implant an error term from a distribution χ_e into the Diffie-Hellman exponent. Another way to look at LWEE is as compressing an LWE instance within some group $\mathbb{G}$ of order q.

3.2 Relations to Group and Lattice Problems

To clarify the hardness of LWEE, we establish a connection to the representation and learning with errors problem. We summarize our main results in following four propositions. Proofs appear in Appendix C.

Proposition 3.2. *If $\mathsf{SRP}_{\mathbb{G},\chi_s,\ell,m}$ is (t, ϵ)-hard in a cyclic group $\mathbb{G}$ of order q, then for any distribution χ_e and any number of samples $m > 0$ $\mathsf{SLWEE}_{\mathbb{G},\ell,m,q,\chi_e}(\chi_s)$ is (t', ϵ)-hard with $t' \approx t$.*

Proposition 3.3. *If* $\mathsf{SLWE}_{n,m,q,\chi_e}(\chi_s)$ *is* (t,ϵ)*-hard, then for any cyclic group* $\mathbb{G}$ *of order* q *with known (or efficiently computable) generator* $\mathsf{SLWEE}_{\mathbb{G},n,m,q,\chi_e}(\chi_s)$ *is* (t',ϵ)*-hard with* $t' \approx t$*.*

Proposition 3.4. *If* $\mathsf{DRP}_{\mathbb{G},\chi_s,\ell,m}$ *is* (t,ϵ)*-hard in a cyclic group* $\mathbb{G}$ *of order* q*, then for any distribution* χ_e *and any number of samples* $m > 0$ $\mathsf{DLWEE}_{\mathbb{G},\ell,m,\chi_e}(\chi_s)$ *is* (t',ϵ)*-hard with* $t' \approx t$*.*

Proposition 3.5. *If* $\mathsf{DLWE}_{n,m,q,\chi_e}(\chi_s)$ *is* (t,ϵ)*-hard, then for any cyclic group* $\mathbb{G}$ *of order* q *with known (or efficiently computable) generator* $\mathsf{DLWEE}_{\mathbb{G},n,m,\chi_e}(\chi_s)$ *is* (t',ϵ)*-hard with* $t' \approx t$*.*

Discussion. The essence from above propositions is that there exist tight reductions from the search (resp. decision) learning with errors in the exponent problem to either the search (resp. decision) representation problem and the search (resp. decision) learning with errors problem. This has several interesting property preserving implications. As a corollary we infer that for appropriate parameter choices LWEE preserves the *hardness* and *robustness* properties of the representation and/or learning with errors problem. Essentially then LWEE boils down to the security of either of the two underlying problems. This way, the cryptosystem can be instantiated to leverage leakage resistance and post-quantum hardness thanks LWE [GKPV08, Reg05]. On the flip side, the cryptosystem may offer short instance sizes through the underlying RP problem (when instantiated on elliptic curves). Of particular interest for many emerging applications is the partnering of the two hardness assumptions. One may choose parameters such that both RP and LWE hold. We call the case *double-hard*, which appeals to provide in some sense hedged security.

3.3 On the Generic Hardness of LWEE

With Proposition 3.2-3.5 in our arsenal we conjecture LWEE to be harder than either of the underlying RP or LWE problems. The argument is heuristic and based on what is known about the hardness of each intractability problem. We refer to Appendix B for a recap and concrete security parameters.

Fix parameters such that RP and LWE problem instances give κ bits security. The only obvious known approach today to solve the LWEE instance is to first compute the discrete logarithm of samples $(g^{\mathbf{a}_i}, g^{b_i})$ and then solve the LWE problem for samples $(\mathbf{a}_i, b_i)$. Note that an adversary must solve $n^2 + n$ many discrete logarithms because the secret vector $\mathbf{s}$ is information-theoretically hidden, if less than n samples of LWE are known. Solving $N := n^2+n$ discrete logarithms in generic groups of order q takes time $\sqrt{2Nq}$ while computing a single discrete logarithm takes time $\sqrt{\pi q/2}$ [KS01, HMCD04].[4] In fact, this bound is proven to be optimal in the generic group model [Yun14]. Note, parameters for LWEE are chosen such that computing a single discrete logarithm takes time 2^κ. Hence, in order to solve the LWEE instance for $N = \mathcal{O}(\kappa^2)$, one requires time $\frac{2}{\sqrt{\pi}}\sqrt{N} \cdot 2^\kappa + 2^\kappa > 2^{\kappa+2\log(\kappa)}$. This shows that generically the concrete instance of LWEE is logarithmically harder in the security parameter κ.

[4] Solving N-many discrete logarithms is easier than applying N times a DL solver for a single instance.

4 Public-Key Encryption from LWEE

In this section we give a construction of a provably secure public-key bit encryption scheme with a reduction to the decisional learning with errors in the exponent assumption in the standard model.

4.1 Our Construction

The scheme is parameterized by positive integers $n, k, \ell < k$ and Gaussian parameters σ_s, σ_e.

KeyGen: Sample prime numbers p' and q', such that $p = 2^k p' + 1$ and $q = 2^k q' + 1$ are prime. Set $N = pq$ and $M = 2^k p' q'$. Sample $\mathbf{s} \leftarrow_R \mathsf{D}_{\sigma_s}^n$, $\mathbf{A} \leftarrow_R \mathbb{Z}_M^{n \times n}$ and $\mathbf{x} \leftarrow_R \mathsf{D}_{\sigma_e}^n$ and compute $\mathbf{b} = \mathbf{A}^\top \mathbf{s} + \mathbf{x}$. Sample $g \in \mathbb{J}_N \setminus \mathbb{QR}_N$ of order M. The public key consists of $\mathsf{pk} = (g, g^{\mathbf{A}}, g^{\mathbf{b}}, N)$, and the secret key of $\mathsf{sk} = (p, \mathbf{s})$.

Encrypt(pk, μ): To encrypt ℓ bits $\mu \in \{0, 1, \ldots 2^\ell - 1\}$ given public key pk choose $\mathbf{r} \leftarrow_R \mathsf{D}_{\sigma_s}^n$, $\mathbf{e}_0 \leftarrow_R \mathsf{D}_{\sigma_e}^n$ and $e_1 \leftarrow_R \mathsf{D}_{\sigma_e}$. Use $g^{\mathbf{A}}$, $\mathbf{r}$ and $\mathbf{e}_0$ to compute $g^{\mathbf{A}\mathbf{r}+\mathbf{e}_0}$, and $g^{\mathbf{b}}$, $\mathbf{r}$ and e_1 to compute $g^{\langle \mathbf{b}, \mathbf{r} \rangle + e_1}$. The ciphertext is $\mathbf{c}_0, c_1$ with

$$\mathbf{c_0} = g^{\mathbf{A}\mathbf{r}+\mathbf{e}_0}, \quad c_1 = g^{\langle \mathbf{b}, \mathbf{r} \rangle + e_1} \cdot g^{2^{k-\ell}\mu} .$$

Decrypt($\mathsf{sk}, (\mathbf{c}_0, c_1)$): To decrypt the ciphertext $(\mathbf{c}_0, c_1)$ given secret key $\mathsf{sk} = (p, \mathbf{s})$, first compute $g^{\langle \mathbf{s}, \mathbf{A}\mathbf{r}+\mathbf{e}_0 \rangle}$ and then $h = c_1 / g^{\langle \mathbf{s}, \mathbf{A}\mathbf{r}+\mathbf{e}_0 \rangle}$. Run Algorithm 1 to synthesize $v = \log_g(h) \mod 2^k$ and return $\left\lfloor \frac{v}{2^{k-\ell-1}} \right\rceil$.

Algorithm 1:

Input: Generator g of a group with order $p - 1 = 2^k p'$, p and k
Output: k least significant bits of $\log_g(h)$
begin
  $a = 0, B = 1$;
  for $i \in \{1, \ldots, k\}$ **do**
    $z \leftarrow \mathbb{L}(h, p)_{2^i} \mod p$;
    $t \leftarrow \mathbb{L}(g, p)_{2^i}^a \mod p$;
    if $z \neq t$ **then**
      $a \leftarrow a + B$;
    end
    $B \leftarrow 2B$;
  end
  return a
end

To show correctness of our construction we build upon two facts. First, Algorithm 1 synthesizes the k least significant bits of a discrete logarithm. The algorithm's correctness

for a modulus being a multiple of 2^k is proven in [JL13, Section 3.2]. Second, noise in the exponent does not overlap with the message. To this end, we need to bound the size of the noise.

Lemma 4.1 (adapted from [LP11][Lemma 3.1])**.** *Let c, T be positive integers such that*

$$\sigma_s \cdot \sigma_e \leq \frac{\pi}{c}\frac{T}{\sqrt{n\ln(2/\delta)}} \quad \text{and} \quad \left(c \cdot \exp(\frac{1-c^2}{2})\right)^{2n} \leq 2^{-40}\ .$$

Then, for $\mathbf{x}, \mathbf{s} \leftarrow_R \mathsf{D}^n_{\sigma_e}, \mathbf{r}, \mathbf{r}_0 \leftarrow_R \mathsf{D}^n_{\sigma_e}, e_1 \leftarrow_R \mathsf{D}_{\sigma_e}$, we have $|\langle \mathbf{x}, \mathbf{r}\rangle - \langle \mathbf{s}, \mathbf{e}_0\rangle + e_1| < T$ with probability at least $1 - \delta - 2^{-40}$.

