Augmented Learning with Errors: The Untapped Potential of the Error Term

Rachid El Bansarkhani, Özgür Dagdelen, and Johannes Buchmann

Technische Universität Darmstadt
Fachbereich Informatik
Kryptographie und Computeralgebra,
Hochschulstraße 10, 64289 Darmstadt, Germany
elbansarkhani@cdc.informatik.tu-darmstadt.de, oezguer.dagdelen@cased.de,
buchmann@cdc.informatik.tu-darmstadt.de

Abstract. The Learning with Errors (LWE) problem has gained a lot of attention in recent years leading to a series of new cryptographic applications. Specifically, it states that it is hard to distinguish random linear equations disguised by some small error from truly random ones. Interestingly, cryptographic primitives based on LWE often do not exploit the full potential of the error term beside of its importance for security. To this end, we introduce a novel LWE-close assumption, namely Augmented Learning with Errors (A-LWE), which allows to hide auxiliary data injected into the error term by a technique that we call message embedding. In particular, it enables existing cryptosystems to strongly increase the message throughput per ciphertext. We show that A-LWE is for certain instantiations at least as hard as the LWE problem. This inherently leads to new cryptographic constructions providing high data load encryption and customized security properties as required, for instance, in economic environments such as stock markets resp. for financial transactions. The security of those constructions basically stems from the hardness to solve the A-LWE problem.
As an application we introduce (among others) the first lattice-based replayable chosen-ciphertext secure encryption scheme from A-LWE.
Keywords: Lattice-Based Cryptography, Encryption Scheme, Lattice-Based Assumptions

1 Introduction

Lattice-based cryptography constitutes arguably one of the most promising alternatives to classical cryptography. This observation is supported by various arguments such as the conjectured resistance against quantum attacks. Moreover, lattice-based cryptography is equipped with a rich combinatorial structure providing provable-security guarantees [Ajt96, Reg04, MR04], while carrying out low complexity operations and thus allowing for efficient constructions. The security of such cryptosystems is mainly based on the hardness of either solving the Small Integer Solution (SIS) problem or the Learning With Errors (LWE) problems. The former is widely employed for building provably secure primitives from Minicrypt, such as collision-resistant hash functions [LMPR08, ADL$^+$08] and signature schemes [GPV08, MP12, DDLL13, GLP12, Lyu12], while the latter mainly serves as a hard underlying problem for the security of primitives from Cryptomania, such as key exchange [KV09, JD12, Pei14] and oblivious transfer [PVW08]. Remarkably, both problems are strongly related as SIS is considered to be the dual problem of LWE.

The LWE problem exists essentially in two variants, the decision and search version. Following this, the challenger is given $poly(n)$ number of independent samples $(\mathbf{A}_i, \mathbf{b}_i^\top) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, where $\mathbf{A}_i \leftarrow_R \mathbb{Z}_q^{n\times m}$, $\mathbf{e}_i \leftarrow_R \chi$ and $\mathbf{b}_i^\top = \mathbf{s}^\top \mathbf{A}_i + \mathbf{e}_i^\top \mod q$ for $\mathbf{s} \in \mathbb{Z}_q^n$ where χ is some arbitrary distribution over $\mathbb{Z}_q^m$, typically discrete Gaussian. He is then asked to distinguish those samples from uniformly random samples from $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$. In search-LWE, however, the challenger is required to find the secret $\mathbf{s}$. Besides its presumably quantum hardness, one of the most noteworthy properties lattice-based assumptions offer is worst-case hardness of average-case instances. Starting with the works of Ajtai [Ajt96] and Micciancio and Regev [MR04], the hardness of some average-case instances of the SIS problem was shown to be hard as long as worst-case instances of the (decision version of the) shortest vector problem, known as

GapSVP, are hard. The worst-case hardness for LWE was first stated by Regev [Reg05]. Regev showed that if the error vector follows the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,\alpha q}$ with parameter $\alpha q \geq 2\sqrt{n}$, solving search-LWE is at least as hard as quantumly solving $\widetilde{O}(n/\alpha)$-SIVP and GapSVP in n-dimensional worst-case lattices. Later, Peikert [Pei09] and Brakerski et al. [BLP+13] gave a classical reduction from GapSVP to LWE. In [DMQ13] Döttling and Müller-Quade proved the hardness of LWE for uniformly distributed errors. Subsequently, Micciancio and Peikert [MP13] show that LWE remains hard even for binary errors.

In the realm of encryption, the first provably secure lattice-based encryption scheme was due to Ajtai and Dwork [AD97]. The security of this scheme relies on the worst-case hardness of approximating SVP within polynomial factors. Several other works followed with focus on improving the efficiency [GGH97,Reg04]. Ever since the breakthrough work of Regev [Reg05], the learning with errors assumption and its ring variant are widely used in lattice-based cryptography to base the security of cryptographic schemes upon LWE. Indeed, since then lattice-based cryptography emerged and novel encryption schemes have been built upon this assumption, such as fully homomorphic encryption [Gen09, BV11a, GH11, Bra12, BGV12] and identity-based encryption [GPV08, CHKP10, ABB10a, ABB10b] besides of CPA-secure [Reg05, PVW08, LPR10, LP11, LPR13] and CCA-secure encryption schemes [PW08, Pei09, MP12, Pei14]. Many of those encryption schemes utilize LWE in order to blind certain sensitive data.

Cryptographic constructions which rely on the LWE assumption usually sample an error term according to some distribution, most often Gaussian. Such a choice has many advantages over other distributions. However, many of the existing LWE-based schemes do not exploit the full potential of the error term. This observation is mainly due to three reasons, which can be summarized using the example of encryption schemes.