We are now ready to prove the following theorem.

Theorem 4.2. *Let c, T be as in Lemma 4.1. Then, the decryption is correct with probability at least $1 - \delta - 2^{-40}$.*

Proof. To see that the above scheme decrypts properly the message μ, observe first that canceling out the term $g^{\langle \mathbf{s},\mathbf{u}\rangle}$ from c_1 gives the encryption of μ with some small noise term in the exponent. That is,

$$h = g^{\langle \mathbf{b},\mathbf{r}\rangle + e_1 + 2^{k-\ell}\mu - \langle \mathbf{s}, \mathbf{A}\mathbf{r} + \mathbf{e}_0\rangle} = g^{\langle \mathbf{A}^\top \mathbf{s},\mathbf{r}\rangle + \langle \mathbf{x},\mathbf{r}\rangle + e_1 - \langle \mathbf{s},\mathbf{A}\mathbf{r}\rangle - \langle \mathbf{s},\mathbf{e}_0\rangle + 2^{k-\ell}\mu} = g^{\langle \mathbf{x},\mathbf{r}\rangle - \langle \mathbf{s},\mathbf{e}_0\rangle + e_1 + 2^{k-\ell}\mu}.$$

As Algorithm 1 recovers the k least significant bits of

$$\langle \mathbf{x}, \mathbf{r}\rangle - \langle \mathbf{s}, \mathbf{e}_0\rangle + e_1 + 2^{k-\ell}\mu \mod p'q'2^k,$$

we have

$$v = \langle \mathbf{x}, \mathbf{r}\rangle - \langle \mathbf{s}, \mathbf{e}_0\rangle + e_1 + 2^{k-\ell}\mu \mod 2^k.$$

Lemma 4.1 for $T = 2^{k-\ell-1}$ shows that $\langle \mathbf{x}, \mathbf{r}\rangle - \langle \mathbf{s}, \mathbf{e}_0\rangle + e_1 < 2^{k-\ell-1}$, and therefore

$$\left\lfloor \frac{v}{2^{k-\ell}} \right\rceil = \left\lfloor \frac{\langle \mathbf{x}, \mathbf{r}\rangle - \langle \mathbf{s}, \mathbf{e}_0\rangle + e_1 + 2^{k-\ell}\mu \mod 2^k}{2^{k-\ell}} \right\rceil = \left\lfloor \frac{\langle \mathbf{x}, \mathbf{r}\rangle - \langle \mathbf{s}, \mathbf{e}_0\rangle + e_1}{2^{k-\ell}} \right\rceil + \mu = \mu$$

□

4.2 Security Analysis

We now proceed to the security analysis. In the forthcoming Section 4.3, we discuss concrete parameter choices for different security levels

Theorem 4.3. *Let $\mathbb{G} = \langle g\rangle$ be the cyclic group generated by g. If $\mathsf{DLWEE}_{\mathbb{G},n,n+1,q,\mathsf{D}_{\sigma_e}}(\mathsf{D}_{\sigma_s})$ is (t,ϵ)-hard, then the above cryptosystem is $(t, 2\epsilon)$-indistinguishable against chosen plaintext attacks.*

Proof. In a high level, our proof works as follows. Instead of showing IND-CPA security via a direct argument we show that the distribution $(\mathsf{pk}, \mathbf{c}_0, c_1)$ is indistinguishable from the uniform distribution over $(\mathbb{G}^{n\times n} \times \mathbb{G}^{2n+1})$. That is, a ciphertext $(\mathbf{c}_0, c_1)$ under public key pk appears completely random to an adversary. This holds, in particular, in the IND-CPA experiment when the adversary chooses the underlying plaintext. We prove the theorem via a series of hybrid arguments, Hybrid_0 to Hybrid_2, where in each consecutive argument we make some slight changes with the provision that the adversary notices the changes with negligible probability only. In the following, we use the abbreviations $\mathbf{u} = \mathbf{A}\mathbf{r} + \mathbf{e}_0$ and $v = \langle \mathbf{b}, \mathbf{r} \rangle + e_1 + 2^{k-\ell}\mu$.

Hybrid_0: In this hybrid we consider the original distribution of the tuple

$$(\mathsf{pk}, (\mathbf{c}_0, \mathbf{c}_1)) = (g^{\mathbf{A}}, g^{\mathbf{b}}, g^{\mathbf{u}}, g^{v}).$$

Hybrid_1: In this hybrid we modify the distribution and claim

$$(g^{\mathbf{A}}, g^{\mathbf{b}}, g^{\mathbf{u}}, g^{v}) \approx_c (g^{\mathbf{A}'}, g^{\mathbf{b}'}, g^{\mathbf{A}'\mathbf{r}+\mathbf{e}_0}, g^{\langle \mathbf{b}'\cdot\mathbf{r}\rangle+e_1} \cdot g^{2^{k-\ell}\mu})$$

for a uniformly sampled elements $g^{\mathbf{A}'}, g^{\mathbf{b}'} \in \mathbb{G}^{n\times n} \times \mathbb{G}^n$. We argue that any successful algorithm distinguishing between Hybrid_0 and Hybrid_1 can be easily turned into a successful distinguisher $\mathcal{B}$ in the $\mathsf{DLWEE}_{\mathbb{G},n,n,q,\mathsf{D}_{\sigma_e}}(\mathsf{D}_{\sigma_s})$ problem. The DLWEE-adversary $\mathcal{B}$ is given as challenge the tuple $(g^{\mathbf{A}}, g^{\mathbf{b}})$ and is asked to decide whether there exist vectors $\mathbf{s} \leftarrow_R \mathsf{D}_{\sigma_s}$, $\mathbf{x} \leftarrow_R \mathsf{D}^n_{\sigma_e}$ such that $g^{\mathbf{b}} = g^{\mathbf{A}^\top\mathbf{s}+\mathbf{x}}$ or $g^{\mathbf{b}}$ was sampled uniformly from $\mathbb{G}^n$.

Let $\Pr[\mathsf{Hybrid}_i(t)]$ denote the probability of any algorithm with runtime t to win the IND-CPA experiment in hybrid i. Then, we have

$$\Pr[\mathsf{Hybrid}_0(t)] \leq \Pr[\mathsf{Hybrid}_1(t)] + \mathsf{Adv}^{\mathsf{DLWEE}}_{\mathbb{G},n,n,q,\mathsf{D}_{\sigma_e},\mathsf{D}_{\sigma_s}}(t)\ .$$

Hybrid_2: In this hybrid we modify the distribution and claim

$$(g^{\mathbf{A}'}, g^{\mathbf{b}'}, g^{\mathbf{A}'\mathbf{r}+\mathbf{e}_0}, g^{\langle \mathbf{b}'\cdot\mathbf{r}\rangle+e_1} \cdot g^{2^{k-1}\mu}) \approx_c (g^{\mathbf{A}''}, g^{\mathbf{b}''}, g^{\mathbf{u}'}, g^{v'} \cdot g^{2^{k-1}\mu})$$

for a uniformly sampled elements $g^{\mathbf{A}''}, g^{\mathbf{b}''}, g^{\mathbf{u}'}, g^{v'} \cdot g^{\mu} \in \mathbb{G}^{(n+1)\times n} \times \mathbb{G}^{n+1}$. We argue that any successful algorithm distinguishing between Hybrid_1 and Hybrid_2 can be easily turned into a successful distinguisher $\mathcal{B}$ in the $\mathsf{DLWEE}_{\mathbb{G},n,n+1,q,\mathsf{D}_{\sigma_e}}(\mathsf{D}_{\sigma_s})$ problem. Note that $g^{\mathbf{b}'}, g^{\langle \mathbf{b}'\cdot\mathbf{r}\rangle+e_1}$ is an additional sample from the LWEE distribution from which $g^{\mathbf{A}'}, g^{\mathbf{A}'\mathbf{r}+\mathbf{e}_0}$ is sampled.

We have

$$\Pr[\mathsf{Hybrid}_1(t)] \leq \Pr[\mathsf{Hybrid}_2(t)] + \mathsf{Adv}^{\mathsf{DLWEE}}_{\mathbb{G},n,n+1,q,\mathsf{D}_{\sigma_e},\mathsf{D}_{\sigma_s}}(t)\ .$$

Note that now all exponents are uniformly distributed, and, in particular, independent of μ and thus, independent of b in the IND-CPA game. Hence, any algorithm has in Hybrid_2 exactly a success probability of $1/2$.

This completes the proof of semantic security. □

4.3 Candidate Instantiations of our Encryption Scheme

We give three possible instantiation to derive a system with short key sizes, post-quantum security or double hardness. Throughout this section we instantiate our scheme such that the encryption scheme from Section 4.1 encrypts only a single bit. Wlog, parameters can easily be upscaled to many bits.

The Classical Way. We obtain the shortest key and ciphertext sizes when instantiating LWEE parameters such that the underlying DRP is intractable, and neglecting the hardness of the underlying LWE.[5] In Appendix B.1 we recall some groups where we believe DRP is hard to solve. Our encryption scheme works in the group $\mathbb{J}_N := \{x \in \mathbb{Z}_N \ : \ \mathbb{J}(x, N) = 1\}$ for $N = pq$ with p, q being k-safe primes. In fact, we can even take safe primes p, q (i.e., $k = 1$) since we do not need any noise in the exponent if we neglect the underlying LWE hardness. Thus, we embed the message to the least significant bit in the exponent. For this reason, we can sample $g \leftarrow_R \mathbb{J}_N/\mathbb{QR}_N$ where $\langle g \rangle$ has order $2p'q'$. Since the LWE instance within LWEE is not an issue here we select $n = m = 1$, $\sigma_s = \infty$ and $\sigma_e = 0$.

We obtain 80-bit security for the underlying DRP problem if we choose safe primes p and q such that $\log p = \log q = 565$ (see Table 3 in Appendix B.1). Table 1 lists possible key sizes for our encryption scheme. Recall that the public key consists of $\mathsf{pk} = (g, g^{\mathbf{A}}, g^{\mathbf{b}}, k, N)$ (i.e., 4 group elements if we fix $k = 1$) and the secret key of $\mathsf{sk} = (p, \mathbf{s})$.

Sizes / Security	$\approx$ 80-bit	$\approx$ 128-bit	$\approx$ 256-bit
public-key size	0.565 kbytes	1.500 kbytes	7.500 kbytes
secret-key size	0.212 kbytes	0.563 kbytes	2.813 kbytes
ciphertext size	0.283 kbytes	0.750 kbytes	3.750 kbytes

Table 1: Key sizes for our encryption scheme basing security on DRP.

The Post-Quantum Way. Here we give example instantiations of our encryption scheme when it is based on a presumably quantum-resistant LWEE assumption. That is, we select parameters such that the underlying LWE assumption is intractable without relying on the hardness of DRP. For this, we modify the scheme slightly by choosing fixed values for p' and q' instead of sampling. A good choice is $k = 15$, since it allows to choose $p' = 2$ and $q' = 5$, which are very small prime numbers such that $2^k p' + 1$ and $2^k q' + 1$ are prime. For the LWE modulus, this leads to $M = 2^k p'q' = 327680$. Like Lindner and Peikert [LP11], we choose the Gaussian parameter such that the probability of decoding errors is bounded by 1%. We choose furthermore the same parameter for error and secret distribution (i.e. $\sigma_s = \sigma_e = \sigma$), since a standard argument reduces LWE with arbitrary secret to LWE with secret chosen according to the error distribution. For this choice of k, p' and q', we obtain 80-bit security

[5] Admittedly the keys are only shorter for 80-bit security. This is the case, as there exists subexponential attacks against DL in our group.

by choosing $n = 240$ and $\sigma = 33.98$. Table 2 lists the key sizes when our encryption scheme is instantiated with parameters corresponding to Table 4 in Appendix B.2.

Sizes / Security	$\approx$ 80-bit	$\approx$ 128-bit	$\approx$ 256-bit
public-key size	235 kbytes	417 kbytes	1233 kbytes
secret-key size	0.976 kbytes	1.302 kbytes	2.237 kbytes
ciphertext size	0.980 kbytes	1.306 kbytes	2.241 kbytes

Table 2: Key sizes for our encryption scheme basing security on LWE.

The Hardest Way (Double-Hardness). The most secure instantiation of our encryption is such that even if one of the problems DRP or LWE is efficiently solvable at some point, our encryption scheme remains semantically secure. Selecting parameters for double hardness, however, is non-trivial.

To select appropriate parameters for the case of double hardness, we apply the following approach: For a given security level (say $\kappa = 80$), we select N such that the Number Field Sieve needs at least 2^κ operations to factor N. Following Table 3, we choose $\log N = 1130$. Since factoring N must also be hard for McKee-Pinch's algorithm, which works well when (p-1) and (q-1) share common factor, k must be chosen such that $N^{1/4}2^{-k} \geq 2^\kappa$, i.e. $k \leq \frac{\log(N)}{4} - \kappa$. This leads to $k = 203$. Given N and k, we can calculate the sizes of the primes $\log(p') \approx \log(q') \approx 362$ and $log(p) \approx \log(q) \approx 565$ and the LWE modulus $\log(M) \approx 927$. Taking $n = 67000$ and $\sigma = 2^{97}$ from Table 5, Lemma 4.1 shows that the algorithm decrypts correctly with high probability. Other security levels κ (e.g., $\kappa = 128$ and $\kappa = 256$) can be achieved with the LWE instances depicted in Table 5 in Appendix B.2. We note that extrapolation to such large dimensions hardly give a good estimation for the hardness of LWE. Hence, one has to take these parameters for double hardness with care. The corresponding key and ciphertext sizes of our scheme are admittedly very large and unpractical, but they shall serve as a feasibility of double hardness in the first place.

Acknowledgements

The authors would like to thank Steven Galbraith and Dan Bernstein to point to a bug in a previous version of the paper, and the attendees of the *Cryptography Workshop in Oberwolfach* for their valuable feedback. Özgür Dagdelen is supported by the German Federal Ministry of Education and Research (BMBF) within EC-SPRIDE.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology EU-*

ROCRYPT 2010, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer Berlin Heidelberg, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Tal Rabin, editor, *Advances in Cryptology CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 98–115. Springer Berlin Heidelberg, 2010.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer Berlin Heidelberg, 2009.

[ADVW13] Shweta Agrawal, Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. On continual leakage of discrete log representations. 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 II*, pages 401–420, 2013.

[AFG13] Martin R. Albrecht, Robert Fitzpatrick, and Florian Göpfert. On the efficacy of solving lwe by reduction to unique-svp. Cryptology ePrint Archive, Report 2013/602, 2013. `http://eprint.iacr.org/`.

[Bab86] L. Babai. On Lovasz lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Bac84] Eric Bach. Discrete logarithms and factoring. Technical Report UCB/CSD-84-186, EECS Department, University of California, Berkeley, Jun 1984.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 575–584. ACM, 2013.

[BM92] Ernest F. Brickell and Kevin S. McCurley. An interactive identification scheme based on discrete logarithms and factoring. *Journal of Cryptology*, 5(1):29–39, 1992.

[Bon98] Dan Boneh. The decision Diffie-Hellman problem. In JoeP. Buhler, editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer Berlin Heidelberg, 1998.

[Bra93] Stefan A. Brands. An efficient off-line electronic cash system based on the representation problem. Technical report, Amsterdam, The Netherlands, The Netherlands, 1993.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer Berlin Heidelberg, 2012.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 97–106. IEEE, 2011.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer Berlin Heidelberg, 2011.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Jr. Kaliski, BurtonS., editor, *Advances in Cryptology CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer Berlin Heidelberg, 1997.

[CCK+13] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, MoonSung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In Thomas Johansson and PhongQ. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335. Springer Berlin Heidelberg, 2013.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer Berlin Heidelberg, 2010.

[CN11] Yuanmi Chen and PhongQ. Nguyen. BKZ 2.0: Better lattice security estimates. In DongHoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin Heidelberg, 2011.

[Cop96] Don Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In Ueli Maurer, editor, *Advances in Cryptology EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 178–189. Springer Berlin Heidelberg, 1996.

[Cop97] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.

[CS06] An Commeine and Igor Semaev. An algorithm to solve the discrete logarithm problem with the number field sieve. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 174–190. Springer Berlin Heidelberg, 2006.

[DG06] Alexander W. Dent and Steven D. Galbraith. Hidden pairings and trapdoor DDH groups. In Florian Hess, Sebastian Pauli, and Michael Pohst, editors, *Algorithmic Number Theory*, volume 4076 of *Lecture Notes in Computer Science*, pages 436–451. Springer Berlin Heidelberg, 2006.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[DV14] Özgür Dagdelen and Daniele Venturi. A second look at Fischlin's transformation. In *Progress in Cryptology - AFRICACRYPT 2014 - 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings*, pages 356–376, 2014.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An algebraic framework for Diffie-Hellman assumptions. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 129–147, 2013.

[FMR99] G. Frey, M. Müller, and H.-G. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *Information Theory, IEEE Transactions on*, 45(5):1717–1719, 1999.

[FPSZ06] Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Sebastien Zimmer. Hardness of distinguishing the MSB or LSB of secret keys in diffie-hellman schemes. In *ICALP*, pages 240–251. Springer, 2006.

[FS97] R. Fischlin and C.P. Schnorr. Stronger security proofs for RSA and Rabin bits. In Walter Fumy, editor, *Advances in Cryptology EUROCRYPT 97*, volume 1233 of *Lecture Notes in Computer Science*, pages 267–279. Springer Berlin Heidelberg, 1997.

[Gen09] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford, CA, USA, 2009.

[GH11] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 107–109, Washington, DC, USA, 2011. IEEE Computer Society.

[Gir91] Marc Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In IvanBjerre Damgrd, editor, *Advances in Cryptology EUROCRYPT 90*, volume 473 of *Lecture Notes in Computer Science*, pages 481–486. Springer Berlin Heidelberg, 1991.

[GKPV08] Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *In ICS. 2010. [GPV08] [GRS08*, 2008.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. ACM.

[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*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[HK09] Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 637–653. Springer Berlin Heidelberg, 2009.