- First, previous LWE-based encryption schemes produce ciphertexts mainly following the idea of one-time pad encryption, where LWE samples play the role of random vectors. As a consequence, the underlying constructions heavily rely on the error term to be short in order to correctly recover the message. A major drawback of such schemes is the waste of bandwidth, i.e., all bits created for the error term are sacrificed for a few message bits.
- Second, there exist no proposals using the error term or other involved random variables as additional containers carrying auxiliary data, besides of its task to provide the required distributions. Once recognizing its feasibility, it fundamentally changes the way of building cryptosystems. For instance, in encryption schemes one may inject the message into the error term without necessarily changing the target distributions.
- Third, there is a lack of efficient trapdoor functions that recover the secret *and* the error term from an LWE instance, which is obviously a necessary condition for exploiting the error term. Only a few works such as [SSTX09, MP12] provide mechanisms to recover the error term.[1] The most promising trapdoor construction is proposed by Micciancio and Peikert [MP12].

We make the following conclusions. The above limitations of LWE intuitively ask for an alternative LWE definition that takes account for the modifications made to the error term, while ensuring essentially the same hardness results as the traditional LWE problem. Since such an assumption already encompasses message data within the error term, one obtains, as a consequence, a generic and practically new encryption scheme secure under the new variant of the LWE assumption, where the trapdoor function is viewed as a black box recovering the secret and the error vector from a modified LWE instance. The message is subsequently extracted from the error vector. This allows one to exploit the full bandwidth of the error vector with full access to all its entries and not just its length. Remarkably, one could even combine this approach with existing methods for encryption in order to further increase the message throughput per ciphertext.

[1] We show in Appendix 4.5 that the symmetric-key somewhat-homomorphic encryption scheme from [BV11b] can be modified such that one also obtains the error term (in addition to the plaintext) when decrypting.

In this work we address this challenge and give a detailed description of how to exploit the error vector.

Our Contribution. Based on these observations and subsequently made conclusions, we start by giving an alternative LWE definition, called Augmented LWE (A-LWE), that extends the existing one by modifying the error term in such a way that it encapsulates additional information. We further show which instantiations yield A-LWE samples that are indistinguishable from traditional LWE samples, thereby enjoying the hardness of traditional LWE. In conjunction with the high quality trapdoor candidate from [MP12], we have full access to the error term. This result inherently yields new cryptographic applications, which ensure security in various models while simultaneously allowing for high data load encryption that are applicable, for instance, in financial environments such as stock markets operating with huge amounts of stock information. It is even possible to encrypt lattice-based signatures much more efficiently than ordinary messages, which is an interesting technique for internet protocols, where the acknowledgement of ip-packets represents an important measure for reliability. In this case, the whole entropy of the error term is supplied by lattice-based signatures.

For instance, this strategy allows us to derive a generic encryption scheme, where ciphertexts are represented by plain A-LWE samples. Besides of its evident security properties, that can directly be deduced from A-LWE, our construction benefits from encrypting more message bits per ciphertext and a faster decryption engine through a conceptually easier instantiation as compared to other proposals. Furthermore, we give a detailed description of how to achieve publicly-detectable replayable CCA (pd-RCCA) security [CKN03], a slightly relaxed version of CCA2, but strictly stronger than CCA1. In fact, we propose the first lattice-based RCCA-secure encryption scheme. Due to the versatility of the error term, this functionality does not involve ciphertext expansion. As a third application, it is possible to replace parts of the error term by signatures that are generated according to the best known and widely used lattice-based signature schemes. Specifically, we focus on the GPV signature scheme [GPV08] in combination with the trapdoor construction [MP12] and the practical signature schemes presented in [DDLL13, Lyu12], and thus realize an asymmetric authenticated encryption scheme. As a nice byproduct, one can immediately transfer the proposed concepts to the CCA-secure construction provided in [MP12]. This allows us to increase the message throughput per ciphertext, while enjoying RCCA-security at almost no costs. Noteworthy, all the proposed concepts are also applicable to specific constructions such as the somewhat homomorphic symmetric key encryption scheme due to [BV11b], which does not rely on the trapdoor construction from [MP12].

1.1 Augmented Learning with Errors

In many lattice-based cryptographic schemes, one has to sample error terms following the discrete Gaussian distribution as a requirement for the scheme to be secure. This is often due to an LWE-based security reduction. The key concept underlying our proposal is to embed further information in the error term $\mathbf{e} \in \mathbb{Z}^m$, but in such a way that the distribution of the augmented error term is computationally close to the discrete Gaussian distribution over $\mathbb{Z}^m$. We also show that one can embed messages in uniformly distributed error vectors using the same methodology.

The idea of our technique is the following. We employ the gadget matrix $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^\top$, firstly introduced in [MP12], with $\mathbf{g}^\top = (1, 2, \ldots, 2^{k-1})$ and modulus $q = 2^k$ in order to sample vectors according to the discrete Gaussian distribution $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),r}$. Vectors $\mathbf{e} \in \mathbb{Z}_q^m$ distributed according to $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),r}$ satisfy the equation $\mathbf{Ge} \equiv \mathbf{v} \bmod q$ for arbitrary $\mathbf{v} \in \mathbb{Z}_q^{m/k}$. Let $H : \{0,1\}^* \to \{0,1\}^m$ be some function and $(\mathsf{encode}, \mathsf{decode})$ a pair of algorithms which allow one to switch between the representations $\mathbb{Z}_q^{m/k}$ and $\{0,1\}^m$. We compute a random coset $\mathbf{v} = \mathsf{encode}(H(\mathsf{seed}) \oplus \mathbf{m}) \in \mathbb{Z}^{m/k}$, where $\mathbf{m} \in \{0,1\}^m$ denotes an arbitrary message of bit length m. We show that if H is instantiated by a cryptographic hash function modeled as a random oracle, $\mathbf{v}$ is indeed indistinguishable from uniform. We only have to take care that

the input to the function H, namely the seed, has sufficient (computational) min-entropy. Whoever has access to this seed can deterministically recover the message by $\mathbf{m} = \mathsf{decode}(\mathbf{Ge} \mod q) \oplus H(\mathsf{seed})$. This result immediately impacts all schemes that allow for error term recovery, as it enhances the compactness of the scheme.