[HMCD04] Yvonne Hitchcock, Paul Montague, Gary Carter, and Ed Dawson. The efficiency of solving multiple discrete logarithm problems and the implications for the security of fixed elliptic curves. *International Journal of Information Security*, 3(2):86–98, 2004.

[JL13] Marc Joye and Benot Libert. Efficient cryptosystems from 2^k-th power residue symbols. In Thomas Johansson and PhongQ. Nguyen, editors, *Advances in Cryptology EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 76–92. Springer Berlin Heidelberg, 2013.

[JS13] Tibor Jager and Jörg Schwenk. On the analysis of cryptographic assumptions in the generic ring model. *J. Cryptology*, 26(2):225–245, 2013.

[KS01] Fabian Kuhn and Rene Struik. Random walks revisited: Extensions of pollard's rho algorithm for computing multiple discrete logarithms. In *8th Annual Workshop on Selected Areas in Cryptography (SAC), Toronto, Ontario, Canada*, August 2001.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 703–720, 2009.

[KY05] Aggelos Kiayias and Moti Yung. Efficient secure group signatures with dynamic joins and keeping anonymity against group managers. In Ed Dawson and Serge Vaudenay, editors, *Progress in Cryptology Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 151–170. Springer Berlin Heidelberg, 2005.

[LL95] Chae Hoon Lim and Pil Joong Lee. Security and performance of server-aided RSA computation protocols. In *CRYPTO*, pages 70–83, 1995.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer Berlin Heidelberg, 2009.

[LN13] Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In Ed Dawson, editor, *Topics in Cryptology CT-RSA 2013*, volume 7779 of *Lecture Notes in Computer Science*, pages 293–309. Springer Berlin Heidelberg, 2013.

[LP11] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer Berlin Heidelberg, 2011.

[LV00] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 446–465. Springer Berlin Heidelberg, 2000.

[Mic01] Daniele Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*, pages 126–145, 2001.

[Mil75] Gary L. Miller. Riemann's hypothesis and tests for primality. In *Proceedings of seventh annual ACM symposium on Theory of computing*, STOC '75, pages 234–239, New York, NY, USA, 1975. ACM.

[MOV93] A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *Information Theory, IEEE Transactions on*, 39(5):1639–1646, 1993.

[MP98] JF McKee and RGE Pinch. Further attacks on server-aided RSA cryptosystems. *Unpublished manuscript*, 1998.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 18–35, 2009.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.

[PH78] S.C. Pohlig and M.E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.

[Pol78] John M Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of computation*, 32(143):918–924, 1978.

[Reg04] Oded Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[Reg10] Oded Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010.

[Sch90] C.P. Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology CRYPTO 89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer New York, 1990.

[SE94] C.P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1-3):181–199, 1994.

[Seu13] Yannick Seurin. New constructions and applications of trapdoor DDH groups. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography PKC 2013*, volume 7778 of *Lecture Notes in Computer Science*, pages 443–460. Springer Berlin Heidelberg, 2013.

[Sha07] Hovav Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. `http://eprint.iacr.org/`.

[Sho94] Peter Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, 1994 Proceedings.*, pages 124–134, 1994.

[Sho97a] Peter Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Sho97b] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266, 1997.

[Yun14] Aaram Yun. Generic hardness of the multiple discrete logarithm problem. Cryptology ePrint Archive, Report 2014/637, 2014. `http://eprint.iacr.org/`.

A Justifying the DRP assumption in the Generic Group Model

To gain confidence about the decisional representation problem we justify the assumption in the generic group model [Sho97b]. We emphasize however that proofs in the generic group model have to be considered with much care. They do not provide a lower complexity bound in any specific group. It is also known that there exist operations that are easy to compute when instantiated in a specific group, but they are hard in the generic group [JS13]. To sum up, proofs in the generic group model should be treated as proofs in related idealized, but imperfect models, such as the random oracle or ideal cipher model: They give stronger confidence than no security argument at all.

A.1 Complexity Lower Bound

In the generic model, elements in $\mathbb{G}$ are encodings of random strings. Algorithms are not given "actual" representations of the group elements, but rather operate via their "handles". Admissible group operations are performed by oracles that maintain a list of handles and encodings. Note that this formalism allows an algorithm for testing equality, since two elements are identical if they have the same handle. Two oracles perform operations between the elements, computing the group actions (multiplication and exponentiation) in $\mathbb{G}$. We also provide an oracle which allows an algorithm to make use of a $(\ell-1)$-DRP solver. We show thereby that the ℓ-decisional representation assumption becomes strictly weaker by increasing ℓ. For sake of simplicity, in the proof below we assume that $m = 1$. We note that one can easily extend it to the case where multiple samples ($m > 1$) are given to an adversary.

Let $\xi : \mathbb{Z}_q^* \to \{0,1\}^*$ be the encodings of elements in $\mathbb{G}$ which maps all $X \in \mathbb{Z}_q^*$ to the string representation $\xi(X)$ of $g^X \in \mathbb{G}$. Let $\{a_i\}_{i=1}^{\ell} \xleftarrow{r} \mathbb{Z}_q^*$, $\{x_i\}_{i=1}^{\ell} \xleftarrow{r} \mathbb{Z}_q^*$, $c \xleftarrow{r} \mathbb{Z}_q^*$, $T_b = g^{\sum_i a_i x_i}$, $T_{1-b} = g^c$ for $b \xleftarrow{r} \{0,1\}$. We show that no generic algorithms $\mathcal{A}$ given $g, g^{\{a_i\}_{i=1}^{\ell}}, g^{\{x_i\}_{i=1}^{\ell}}, T_0, T_1$ making at most q oracle queries can guess the value of b with probability greater that $1/2 + O((q+2\ell+2)^2/p)$ where p is the group order. Note that $g^{\{a_i\}}$ captures the role of the generator g_i in the ℓ-decisional representation problem.

Theorem A.1. *Let $\mathcal{A}$ be an algorithm that solves the ℓ-Decisional Representation Problem in a group of order p in the generic group model while making at most q oracle queries. Then, its success probability is upper bounded by $\frac{1}{2} + \frac{(q+2\ell+2)^2}{p}$.*

Proof. Assume that ξ is a random encoding function for $\mathbb{G}$ where $\mathbb{G}$ is of prime order p. We show that if $\mathcal{A}$ makes at most q oracle queries, then

$$Pr\left[\mathcal{A}\left(\begin{array}{c} p, \xi(1), \{\xi(a_i)\}_{i=1}^{\ell}, \{\xi(x_i)\}_{i=1}^{\ell}, \\ \xi(t_0), \xi(t_1) \end{array}\right) = b \;\middle|\; \begin{array}{c} \{a_i\}_{i=1}^{\ell}, \{x_i\}_{i=1}^{\ell} \xleftarrow{r} \mathbb{Z}_p, \\ c \xleftarrow{r} \mathbb{Z}_p, b \xleftarrow{r} \{0,1\}, \\ t_b = \sum_i a_i x_i, t_{1-b} = c \end{array}\right] \leq \frac{1}{2} + \frac{(q+2\ell+2)^2}{p} .$$

We will simulate the view of $\mathcal{A}$. The simulation proceeds as follows. We maintain a list of pairs, $L = \{(F_k, \xi_k) : k = 0, \ldots, \tau_1 - 1\}$, under the invariant that at the τ^{th}-step, it holds that $\tau_1 = \tau + (2\ell + 3)$. Informally, list L contains the handles and encodings of the group elements during the game. More precisely, we keep track of elements handled by $\mathcal{A}$ as polynomials F_k in the commutative ring $R = \mathbb{Z}_p[A_1, \ldots, A_\ell, X_1, \ldots, X_\ell, C]$; the $\xi_k \in \{0,1\}^*$ are random encodings of the polynomials.

We initialize at step $\tau = 0$ the list L by setting $F_0 = 1$, $\{F_i = A_i\}_{i=1}^{\ell}$, $\{F_{\ell+i} = X_i\}_{i=1}^{\ell}$, $F_{2\ell+1} = T_0$, and $F_{2\ell+2} = T_1$. Note that all polynomials except T_b are of degree at most 1. For the corresponding encoding ξ_k we choose distinct strings from $\{0,1\}^*$. We assume that $\mathcal{A}$ makes only queries on strings previously retrieved from her oracles, since we can make them arbitrarily hard to guess. We then begin the game by giving $\mathcal{A}$ the encodings ξ_k for $k \in [1, 2\ell + 2]$ while keeping the corresponding internal handles F_k secret. The oracle queries as simulated as follows.

Multiplication/Division. A query consists of two operands ξ_i, ξ_j with $1 \leq i, j \leq \tau_1$ and a flag bit interpreted as multiplication or division of the two group elements. Let $\tau_1' = \tau_1 + 1$. To perform the group operation, perform the polynomial addition or subtraction $F_{\tau_1'} = F_i \pm F_j$ depending on whether multiplication or division is requested. If the result $F_{\tau_1'} = F_k$ for some $k \leq \tau_1$ matches a polynomial already stored in the list L, then set $\xi_{\tau_1'} \leftarrow \xi_k$. Otherwise, sample a fresh random string from $\{0,1\}^* \backslash \{\xi_1, \ldots, \xi_{\tau_1}\}$. Finally, add $(F_{\tau_1'}, \xi_{\tau_1'})$ to list L.

Exponentiation. A query consists of one operand ξ_i with $i \in [\tau_1]$ and an integer $r \in \mathbb{Z}_p$ interpreted as raising the group element behind ξ_i to the power of r. Let $\tau_1' = \tau_1 + 1$. To perform the group operation, perform a scalar multiplication to the polynomial F_i, i.e., $F_{\tau_1'} = F_i \cdot r$. If the result $F_{\tau_1'} = F_k$ for some $k \leq \tau_1$ matches a polynomial already stored in the list L, then set $\xi_{\tau_1'} \leftarrow \xi_k$. Otherwise, sample a fresh random string from $\{0,1\}^* \backslash \{\xi_1, \ldots, \xi_{\tau_1}\}$. Finally, add $(F_{\tau_1'}, \xi_{\tau_1'})$ to list L.

Decide $\ell - 1$-DRP. A query consists of $2 \cdot (\ell - 1) + 1$ encodings $\xi_{i_1}, \ldots \xi_{i_{2\cdot(\ell-1)+1}}$ with $i_j < \tau_1$ for all j. The oracle checks whether the group elements for those encoding represent a genuine $(\ell - 1)$-DRP tuple. To this end, check whether $F_{i_{2\cdot(\ell-1)+1}} = \sum_{j=1}^{\ell-1} F_{i_j} \cdot F_{i_{j+\ell-1}}$. If so, return 1; else return 0.

After at most q queries $\mathcal{A}$ eventually outputs a guess b'. At this point, we select random $a_1, \ldots, a_\ell, x_1, \ldots, x_\ell, c \xleftarrow{r} \mathbb{Z}_p$ and set $t_b = \sum_i a_i x_i$ and $t_{1-b} = c$. For $i = 1, \ldots, \ell$, we set $A_i = a_i$, $X_i = x_i$, $T_0 = t_0$ and $T_1 = t_1$. It is easy to see that the simulation is perfect unless the chosen random values for the variables $A_1, \ldots, A_\ell, X_1, \ldots, X_\ell, C$ result in an equality relation between intermediate values that is not an equality of polynomials. In other words, the simulation is perfect unless for some i, j the following holds:

$$F_i(a_1, \ldots, a_\ell, x_1, \ldots, x_\ell, c) - F_j(a_1, \ldots, a_\ell, x_1, \ldots, x_\ell, c) = 0 \wedge F_i \neq F_j \tag{1}$$

We call the adversary $\mathcal{A}$ is successful, if she finds such a collision or if she guesses correctly b. Note that the random variables are initialized by values all independent of each other except T_b, which takes the value $\sum_i a_i x_i$. Hence, without a collision as described above, the probability to guess b is (at most) $1/2$.

We now bound the probability that such a collision occurs, denoted by the event fail. When event fail occurs, then our responses to $\mathcal{A}$'s queries deviate from the real oracles' responses when the input tuple is derived from $a_1, \ldots, a_\ell, x_1, \ldots, x_\ell, c$. We need to argue that the adversary is unable to engineer the above equality, so that they can occur only due to an unfortunate choice of $a_1, \ldots, a_\ell, x_1, \ldots, x_\ell, c$. Note that only $T_b = \sum_i a_i x_i$ is dependent on the other values. Thus, an independent collision can be caused only if $\mathcal{A}$ manages to produce a polynomial that is a multiple of $\sum_i A_i X_i$, say $\alpha \sum_i A_i X_i$ for some $\alpha \in \mathbb{Z}_p$.

First, observe that the adversary can manipulate the polynomials F_k through additions and subtractions (as a result of the interplay with the multiplication oracle); thus, the degree of resulting polynomials remain equal. The oracle for exponentiation does not increase the degree either since the group elements are raised by scalars. Hence, all polynomials remain of same degree through the oracles. Note that the $(\ell - 1)$-DRP oracle does not give the adversary any new encodings.

Given the available operations, the adversary is unable to generate a polynomial F_k out of given polynomials $F_1, \ldots, F_{2\ell}$ which contains at least one of the monomials $\alpha A_i X_i$ for any $\alpha \in \mathbb{Z}_p$ since all those polynomials are of degree 1. Unfortunately, this is necessary to synthesize a multiple of T_b. Since the polynomial difference in (1) are linear combinations of the arguments, it is easy to see that the adversary will not cause to trivially cancel out identical multiples of monomials $\alpha A_i X_i$.

It remains to bound the probability that a random choice of values $a_1, \ldots, a_\ell, x_1, \ldots, x_\ell, c$ will cause two distinct polynomials F_i, F_j, for $i \neq j$, to have the same image. All polynomials F_k have degree at most 1. Using the Schwartz-Zippel Lemma, the probability that $F_i() = F_j()$ is $1/p$ over the choice of values. Since the list is initially set up with $2\ell + 2$ elements and the adversary makes at most q oracle queries, a sum over all pairs of entries gives a lower bound on the success probability:

$$Pr[\mathsf{fail}] \leq \binom{q + 2\ell + 2}{2} \frac{1}{p} \leq \frac{(q + 2\ell + 2)^2}{p}$$

□

B Hardness of Assumptions

Here, we review the decisional representation and learning with errors problem. Our aim is to give empirical arguments of the hardness of learning with errors in the exponent and justify the parameter choices in Section 4.3.

B.1 Hardness of the Decisional Representation Problem

Little is known about the decisional representation problem. Proposition 2.6 shows that $(\ell+1)$-DRP for any ℓ is generically at least as hard as the ℓ-DRP problem. As ℓ-DRP for $\ell = 1$ coincides with DDH, we lay our argumentation on the well-studied decisional Diffie-Hellman problem.

Decisional Diffie-Hellman. We start by recalling groups in which the DDH problem is believed to be instractable. Boneh gives several examples in [Bon98]. Among them are the following ones:

1. In the cyclic subgroup $\mathbb{QR}(p) \subset \mathbb{Z}_p^*$ of quadratic residues in $\mathbb{Z}_p^*$, where $p = 2p' + 1$ with p and p' both prime, DDH is believed to be intractable.
2. Let $N = pq$ for primes $p, q, \frac{(p-1)}{2}, \frac{(q-1)}{2}$. The cyclic subgroup T in $\mathbb{Z}_N^*$ of non-prime order $(p-1)(q-1)/2$ is believed to be a DDH-hard group. The same is claimed for subgroup $\mathbb{QR}_N \subset \mathbb{Z}_N^*$ of order $(p-1)(q-1)/4$, which even holds if p, q is known, and thus, the hardness of DDH is independent of the factorization [KY05].
3. The elliptic curve $E_{a,b}/\mathbb{F}_p$ where $|E_{a,b}|$ and p are prime is believed to resist against DDH attacks.

Note that one might believe that the multiplicative group $\mathbb{Z}_p^*$ with prime p is a safe choice. However, this group has an even order which is also publicly known. Hence, one can evaluate the Legendre symbol on g^a and g^b and compare the result with the given challenge g^c. This gives a significant non-negligible advantage to a distinguisher. Moreover, the group of signed quadratic residues $\mathbb{QR}_N^+ := \{|x| \ : \ x \in \mathbb{QR}_N\}$, introduced in [FS97] and revisited in [HK09] is publicly recognizable and thus non DDH-hard.

Trapdoor Decisional Diffie-Hellman. While many cryptographic applications can be instantiated in the above groups, our encryption scheme in Section 4 requires a special DDH-hard group where DDH is easy given a secret trapdoor. The requirement is reminiscent of trapdoor decisional Diffie-Hellman (TDDH) groups, introduced by Dent and Galbraith [DG06] and studied further by Seurin [Seu13]. Informally, TDDH groups satisfy two properties: (i) the DDH problem is assumed to be hard without a trapdoor, (ii) DDH becomes easy but CDH remains hard given a trapdoor. Thus, anyone in possession of the trapdoor is able to efficiently solve the DDH problem. We remark that for our construction groups satisfying property (i) suffice, and we do not necessarily require hardness of CDH.

Looking at TDDH groups, there are several candidates:

1. Dent and Galbraith [DG06] gave two constructions based on hidden pairings. Here, the trapdoor permits to compute pairings on a specific elliptic curve what is assumed to be infeasible without the trapdoor. One such construction of a TDDH is as follows. Let $N = pq$ be an Blum integer, i.e., the product of two primes $p \equiv q \equiv 3 \mod 4$, where there exists two large primes p' and q' such that $p'|(p+1)$ and $q'|(q+1)$. The order of an elliptic curve $E : y^2 = x^3 + x$ over the ring $\mathbb{Z}_N$ is $|E(\mathbb{Z}_N)| = (p+1)(q+1)$. The group $E(\mathbb{Z}_N)$ with the generator point $P = (x_P, y_P) \in E(\mathbb{Z}_N)$ of order $p'q'$ is assumed DDH-hard. However, if one is given the trapdoor $\tau = (p, p', q, q')$, one can solve the DDH problem by the Chinese Remainder Theorem. A tuple $(A, B, C) \in E(\mathbb{Z}_N)^3$ is a true DDH tuple iff the elements reduce modulo p and q to valid tuples in the subcurve $E(\mathbb{F}_p)$ and $E(\mathbb{F}_q)$. Those two checks can be performed with the knowledge of the trapdoor using Weil or Tate pairing [MOV93, FMR99]. In fact, given the trapdoor one can also efficiently test subgroup memberships. Dent and Galbraith [DG06] also consider an elliptic curve E over $\mathbb{F}_{2^{mn}}$ with mn being odd. Again, a hidden pairing allows one to solve DDH with the knowledge of a trapdoor.