Embedding auxiliary private information into the error term raises certain new computational problems. In addition to the secret and error vector of an LWE instance, also the new embedded message is concealed. In fact, we claim that LWE samples modified as above are indistinguishable from uniform even for adversarially chosen messages. To this end, we introduce a novel problem, namely the *Augmented* LWE (A-LWE) problem, which differs from the traditional LWE problem only in the way the error term is produced. More specifically, we split the error term $\mathbf{e} \in \mathbb{Z}_q^m$ of LWE into $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$, where $\mathbf{e}_1 \in \mathbb{Z}_q^{m_1}$ and $\mathbf{e}_2 \in \mathbb{Z}_q^{m_2}$. An A-LWE sample is then distributed as follows. For a given $\mathbf{s} \in \mathbb{Z}_q^n$, first choose $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n \times m}$ uniformly at random. Then, sample $\mathbf{e}_1 \leftarrow_R \mathcal{D}_{\mathbb{Z}^{m_1}, \alpha q}$ and $\mathbf{e}_2 \leftarrow_R \mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}), \alpha q}$, where $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}, \mathbf{e}_1) \oplus \mathbf{m})$ for some function H. The tuple $(\mathbf{A}, \mathbf{b}^t = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top)$ represents an A-LWE sample. We show that distinguishing A-LWE samples from traditional LWE samples is hard for properly chosen random function H. More formally, if H is a cryptographic hash function modeled as a random oracle, the tuple $(\mathbf{s}, \mathbf{e}_1)$ has sufficient entropy in each sample and the LWE problem for parameters m, n, α, q is hard to solve, then we obtain a negligible computational distance between LWE and A-LWE distributions. Thus, we immediately deduce the hardness of A-LWE from LWE. As an immediate consequence, the confidentiality of the message is protected as long as $\mathsf{decision\ A\text{-}LWE}$ and hence $\mathsf{decision\ LWE}$ is hard.

Based on the A-LWE hardness, we present a novel and generic encryption scheme, where ciphertexts are embodied by plain A-LWE samples. One merely employs an arbitrary suitable trapdoor construction for the function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ that allows for error term recovery. Hence, the efficiency of encryption and decryption greatly depends on the quality of the trapdoor and the inversion algorithm. The currently most efficient candidate function is known from Micciancio and Peikert [MP12]. Note that while some encryption schemes like [SSTX09, MP12] utilize such a trapdoor function, the error term is left unpacked. To the best of our knowledge, we provide the first lattice-based encryption schemes exploiting the error term as an (additional) data container in addition to its necessity for security.

1.2 Applications

CCA-Secure Encryption. Based on the A-LWE hardness, we build a conceptually new and very simple CCA1 secure encryption scheme. In previous lattice-based encryption schemes such as [PW08, ABB10a, MP12, LP11], ciphertexts are computed in a one-time pad manner by adding the message to a random vector coming from the LWE distribution. Thus, an adversary succeeds in the respective security game, if she is able to distinguish LWE samples from random ones with non-negligible advantage. Our scheme, however, moves apart from this approach and focuses on the error term recovery of A-LWE samples and subsequently decoding the error term. By this means, the ciphertext represents an A-LWE instance in its purest form. This implies a direct security reduction of the scheme to A-LWE. Employing the framework proposed in [MP12], we construct a random public key $\mathbf{A}$ that is endowed with a trapdoor. In conjunction with the corresponding inversion algorithm, we can efficiently recover the secret and the error term from the ciphertext $\mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ with $\mathbf{e} \leftarrow_R \mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}), \alpha q}$ for $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m})$.[2] Due to $\alpha q \geq 2\sqrt{n} \geq \eta_\epsilon(\Lambda_q^{\perp}(\mathbf{G}))$, we even do not impose any further restrictions to the parameters. Such a construction is almost optimal, since we do not initiate any further transformations.

The bit size of the message is equal to the dimension of the ciphertext m resulting in a small message expansion factor, which is lower than most of the existing schemes. In fact, due to this relationship there is an incentive to increase the parameter m in order to efficiently encrypt large amounts of data involving

[2] We show that if matrix $\mathbf{A}$ is fixed and for each sample vector $\mathbf{s}$ is uniformly sampled from $\mathbb{Z}_q^n$, the entropy of $\mathbf{s}$ is sufficient and one can sample the entire error term from $\mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}), \alpha q}$.