2. Seurin [Seu13] continues the study and identifies additional trapdoor groups. Let $N = pq$ where p, q are safe primes, i.e., p and q are of the form $p = 2p' + 1$ and $q = 2q' + 1$ where p', q' are prime. The DDH problem in the group $\mathbb{QR}_{N^2}$ of quadratic residues modulo N^2 is hard given the description of $\mathbb{Z}_N^*$ if factoring N is hard. The use of a trapdoor $\tau = (p, q)$ which is the factorization of N enables to solve the DDH efficiently. $\mathbb{QR}_{N^2}$ is a cyclic group of order $\mathsf{ord}(\mathbb{QR}_{N^2}) = Np'q'$.

3. Let N be as before. The subgroup $\mathbb{J}_N$ of $\mathbb{Z}_N^*$ consists of all elements $x \in \mathbb{Z}_N^*$ such that $\mathbb{J}(x, N) = 1$. This subgroup has order $\mathsf{ord}(\mathbb{J}_N) = \phi(N)/2 = 2p'q'$. Moreover, $\mathbb{J}_N$ is cyclic because all prime factors of $\phi(N)/4 = p'q'$ are (pairwise) distinct [HK09]. Given the description of $\mathbb{J}_N$ with generator $g \in \mathbb{J}_N$, it is assumed one cannot solve the DDH problem in $\mathbb{J}_N$ without knowledge of the factorization of N. The trapdoor here is thus defined as $\tau = (p, q)$ or $\tau = \mathsf{ord}(\mathbb{J}_N)$. The assumption known as the quadratic residues problem appeared first in the security proof of the Goldwasser and Micali cryptosystem [GM82]. Joye and Libert [JL13] generalize the Goldwasser-Micali encryption scheme to groups $\mathbb{J}_N$ where p and q are *k-quasi-safe-primes*. That is, p (resp. q) are of the form $p = 2^k p' + 1$ (resp. $q = 2^k q' + 1$) where p, p', q, q' are all prime. They assume that without knowing the factorization of N, random elements of $\mathbb{QR}_N$ are computationally indistinguishable from elements in $\mathbb{J}_N / \mathbb{QR}_N$. This assumption is believed to hold for the group even when the distinguisher is given k.

Parameters. For all the above groups, there exists no specialized DDH distinguisher. In fact, the best algorithms to solve the DDH problem is to solve the DL problem in that group. Some groups above are assumed to be hard only if also factoring a composite number $N = pq$ of two large primes is hard or the quadratic residue assumption holds.

In Table 3 we give instantiations for four of the above groups for different security levels. We select

Security / Group	$E_{a,b}/\mathbb{F}_p$ $a, b \leftarrow_R \mathbb{F}_p$	$\mathbb{QR}(p) \subset \mathbb{Z}_p^*$ p safe prime	$\mathbb{J}_N \subset \mathbb{Z}_N^*$ p, q safe primes	$\mathbb{J}_N \subset \mathbb{Z}_N^*$ p, q k-safe-primes
80-bit	$\log p \approx 160$	$\log p \approx 1130$	$\log N = 1130$ $\log p \approx \log q \geq 160$	$\log N = 1130$ with $k < 202$ $\log p \approx \log q \geq 565$
128-bit	$\log p \approx 256$	$\log p \approx 3000$	$\log N = 3000$ $\log p \approx \log q \geq 256$	$\log N = 3000$ with $k < 622$ $\log p \approx \log q \geq 1500$
256-bit	$\log p \approx 512$	$\log p \approx 15000$	$\log N = 15000$ $\log p \approx \log q \geq 512$	$\log N = 15000$ with $k < 3494$ $\log p \approx \log q \geq 7500$

Table 3: Example instantiation of some DDH-hard groups for different security levels

(a) the elliptic curve $E_{a,b}/\mathbb{F}_p$ where $|E_{a,b}|$ and p are prime,

(b) the subgroup $\mathbb{QR}(p)$ of quadratic residues in $\mathbb{Z}_p^*$, where $p = 2p' + 1$ with p and p' both prime,

(c) the subgroup $\mathbb{J}_N$ of $\mathbb{Z}_N^*$ defined as $\{x \in \mathbb{Z}_N^* \mid \mathbb{J}(x, N) = 1\}$ where $N = pq$ and p, q being safe primes, and

(d) the subgroup $\mathbb{J}_N$ of $\mathbb{Z}_N^*$ defined as $\{x \in \mathbb{Z}_N^* \mid \mathbb{J}(x, N) = 1\}$ where $N = pq$ and p, q being k-quasi-safe-primes,

where the latter subgroup is particularly important for the instantiation of our encryption scheme. Note that the remaining groups are appealing to instantiate LWEE in other applications.

When instantiating the group in finite fields or in elliptic curves, one chooses the Number Field Sieve (NFS) algorithm (for finite fields such as in (b)) or the Pohlig-Hellman [PH78] (and resp. Pollard-Rho [Pol78]) algorithm (in a generic group such as in (a)). For the groups (c) and (d) we use the results from [Mil75, Bac84]. Bach [Bac84] and Miller [Mil75] show that if there exists a PPT algorithm $\mathcal{A}$ solving the DL problem for a composite modulus on all inputs, then a PPT algorithm exists which solves the Factoring problem with arbitrarily high probability. Hence, we demand that factoring is hard for which the best algorithm is NFS, too. If we consider the computation of discrete logarithms in subgroups T (of order p) of a multiplicative group $\mathbb{G}$ (of order q), DL in T is hard if the NFS attack in $\mathbb{G}$ and the generic Pollard-Rho attack for groups of order $|T| = p$ is hard. Moreover, in the group $\mathbb{J}_N$ in (d) we have that $(p-1)$ and $(q-1)$ share common factors, namely 2^k, for which one can apply McKee and Pinch's algorithm [MP98], factoring $N = pq$ in essentially $O(N^{1/4}/2^k)$ operations. This is also observed in [Gir91, LL95, JL13]. Furthermore, the Coppersmith algorithm [Cop96, Cop97] (based on LLL) factors N efficiently if $k > \frac{1}{2}\min(\log_2 p, \log_2 q)$. For this reason we pick primes p, q of similar bit length and hinder both attack algorithms. NFS [CS06] has the running time $L_p[1/3, \sqrt[3]{64/9}]$ for modulus p where the complexity function $L_p[t, s]$ is defined by $L_p[t, s] = e^{s(1+o(1))(\ln p)^t(\ln\ln p)^{1-t}}$. The Pohlig-Hellman [PH78] and Pollard-Rho [Pol78] algorithms take time roughly $\sqrt{p}$ for computing individual discrete logarithms.

When estimating security parameters we take previously known attacks and timings into account by saying that if computing discrete logarithms in groups of order p takes time t, then

we expect that computing DLs in groups of order p' takes time roughly $t' \approx t\frac{L_{p'}[1/3, \sqrt[3]{64/9}]}{L_p[1/3, \sqrt[3]{64/9}]}$. If the difference between p' and p is not too large, the term $o(1)$ goes to zero. A similar strategy has been recommended in [LV00].

We take as reference the 2009 factorization of a 768-bit modulus, which offers roughly 66 security bits ($t \approx 2^{66}$). We stress that the parameters suggested in Table 3 should be handled with care. If one selects parameters for cryptographic constructions based on the hardness of DRP or LWEE, respectively, then the tightness of security reduction to the underlying problem takes an important role. Assume the security reduction says that if an adversary $\mathcal{A}$ breaks the security of the cryptographic scheme in time t with probability ϵ, then one can solve the $\mathsf{DRP}_{\mathbb{G},\ell}$ problem in time t' with probability ϵ'. In order that the scheme offers κ security bits, the parameters have to be chosen such that $(t'\epsilon)/(t\epsilon) \leq 2^{\kappa}$. Thus, one has to compensate a non-tight reduction by strengthening the underlying hardness assumption.

B.2 Hardness of the Learning with Errors Problem

Determining the hardness of lattice-based problems is a delicate issue. There are several reasons for this. First, lattice problem instances typically are defined over multiple parameters. Thus solvers rather depend on the particular configuration of the problem instance. Second, there are few theoretical results known about the behavior and running time of lattice algorithms.

In this work, we review Lindner and Peikert's "nearest-plane approach" [LP11] (revisited and improved by Liu and Nguyen [LN13]) which is considered these days as the status-quo.

Nearest-Plane Approach. Linder-Peikert's attack is a generalizes Babai's nearest plane algorithm [Bab86]. The attack consists of two steps: (a) a basis reduction to precompute a good basis of a lattice defined by the matrix $\mathbf{A}$, and (b) a probabilistic search algorithm with a success probability related to the quality of the basis. Lindner-Peikert's approach inherently allows a trade-off between the time spend on the basis reduction and the search algorithm. That trade-off is controlled by the Hermite factor δ, which measures the quality of the basis. We say that a basis $\mathcal{B} = \{b_0, \ldots, b_{m-1}\}$ of an m-dimensional lattice Λ has Hermite factor δ, if $\|b_0\| = \delta^m \det(\Lambda)^{1/m}$. For a given probability p and Hermite factor δ, one can compute the effort of the search algorithm needed to succeed at least with success p. Lindner and Peikert claim in [LP11] that it takes about 2^{-16} seconds to perform one "search-step" (for readers familiar with the nearest-plane algorithm: to search one parallelepiped spanned by the Gram-Schmidt orthogonalized basis). This allows us to estimate the running time of the search step, given p and δ. It is folklore that the running time of a basis reduction depends mainly on the desired Hermite factor of the reduced basis. The original paper considers the BKZ basis-reduction algorithm [SE94]. There have however been several improvements to BKZ. Most improvements are summarized in the remarkable work by Chen and Nguyen [CN11]. The BKZ 2.0 algorithm comes together with a simulation algorithm that can be used to predict its behavior. Albrecht et al. [AFG13] used the results of [LN13] to give an easy formula that roughly estimates the running time t necessary to compute a basis with

given Hermite factor. They conjecture that the time t can be approximated by

$$\log_2(t) = 0.009/\log_2^2 \delta_0 - 27.$$

Parameters. Since we are now able to estimate the total running time of the attack, given the desired success probability and Hermite factor, we can use a numerical method to obtain the best parameters and thereby the expected running time necessary to break the LWE instance. Given that the computers used for these experiments execute about 2^{10} operations per second, this can be used to estimate the bit security of LWE instances. Table 4 summarizes the results.

Security / Parameters	n	modulus	σ
80-bit	240	327680	33.98
128-bit	320	327680	32.01
256-bit	550	327680	28.55

Table 4: Example instantiation of LWE for different security levels

Exponential Gap Between Error and Modulus. For our double hardness instantiation, we have to estimate the security of LWE instances with an exponential gap between the error size and the modulus. The hardness of LWE with exponentially small gap between error and modulus is not well understood today. Brakerski and Vaikuntanathan [BV11b] say that if the error is a $1/2^{n^\varepsilon}$ fraction of the modulus N, the best known algorithm runs in time approximately $2^{n^{1-\varepsilon}}$. With the methodology, we can perform a binary search for the smallest dimension that suits our needs. Table 5 gives LWE instances that are suitable for double hardness instantiation of our scheme.

Security / Parameters	n	log(modulus)	log(σ)
80-bit	67000	927	97
128-bit	270000	2378	306
256-bit	2500000	11506	1741

Table 5: Example instantiation of LWE for different security levels

C Proofs

C.1 Proposition 2.6

Proof. Suppose there exists an adversary $\mathcal{A}$ which solves the $\mathsf{DRP}_{\mathbb{G},\chi,\ell+1,m}$ problem in time t with probability ϵ. We show that in this case, there exists an adversary $\mathcal{B}$ with black-box access to $\mathcal{A}$ which solves the $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ problem with probability ϵ.

Adversary $\mathcal{B}$ is given as challenge the tuple $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{h}}) \in \mathbb{G} \times \mathbb{G}^{m\times\ell} \times \mathbb{G}^{\ell} \times \mathbb{G}^{m}$. She invokes adversary $\mathcal{A}$ with input the group $\mathbb{G}$ and its generator g. Adversary $\mathcal{A}$ expects as challenge a tuple $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{h}}) \in \mathbb{G} \times \mathbb{G}^{m\times\ell+1} \times \mathbb{G}^{\ell+1} \times \mathbb{G}^{m}$. To this end, $\mathcal{B}$ samples $x_{\ell+1}$ according distribution χ, and $\mathbf{a} = (a_1, \ldots, a_m)$ uniformly from $\mathbb{Z}_q^m$. Adversary $\mathcal{B}$ provides $\mathcal{A}$ with the challenge $(g, g^{\mathbf{M}'}, g^{\mathbf{x}'}, g^{\mathbf{h}'})$ where $g^{\mathbf{M}'} = (g^{\mathbf{M}'}, g^{\mathbf{a}})$, $g^{\mathbf{x}'} = (g^{\mathbf{x}}, g^{x_{\ell+1}})$, and $g^{h'_i} = g^{h_i} \cdot g^{a_i x_{\ell+1}}$ for $i \in [m]$. Note that $g^{x_{\ell+1}}$ is distributed as expected as we choose $x_{\ell+1} \leftarrow_R \chi$. Moreover, $g^{\mathbf{a}}$ is uniformly distributed in $\mathbb{G}^m$. If the $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ tuples are such that $g^{h} = \prod_{i=1}^{\ell} g_i^{x_i}$, then $g^{h'}$ in the $\mathsf{DRP}_{\mathbb{G},\chi,\ell+1,m}$ distribution is computed correctly. This follows from the fact that for all $i \in [m]$ we have

$$g^{\mathbf{h}'} = g^{\mathbf{h}} \cdot (g^{\mathbf{a}})^{x_{\ell+1}} = g^{\mathbf{M}\mathbf{x}} \cdot (g^{\mathbf{a}})^{x_{\ell+1}} = g^{\mathbf{M}'\mathbf{x}'}$$

given $g^{\mathbf{h}} = g^{\mathbf{M}\mathbf{x}}$. In case $g^{\mathbf{h}}$ is a random group element, so is $g^{\mathbf{h}'}$, since $\mathbf{a}, x_{\ell+1}$ are sampled independently of h. Hence, $\mathcal{B}$ outputs in her game what $\mathcal{A}$ guesses, and wins with $\mathcal{A}$'s advantage ϵ. The running time of $\mathcal{B}$ is essentially the same as $\mathcal{A}$ merely adding the time to sample $O(m)$ uniform group elements. □

C.2 Proposition 2.8

Proof. We prove this theorem by contradiction. We assume that $\mathsf{RH}_{\mathbb{G},m,m+1,m+1,2\ell+1}$ is (t,ϵ)-hard and $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ is (t',ϵ')-hard. However, we assume that there is an algorithm $\mathcal{A}$ which solves $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m+1}$ in time t with probability $\epsilon'' > (1-\epsilon)^{-1}\epsilon'$.

We then build an algorithm $\mathcal{B}$ with black-box access to $\mathcal{A}$ which solves the $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ problem in time $t' \approx t$ with probability larger than ϵ' as follows. The algorithm $\mathcal{B}$ is given a DRP instance $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{h}})$ for uniform matrix $\mathbf{M} \leftarrow_R \mathbb{Z}_q^{m\times\ell}$ and has to decide whether $g^{\mathbf{h}}$ equals $g^{\mathbf{M}\mathbf{x}}$ or was chosen uniformly from $\mathbb{G}^m$. Algorithm $\mathcal{B}$ now prepares a DRP instance for $\mathcal{A}$ by adding a row to the matrix $g^{\mathbf{M}}$ and vector $g^{\mathbf{h}}$ as follows. It chooses a random index $0 \le i \le m$ and samples a random coefficient vector $\mathbf{y} \in \mathbb{Z}_q^m$. Let $\mathbf{u} = g^{\mathbf{y}^\top\mathbf{M}} = g^{y_1\mathbf{m}_1} \cdot \ldots \cdot g^{y_m\mathbf{m}_m}$ and $v = g^{\langle \mathbf{h}_i, \mathbf{y}\rangle}$. Create the matrix $g^{\mathbf{M}'} \in \mathbb{G}^{(m+1)\times\ell}$ by inserting $\mathbf{u}$ before the ith column of $g^{\mathbf{M}}$, and $\mathbf{h}' \in \mathbb{G}^{m+1}$ by inserting $\mathbf{v}$ before the ith entry of $\mathbf{h}$. Now, $\mathcal{B}$ invokes $\mathcal{A}$ upon input $(g, g^{\mathbf{M}'}, g^{\mathbf{x}}, g^{\mathbf{h}'})$.

At this point, we stress that $\mathcal{A}$ will accept the input and work properly even if $(g, g^{\mathbf{M}'}, g^{\mathbf{x}}, g^{\mathbf{h}'})$ is of different rank. In fact, an honestly generated DRP instance for the $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m+1}$ problem will have a rank $\min(\ell, m+1)$ matrix (with overwhelming probability), while our input matrix has rank $\min(\ell, m)$ (with overwhelming probability). Since by assumption there is no algorithm that can distinguish those two inputs (matrices) in time t with a probability greater than ϵ, algorithm $\mathcal{A}$, which also runs in time t, must work for the given input with probability greater than $(1-\epsilon)$. Algorithm $\mathcal{A}$ returns a guess $b \in \{0,1\}$ for its challenge which in turn constitutes the guess of $\mathcal{B}$ for its challenge instance $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{h}})$. Since $\mathcal{A}$ successfully wins its challenge in time t with probability ϵ'', we have constructed an algorithm $\mathcal{B}$ which breaks $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ in time $t' \approx t$ with probability $(1-\epsilon)\epsilon'' > \epsilon'$. This leads to a contradiction to $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m}$ being (t',ϵ')-hard. Hence, $\mathsf{DRP}_{\mathbb{G},\chi,\ell,m+1}$ must be $(t', (1-\epsilon)^{-1}\epsilon')$-hard.