$m = c \cdot nk$, $k = \log q$	CCA1 **[MP12]**	CCA1 Constr. **4.1**	CCA1 Constr. **4.6**	CPA **[LP11]**
Ciphertext size	$m \cdot k$	$m \cdot k$	$m \cdot k$	$m \cdot k$
Signature size	nk	$c \log(\alpha q) nk$	$(c \log(\alpha q) + 1) nk$	$cnk - n$
Message size	nk	$c \cdot nk$	$(c+1) \cdot nk$	$cnk - n$
Message Expansion	$c \cdot k$	k	$k - \frac{k}{(c+1)}$	$k + \frac{k}{ck-1}$
Error rate α	$\tilde{O}(1/n)$	$\tilde{O}(1/n)$	$\tilde{O}(1/n)$	$\tilde{O}(1/n)$
public key size	$n \cdot m$	$n \cdot m$	$n \cdot m$	$n \cdot (m - 2n)$

Table 1. Comparison of key figures among CCA1-secure encryption schemes

less computations per ciphertext as compared to lower dimensions. We considered this case and can even show that decryption is essentially as fast as in lower dimensions. In particular, we provide an enhanced encryption scheme for high data load, where parts of the ciphertext and thus the error term are ignored when inverting the underlying A-LWE instance. That is, one extends any public key $\mathbf{A}_u$

$\mathbf{v} = H(u, \mathbf{s}, \mathbf{e}_2)$ using a random oracle H. Subsequently, we sample a preimage $\mathbf{e}_1 \leftarrow \mathcal{D}_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G})}$ that serves as the upper-part of the error term. Due to the injectivity of the trapdoor function, altering the ciphertext leads to different values for the corresponding variables such that the decryption routine outputs a failure. But modifications caused to the upper part of the error term do not result in a failure as long as short vectors from $\Lambda^{\perp}_q(\mathbf{G})$ are added.

This obviously implies a publicly-detectable RCCA-secure encryption scheme (pd-RCCA), an even stronger security guarantee than plain RCCA. In fact, we have the relation CCA2 $\Rightarrow$ pd-RCCA $\Rightarrow$ secretly-detectable RCCA $\Rightarrow$ RCCA [CKN03]. Security in the pd-RCCA model implies that a public party can check whether a modified ciphertext decrypts to the same message.

When it comes to CCA2 security, there exist many generic constructions [DDN00, CHJ$^+$02, HLM03, BCHK07] that ensure CCA2-security. For instance, one can use strongly unforgeable one-time signature schemes [DDN00], commitment schemes or message authentication codes (MAC) in order to transform a CPA-secure scheme into a CCA2-secure one. However, these generic constructions typically involve high complexity and overhead resulting in a less efficient encryption scheme. Our approach works differently as it uses the error term in order to provide this feature. Once having RCCA-security one can efficiently convert the scheme into a CCA2-secure encryption scheme using generic solutions as provided in [CKN03] or our individual approach at the expense of some small overhead.

Signature Embedding. There exist various approaches to provide message authentication of encrypted data. Many of them are generic and thus coupled to overhead and loss of efficiency. For instance, one can use MACs or digital signatures that are appended to the ciphertext. In our work we aim at providing this feature without suffering from the drawbacks of generic solutions through a thorough analysis of our encryption scheme.

Our goal is to replace parts of the error vector such as $\mathbf{e}_1$ completely by a lattice-based signature rather than appending it to the ciphertext or including it as a part of the message. This allows us to optimally exploit the full bandwidth of $\mathbf{e}_1$ due to some nice properties lattice-based signature schemes offer. One of the features is to let signatures be distributed following the discrete Gaussian distribution. For the underlying signature scheme itself, such a strategy has many advantages over other choices as it allows to decouple the distribution of the signature from the secret key, while sampling short signatures with higher probability. There exist many lattice-based proposals that have similar properties and perform very well in practice [DDLL13, Lyu12, MP12].

Our construction inherently relies on the capability to recover the error term from an A-LWE instance. As a result, we provide an authentication mechanism for encrypted data, since it is by construction possible to retrieve back an arbitrary discrete Gaussian vector with support $\mathbb{Z}^m$, hence also a signature, that was plugged into the error term. Therefore, we can embed signatures of size $m \cdot \log(\alpha q) = O(m \log n)$ bits into the error vector, which is far more (see Table 1) than with the standard encryption schemes that are restricted to the message size. Here, we denote by αq the parameter of the discrete Gaussian vector of the error term. In fact, our proposal allows for a flexible selection of parameters, because we do not impose any new constraints. However, the parameters of the signature scheme should not be too large in order to correctly invert the underlying A-LWE instances.

Remarkably, when using the encryption scheme for high data load with an extended public key $\mathbf{A}^{ext}_u$ the upper part of the error term is ignored when decrypting the ciphertext. This allows us to select the parameters in such a way that A-LWE (and LWE) is hard for arbitrarily chosen parameters of the signature scheme. Therefore, one can employ the upper-part of the error term for signatures. The resulting scheme has a CCA2-like behavior, where changes induced to the ciphertext are detected by the receiver. These ideas immediately help to improve the construction provided in [MP12]. In particular, we can apply the proposed techniques to the error term without changing the other ingredients. More specifically, we still build the ciphertext in a one-time pad manner, while simultaneously endowing the error vector with additional messages. The proof of security will subsequently be based on A-LWE rather than plain LWE.

Embedding Auxiliary Data in Homomorphic Encryption. As already noticed, we improve the CCA1-secure encryption scheme from [MP12], if we apply the proposed concepts from above to the error term. As a result, we have the first message being encrypted following the one-time pad approach and a second message injected into the error-term. However, this encryption scheme heavily relies on a trapdoor construction. But we stress that it is also possible to improve other more specific constructions that do not require trapdoors as such. For instance, if we consider the somewhat homomorphic encryption scheme due to Brakerski and Vaikuntanathan [BV11b], we can apply essentially the same ideas without any major modifications. Indeed, it is a symmetric key encryption scheme, where a ciphertext $(\mathbf{c}_1 = \mathbf{a}, \mathbf{c}_2 = \mathbf{b} + \mathbf{m})$ is derived by adding a ring-LWE samples $\mathbf{b} = \mathbf{as} + t\mathbf{e} \in \mathcal{R}_q = \mathbb{Z}[X]/\langle f(X)\rangle$ to an arbitrary message $\mathbf{m} \in \mathcal{R}_t$ for t coprime to q and freshly sampled $\mathbf{c}_1 = \mathbf{a} \in \mathcal{R}_q$ and error vector $\mathbf{e} \in \mathcal{R}_q$. The secret key is given by the secret ring element $\mathbf{s} \in \mathcal{R}_q$. After decrypting the ciphertext, we get full access to the error-term via $\mathbf{e} = t^{-1}(\mathbf{c}_2 - \mathbf{c}_1\mathbf{s} - \mathbf{m})$. A quick view to this construction reveals, that the error term can be recovered very efficiently. Clearly, this positively impacts the performance of the different concepts, when applied to the error term. More details on this construction can be found in Appendix 4.5.

2 Preliminaries

We recall the CCA1, CCA2, and RCCA security model in Appendix A.

2.1 Notation

By $\oplus$ we denote the XOR operator. We let $[\ell]$ denote the set $\{1, \ldots, \ell\}$ for any $\ell \in \mathbb{N}_{\geq 1}$. We indicate vectors by lower-case bold letters (e.g., $\mathbf{x}$) and use upper-case bold letters for matrices (e.g., $\mathbf{A}$). The set of integers modulo q are denoted by $\mathbb{Z}_q$ and reals by $\mathbb{R}$. Throughout this paper we will mainly consider the case $q = 2^k,\ k > \mathbb{N}$. If $\mathcal{X}$ is a set, we write $x \leftarrow_R \mathcal{X}$ to denote that x is sampled uniformly from $\mathcal{X}$. If $\mathcal{X}$ is a distribution, $x \leftarrow_R \mathcal{X}$ means that x was sampled according to $\mathcal{X}$.

Let $\mathcal{X} = \{\mathcal{X}_n\}_{n\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_n\}_{n\in\mathbb{N}}$ be two distribution ensembles. We say $\mathcal{X}$ and $\mathcal{Y}$ are (computationally) indistinguishable, if for every polynomial time distinguisher $\mathcal{A}$ we have $|\Pr[\mathcal{A}(\mathcal{X}) = 1] - \Pr[\mathcal{A}(\mathcal{Y}) = 1]| = \mathsf{negl}(n)$, and we write $\mathcal{X} \approx_c \mathcal{Y}$ (resp. $\mathcal{X} \approx_s \mathcal{Y}$ if we allow $\mathcal{A}$ to be unbounded). The statistical distance of two distributions $\mathcal{X}_1$ and $\mathcal{X}_2$ denoted by $\Delta(\mathcal{X}_1, \mathcal{X}_2)$ over a countable set $\mathcal{S}$ is defined by $\Delta(\mathcal{X}_1, \mathcal{X}_2) := \frac{1}{2}\sum_{s\in\mathcal{S}} |\mathcal{X}_1(s) - \mathcal{X}_2(s)|$.

We define by $\rho : \mathbb{R}^n \to (0, 1]$ the n-dimensional Gaussian function $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\cdot\frac{\|\mathbf{x}-\mathbf{c}\|_2^2}{s^2}},\ \forall \mathbf{x}, \mathbf{c} \in \mathbb{R}^n$. The discrete Gaussian distribution $D_{\Lambda+c,s}$ is defined to have support $\Lambda + c$, where $c \in \mathbb{R}$ and $\Lambda \subset \mathbb{R}^n$ is a lattice. For $\mathbf{x} \in \Lambda + c$, it basically assigns the probability $D_{\Lambda+c,s}(\mathbf{x}) = \rho_s(\mathbf{x})/\rho_s(\Lambda + c)$.

Lemma 1. *Let $\mathcal{X}_1$ be a distribution that is indistinguishable from $\mathcal{X}_2$ and M is an efficient non-uniform PPT operation. Then, $M(\mathcal{X}_1)$ is indistinguishable from $M(\mathcal{X}_2)$.*

The min-entropy of a random variable X is $\mathbb{H}_\infty(X) = -\log \max_x \Pr[X = x]$, and measures how well X can be predicted by the best (unbounded) algorithm.

2.2 Lattices

A lattice is an additive subgroup of $\mathbb{R}^n$. For a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ consisting of n linearly independent vectors, we define by Λ the n-dimensional lattice generated by the basis $\mathbf{B}$ where

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{B}\cdot\mathbf{c} = \sum_{i=0}^{n} \mathbf{b}_i \cdot c_i \ :\ \mathbf{c} \in \mathbb{Z}^n\}.$$

We recall two related families of lattices that are come into use in many works. Therefore, let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a full-rank matrix, where n denotes the natural security parameter. The m-dimensional lattice $\Lambda^\perp(\mathbf{A})$

consists of all vectors orthogonal to the parity-check matrix $\mathbf{A}$, i.e., $\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{Ae} = \mathbf{0} \bmod q\}$. Any partition $\Lambda^{\perp}_{\mathbf{v}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{Ae} = \mathbf{v} \bmod q\}$ for $\mathbf{v} \in \mathbb{Z}_q^n$ represents a shift of the lattice $\Lambda^{\perp}(\mathbf{A})$ to $\mathbf{x} + \Lambda^{\perp}(\mathbf{A})$ for arbitrary $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{Ax} = \mathbf{v} \bmod q$. The second lattice $\Lambda(\mathbf{A})$ is generated by the rows of $\mathbf{A}$:

$$\Lambda(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y} = \mathbf{A}^{\top}\mathbf{z} \bmod q \text{ for some } \mathbf{z} \in \mathbb{Z}^n\}\,.$$

Definition 1. *For any n-dimensional lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^* \backslash \{0\}) \leq \epsilon$*

Lemma 2. ***([MR04, Lemma 4.4]).*** *Let Λ be any n-dimensional lattice. Then for any $\epsilon \in (0, 1)$, $s \geq \eta_\epsilon(\Lambda)$, and $\mathbf{c} \in \mathbb{R}^n$, we have*

$$\rho_{s,\mathbf{c}}(\Lambda) \in [\frac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_s(\Lambda)\,.$$

Furthermore, we require the following corollary in order to sample a Gaussian over Λ that is close to being uniformly-distributed modulo a sublattice Λ' in case $s \geq \eta_\epsilon(\Lambda')$ is satisfied.