□

C.3 Proposition 3.2

Proof. Suppose there exists an adversary $\mathcal{A}$ which solves the $\mathsf{SLWEE}_{\mathbb{G},\ell,m,q,\chi_e}(\chi_s)$ problem in time t with probability ϵ. We show that in this case, there exists an adversary $\mathcal{B}$ with black-box access to $\mathcal{A}$ which solves the $\mathsf{SRP}_{\mathbb{G},\chi_s,\ell,m}$ problem in time $\approx t$ with probability ϵ.

Adversary $\mathcal{B}$ is given as challenge a SRP instance $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{Mx}})$ and is asked for a vector $\mathbf{x} \in \mathbb{Z}_q^\ell$ such that $g^{\mathbf{Mx}'} = g^{\mathbf{Mx}}$. Adversary $\mathcal{B}$ invokes $\mathcal{A}$ with input the group $\mathbb{G}$ and its generator g. Whenever $\mathcal{A}$ asks for the i-th sample from the $L^{\mathsf{LWEE}}_{\mathbb{G},\ell,m,\chi_e}$ distribution, adversary $\mathcal{B}$ returns the i-th row of $g^{\mathbf{M}}$ and the i-th element of $g^{\mathbf{Mx}}$ with some noise $e_i \leftarrow \chi_e$, i.e., $g^{\mathbf{a}_i} := g^{\mathbf{M}[i]}$ and $g^{b_i} := g^{(\mathbf{Mx})_i} \cdot g^{e_i}$. Note that $(g^{\mathbf{a}_i}, g^{b_i})$ as such corresponds to the $L^{\mathsf{LWEE}}_{\mathbb{G},\ell,m,\chi_e}$ distribution. The vector $g^{\mathbf{a}_i}$ is uniformly distributed as the input $g^{\mathbf{M}}$ for SRP is uniformly distributed. Moreover, we have

$$g^{b_i} = g^{(\mathbf{Mx})_i} \cdot g^{e_i} = g^{\langle \mathbf{M}[i], \mathbf{x} \rangle + e_i} \ .$$

Note that $\mathcal{B}$ can provide $\mathcal{A}$ enough samples since both algorithms get m samples from their respective distributions.

Eventually, adversary $\mathcal{A}$ will output an element $\mathbf{s} \in \mathbb{Z}_q^\ell$ such that for all $i \in \{1, \ldots, m\}$ it holds $g^{\langle \mathbf{a}_i, \mathbf{s} \rangle + e'_i} = g^{b_i}$ where $e'_i \leftarrow_R \chi_e$. Now, since there can exist only a single vector $\mathbf{s}$ which can fulfill the equation $g^{\langle \mathbf{a}_i, \mathbf{s} \rangle + e'_i} = g^{b_i}$ for errors $e' \leftarrow_R \chi_e$, we must have $\mathbf{s} = \mathbf{x} = (x_1, \ldots, x_\ell)$. Hence, $\mathcal{B}$ outputs $\mathbf{s}$ as the solution vector for her instance.

The running time of $\mathcal{B}$ is almost identical to $\mathcal{A}$, and the success probability is equal, too. The proposition follows accordingly. □

C.4 Proposition 3.3

Proof. Suppose there exists an adversary $\mathcal{A}$ which solves the $\mathsf{SLWEE}_{\mathbb{G},n,m,\chi_e}(\chi_s)$ problem in time t with probability ϵ. We show that in this case, there exists an adversary $\mathcal{B}$ with black-box access to $\mathcal{A}$ which solves the $\mathsf{SLWE}_{n,m,q,\chi_e}(\chi_s)$ problem in time $\approx t$ with probability ϵ.

Adversary $\mathcal{B}$ is allowed to ask for samples $(\mathbf{a}_i, b_i)$ which are distributed either according to the $L^{\mathsf{LWE}}_{n,m,q,\chi_e}$ distribution or distributed uniformly in $(\mathbb{G}^n \times \mathbb{G})$. Adversary $\mathcal{B}$ invokes adversary $\mathcal{A}$ with input $\mathbb{G}$ (the group of order q) and samples a random generator g for that group. When $\mathcal{A}$ asks for i-th sample $(g^{\mathbf{a}_i}, g^{b_i})$, $\mathcal{B}$ asks for samples $(\mathbf{a}_i, b_i)$ in his own game and returns to $\mathcal{A}$ the tuple $(g^{\mathbf{a}_i}, g^{b_i})$.

Eventually, $\mathcal{A}$ outputs the secret $\mathbf{s}$, which $\mathcal{B}$ forwards to his own game as output. Time complexity of $\mathcal{B}$ is the time required by $\mathcal{A}$ plus taking exponentiations, which is a negligible cost. □

C.5 Proposition 3.4

Proof. Suppose there exists an adversary $\mathcal{A}$ which solves the $\mathsf{DLWEE}_{\mathbb{G},\ell,m,\chi_e}(\chi_s)$ problem in time t with probability ϵ. We show that in this case, there exists an adversary $\mathcal{B}$ with black-box access to $\mathcal{A}$ which solves the $\mathsf{DRP}_{\mathbb{G},\chi_s,\ell,m}$ problem in time $\approx t$ with probability ϵ.

Adversary $\mathcal{B}$ is given as challenge a DRP instance $(g, g^{\mathbf{M}}, g^{\mathbf{x}}, g^{\mathbf{h}})$ and has to decide whether $\mathbf{h}$ equals $\mathbf{Mx}$ or was chosen uniformly at random from $\mathbb{Z}_q^m$. Adversary $\mathcal{B}$ invokes $\mathcal{A}$ with input the group $\mathbb{G}$. Whenever $\mathcal{A}$ asks for the i-th sample from the $L^{\mathsf{LWEE}}_{\mathbb{G},\ell,m,\chi_e}$ distribution, adversary $\mathcal{B}$ returns the i-th row of $g^{\mathbf{M}}$ and the i-th element of $g^{\mathbf{h}}$ with some noise $e_i \leftarrow \chi_e$, i.e., $g^{\mathbf{a}_i} := g^{\mathbf{M}[i]}$ and $g^{b_i} := g^{h_i} \cdot g^{e_i}$. Note that $(g^{\mathbf{a}_i}, g^{b_i})$ as such corresponds to the $L^{\mathsf{LWEE}}_{\mathbb{G},\ell,m,\chi_e}$ distribution. The vector $g^{\mathbf{a}_i}$ is uniformly distributed as the input $g^{\mathbf{M}}$ for DRP is uniformly distributed. Moreover, we have

$$g^{b_i} = g^{h_i} \cdot g^{e_i} = g^{(Mx)_i} \cdot g^{e_i} = g^{\langle \mathbf{M}[i], \mathbf{x} \rangle + e_i}$$

if $g^{\mathbf{h}} = g^{\mathbf{Mx}}$. Otherwise, $g^{\mathbf{b}_i}$ is distributed uniformly in $\mathbb{G}$ since $\mathbf{h}$ is. Note that $\mathcal{B}$ can provide $\mathcal{A}$ enough samples since both algorithms get m samples from their respective distributions.

Hence, when adversary $\mathcal{A}$ outputs a bit d, adversary $\mathcal{B}$ outputs d in her decisional representation problem. If $\mathcal{A}$ guessed correctly, so does $\mathcal{B}$. The running time of $\mathcal{B}$ is almost identical to $\mathcal{A}$, and the success probability is equal, too. The proposition follows accordingly. □

C.6 Proposition 3.5

Proof. Suppose there exists an adversary $\mathcal{A}$ which solves the $\mathsf{DLWEE}_{\mathbb{G},n,m,\chi_e}(\chi_s)$ problem in time t with probability ϵ. We show that in this case, there exists an adversary $\mathcal{B}$ with black-box access to $\mathcal{A}$ which solves the $\mathsf{LWE}_{n,m,q,\chi_e}(\chi_s)$ problem in time $\approx t$ with probability ϵ.

Adversary $\mathcal{B}$ is allowed to ask for samples $(\mathbf{a}_i, b_i)$ which are distributed either according to the $L^{\mathsf{LWE}}_{n,m,q,\chi_e}$ distribution or distributed uniformly in $(\mathbb{G}^n \times \mathbb{G})$. Adversary $\mathcal{B}$ invokes adversary $\mathcal{A}$ with input $\mathbb{G}$ (the group of order q) and samples a random generator g for that group. When $\mathcal{A}$ asks for i-th sample $(g^{\mathbf{a}_i}, g^{b_i})$, $\mathcal{B}$ asks for samples $(\mathbf{a}_i, b_i)$ in his own game and returns to $\mathcal{A}$ the tuple $(g^{\mathbf{a}_i}, g^{b_i})$.

Eventually, $\mathcal{A}$ outputs a bit b, which $\mathcal{B}$ forwards to his own game as output. It is easy to verify that the samples $(\mathbf{a}_i, b_i)$ are distributed according to $L^{\mathsf{LWE}}_{n,m,q,\chi_e}$ if and only if the samples $(g^{\mathbf{a}_i}, g^{b_i})$ are distributed according to $L^{\mathsf{LWEE}}_{\mathbb{G},n,m,\chi_e}$. Hence, $\mathcal{B}$ wins whenever $\mathcal{A}$ does while having approximately the same running time $\approx t$. □