Corollary 1. ***([GPV08, Corollary 2.8]).*** *Let Λ, Λ' be n-dimensional lattices, with $\Lambda' \subseteq \Lambda$. Then for any $\epsilon \in (0, 1/2)$, any $s \geq \eta_\epsilon(\Lambda')$, and $\mathbf{c} \in \mathbb{R}^n$, the distribution of $(\mathcal{D}_{\Lambda,s,\mathbf{c}} \mod \Lambda')$ is within statistical distance at most 2ϵ of uniform over $(\Lambda \mod \Lambda')$.*

Lemma 3. ***([GPV08, Theorem 3.1]).*** *Let $\Lambda \subset \mathbb{R}^n$ be a lattice with basis $\mathbf{B}$, and let $\epsilon > 0$. We have*

$$\eta_\epsilon(\Lambda) \leq \| \tilde{\mathbf{B}} \| \cdot \sqrt{\ln(2n(1 + 1/\epsilon))/\pi}\,.$$

Specifically, we have $\eta_\epsilon(\Lambda) \leq b \cdot \sqrt{\ln(2n(1 + 1/\epsilon))/\pi}$ for basis $\mathbf{B} = b \cdot \mathbf{I}$ of Λ.

Lemma 4. ***([GPV08, Lemma 5.2])*** *Assume the columns of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generate $\mathbb{Z}_q^n$, and let $\epsilon \in (0, 1/2)$ and $s \geq \eta_\epsilon(\Lambda^{\perp}(\mathbf{A}))$ and $\mathbf{e} \sim \mathcal{D}_{\mathbb{Z}^m,s}$, the distribution of the syndrome $\mathbf{u} = \mathbf{Ae} \mod q$ is within statistical distance 2ϵ of uniform over $\mathbb{Z}_q^n$.*

Below we define the LWE distribution. For our purposes, we only focus on the error sampled by the discrete Gaussian distribution. One can easily define LWE with respect to any error distribution.

Definition 2 (LWE Distribution). *Let n, m, q be integers and χ_e be distribution over $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_q^n$, define the LWE distribution $L^{\mathsf{LWE}}_{n,m,\alpha q}$ to be the distribution over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ obtained such that one first draws $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n \times m}$ uniformly, $\mathbf{e} \leftarrow_R \mathcal{D}_{\mathbb{Z}^m,\alpha q}$ and returns $(\mathbf{A}, \mathbf{b}^{\top}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ with $\mathbf{b}^{\top} = \mathbf{s}^{\top}\mathbf{A} + \mathbf{e}^{\top}$.*

Definition 3 (Learning with Error (LWE)). *Let $(\mathbf{A}, \mathbf{b})$ be a sample from $L^{\mathsf{LWE}}_{n,m,\alpha q}$ and $\mathbf{c}$ be uniformly sampled from $\mathbb{Z}_q^m$.*

- *The Decision Learning with Error (*decision $\mathsf{LWE}_{n,m,\alpha q}$*) problem asks to distinguish between $(\mathbf{A}, \mathbf{b}^{\top})$ and $(\mathbf{A}, \mathbf{c}^{\top})$ for a uniformly sampled secret $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$.*
- *The Search Learning with Error (*search $\mathsf{LWE}_{n,m,\alpha q}$*) problem asks to output the vector $\mathbf{s} \in \mathbb{Z}_q^n$ given LWE sample $(\mathbf{A}, \mathbf{b})$ for a uniformly sampled secret $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$.*

We say decision $\mathsf{LWE}_{n,m,\alpha q}$ (resp. search $\mathsf{LWE}_{n,m,\alpha q}$) is hard if all polynomial time algorithm solves decision $\mathsf{LWE}_{n,m,\alpha q}$ (resp. search $\mathsf{LWE}_{n,m,\alpha q}$) only with negligible probability.

Various algorithms for different tasks such as sampling from $\Lambda^{\perp}(\mathbf{G})$ or inverting LWE instances are found in Appendix B.

3 Learning with Errors Augmented with Auxiliary Data

In this section, we show how one can augment further useful information in the error vectors of LWE samples without necessarily changing its distribution. We call this technique "message embedding" and formulate a modified LWE problem definition, namely the Augmented LWE (A-LWE) problem, where this technique is applied to LWE. We show that certain instantiations of the A-LWE problem are as hard as the original LWE problem.

3.1 Message Embedding

We start explaining the core functionality of our work leading to conceptually new cryptographic applications such as encryption schemes. In particular, we show how to generate vectors that encapsulate an arbitrary message while simultaneously following the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,r}$. This mechanism can be exploited in cryptographic applications in order to embed further information in discrete Gaussian vectors. For instance, we can apply this technique to LWE-based encryption schemes (e.g., [MP12]), that enable the recovery of the error term. As a result we take advantage of an increased message throughput per ciphertext. In Appendix B.3 we provide a short description of how to embed messages in error vectors that are uniformly distributed rather than from the discrete Gaussian distribution.

Let the very simple operations $\mathsf{encode} : \{0,1\}^m \to \mathbb{Z}_q^{m/k}$ and $\mathsf{decode} : \mathbb{Z}_q^{m/k} \to \{0,1\}^m$ allow to bijectively switch between the bit and vector representations. The embedding approach is realized by use of the gadget $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^\top$. A first idea of doing this is to sample a preimage $\mathbf{x} \leftarrow_R D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),r}$ with $\mathbf{v} = \mathsf{encode}(\mathbf{m})$ for an arbitrary message $\mathbf{m} \in \{0,1\}^m$ such that $\mathbf{G}\mathbf{x} \mod q = \mathsf{encode}(\mathbf{m})$ holds. Sampling from $D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),r}$ is performed very efficiently (see Algorithm Sample in Appendix B.1) and can be reduced to samples from $\mathcal{D}_{2\mathbb{Z},r}$ and $\mathcal{D}_{2\mathbb{Z}+1,r}$. However, since the target Gaussian distribution of many cryptographic schemes, such as the LWE encryption schemes, require to have support $\mathbb{Z}^m$, we modify the message to $\mathbf{m} \oplus \mathbf{r}$ prior to invoking the preimage sampler for a randomly chosen vector $\mathbf{r} \leftarrow_R \{0,1\}^m$. Below in Lemma 5 we show that given this setup we indeed obtain a sample $\mathbf{x}$ that is distributed just as $\mathcal{D}_{\mathbb{Z}^m,r}$ with overwhelming probability. To illustrate this approach exemplary let $\mathbf{e} \in \mathbb{Z}^m$ denote the error term with $m \in O(nk)$. We then split the error term $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2) \in \mathbb{Z}^{m_1+m_2}$ into two subvectors, each serving for a different purpose. The second part $\mathbf{e}_2$ is used for message embedding, whereas $\mathbf{e}_1$ provides enough entropy in order to sample a random vector $\mathbf{r}$. To this end, one has to find a proper trade-off for the choice of m_1 and m_2, since a too large value for m_2 implies low entropy of $\mathbf{e}_1$. A reasonable small lower bound is given by $m_1 \geq n$, since the discrete Gaussian vector $\mathbf{e}_1$ has min-entropy of at least $n-1$ bits as per [GPV08, Lemma 2.10].

The message embedding functionality comes at almost no costs. Let k be a factor of m_2. One samples $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{m_1},r}$, computes $\mathbf{v} = \mathsf{encode}(H(\mathbf{e}_1))$ for some random function $H : \{0,1\}^* \to \{0,1\}^{m_2}$ and finally samples a preimage $\mathbf{e}_2 \leftarrow_R D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),r}$ for the syndrome $\mathbf{v} = \mathsf{encode}(\mathbf{m} \oplus \mathbf{v})$ (see Algorithm Sample in Appendix B.1). Following this approach, the message is recovered by computing $\mathbf{m} = H(\mathbf{e}_1) \oplus \mathsf{decode}(\mathbf{G}_{m_2}\mathbf{e}_2 \mod q)$ where $\mathbf{G}_{m_2} = \mathbf{I}_{m_2/k} \otimes \mathbf{g}^\top$. In many cryptographic applications there are different random sources available, which can replace the role of $\mathbf{e}_1$ such that $\mathbf{e}$ is completely used for message embedding.

In the following theorems we prove that it is possible to simulate the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,r}$ (statistically or computationally) by use of a preimage sampler for any full-rank matrix $\mathbf{A}$. This allows for embedding messages in the error vectors of LWE without changing noticeably the LWE distribution. The proofs of these theorems can be found in Appendix D.1. For uniformly distributed error vectors, for which there exist also worst-case reductions [DMQ13, MP13], we provide a sketch in Appendix B.3 using essentially the same arguments.

Lemma 5 (statistical). *Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $k = \lceil \log q \rceil \geq 1$ with $m = l \cdot k$ be an arbitrary full-rank matrix. The statistical distance $\Delta(\mathcal{D}_{\mathbb{Z}^m,r}, \mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r})$ for uniform $\mathbf{v} \leftarrow_R \mathbb{Z}_q^l$ and $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ is negligible.*

Lemma 6 (computational). *Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $k = \lceil \log q \rceil \geq 1$ with $m = l \cdot k$ be an arbitrary full-rank matrix. If the distribution of $\mathbf{v} \in \mathbb{Z}_q^l$ is computationally indistinguishable from the uniform distribution over $\mathbb{Z}_q^l$, then $\mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{A}),r}$ is computationally indistinguishable from $\mathcal{D}_{\mathbb{Z}^m,r}$ for $r \geq \eta_\epsilon(\Lambda^{\perp}(\mathbf{A}))$.*

3.2 Augmented LWE

Based on the message embedding approach as described above, we introduce an alternative LWE definition that extends the previous one in such a way that the error term is featured with additional information. We show how the modified error still coincides with $\mathcal{D}_{\mathbb{Z}^m,r}$ in order to allow a reduction from LWE to our new assumption. Due to space reasons, we provide the proofs of our theorems in Appendix D.

We make use of the gadget matrix $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^\top$ constructed as described in Appendix B.1. For simplicity, assume $q = 2^k$. For general q, the preimage sampling algorithm for $\Lambda^{\perp}(\mathbf{G})$ is more involved (see [MP12]).

Definition 4 (Augmented LWE Distribution). *Let n, m, m_1, m_2, k, q be integers with $k = \log q$ and $m = m_1 + m_2$, where $k \mid m_2$. Let $H : \mathbb{Z}_q^n \times \mathbb{Z}_q^{m_1} \to \{0,1\}^{m_2}$ be a function. Let $\mathbf{G}_{m_2} = \mathbf{I}_{m_2/k} \otimes \mathbf{g}^\top \in \mathbb{Z}_q^{m_2/k \times m_2}$. For $\mathbf{s} \in \mathbb{Z}_q^n$, define the A-LWE distribution $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ with $\mathbf{m} \in \{0,1\}^{m_2}$ to be the distribution over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ obtained as follows:*

- *Sample $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n \times m}$ and $\mathbf{e}_1 \leftarrow_R \mathcal{D}_{\mathbb{Z}^{m_1},\alpha q}$.*
- *Set $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}, \mathbf{e}_1) \oplus \mathbf{m}) \in \mathbb{Z}_q^{m_2/k}$.*
- *Sample $\mathbf{e}_2 \leftarrow_R \mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),\alpha q}$.*
- *Return $(\mathbf{A}, \mathbf{b}^\top)$ where $\mathbf{b}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ with $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$.*

Accordingly, we define the augmented LWE problem(s) as follows. As opposed to the traditional LWE, augmented LWE blinds, in addition to the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, also some (auxiliary) data $\mathbf{m} \in \{0,1\}^{m_2}$. Thus, we have an additional assumption that the message $\mathbf{m}$ is hard to find given A-LWE samples. Note that the decision version requires that any polynomial bounded number of samples $(\mathbf{A}, \mathbf{b}^\top)$ from the A-LWE distribution are indistinguishable from uniform random samples in $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$. Its hardness implies that no information about $\mathbf{s}$ *and* $\mathbf{m}$ is leaked through A-LWE samples. In some scenarios, e.g., in security notions of an encryption scheme, the adversary may even choose the message $\mathbf{m}$. Hence, we require in the corresponding problems that their hardness holds with respect to A-LWE distributions with adversarially chosen message(s) $\mathbf{m}$ except for the search problem of $\mathbf{m}$.

Definition 5 (Augmented Learning with Errors (A-LWE)).
Let n, m_1, m_2, k, q be integers with $k = \log q$. Let H be some function.

- *The Decision Augmented Learning with Errors (decision A-LWE$^H_{n,m_1,m_2,\alpha q}$) problem asks upon input $\mathbf{m} \in \{0,1\}^{m_2}$ to distinguish in polynomial time (in n) between samples $(\mathbf{A}_i, \mathbf{b}_i^\top) \leftarrow_R L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ and uniform random samples from $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$ for a secret $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$.*
- *The Search-Secret Augmented Learning with Errors (search-s A-LWE$^H_{n,m_1,m_2,\alpha q}$) problem asks upon input $\mathbf{m} \in \mathbb{Z}_q^{m_2/k}$ to output in polynomial time (in n) the vector $\mathbf{s} \in \mathbb{Z}_q^n$ given polynomially many samples $(\mathbf{A}_i, \mathbf{b}_i) \leftarrow_R L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ for secret $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$.*
- *The Search-Message Augmented Learning with Errors (search-m A-LWE$^H_{n,m_1,m_2,\alpha q}$) problem asks to output in polynomial time (in n) the vector $\mathbf{m}$ given polynomially many A-LWE samples $(\mathbf{A}_i, \mathbf{b}_i)$ for a secret $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ and $\mathbf{m} \leftarrow_R \{0,1\}^{m_2}$.*

We say that decision/search-s/search-m A LWE$^H_{n,m_1,m_2,\alpha q}$ *is hard if all polynomial time algorithms solve the* decision/search-s/search-m A LWE$^H_{n,m_1,m_2,\alpha q}$ *problem only with negligible probability.*

Throughout the paper, the function H will be a cryptographic hash function modeled as a random oracle. For this reason we simplify the notation and denote by $\mathsf{decision/search\text{-}s/search\text{-}m\ A\ LWE}_{n,m_1,m_2,\alpha q}$ the ALWE problems where H is specified to be a random oracle in the A-LWE distribution.

In the following, we show that if the function H is instantiated by a random oracle, the hardness of LWE is reducible to the hardness of A-LWE. To this end, we show that the LWE and A-LWE distribution are computationally indistinguishable if we assume that the former search problem is hard and the inputs to function H have sufficient entropy in each sample given previous samples.

Theorem 1. *Let λ be the security parameter. Let n, m, m_1, m_2, k, q be integers where $k = \lceil \log q \rceil$, $m = m_1 + m_2$. Let $H : \mathbb{Z}_q^n \times \mathbb{Z}_q^{m_1} \to \{0,1\}^{m_2}$ be a hash function modeled as a random oracle. Let $\alpha q \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$ for a real $\epsilon > 0$. Furthermore, denote by χ_s and χ_{e_1} the distributions of the random vectors $\mathbf{s}$ and $\mathbf{e}_1$ involved in each A-LWE sample. If $\mathsf{search\ LWE}_{n,m,\alpha q}$ is hard and $\mathbb{H}_\infty(\mathbf{s}, \mathbf{e}_1) > \lambda$, then $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ is computationally indistinguishable from $L^{\mathsf{LWE}}_{n,m,\alpha q}$ for arbitrary $\mathbf{m} \in \{0,1\}^{m_2}$.*

Note that if the first error part $\mathbf{e}_1$ has entropy exceeding the security parameter λ, the (computational) entropy induced by $\mathbf{s}$ is not required. This is important, since a distinguisher could ask for many A-LWE samples using the same secret $\mathbf{s}$ as input to the hash function. However, as typical in encryption schemes (e.g., in [Pei09, LPR10, LP11, MP12] and in ours), if we fix a random matrix $\mathbf{A}$ and sample fresh secret vectors $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random for each A-LWE sample, we can indeed choose m_1 to be zero. This corresponds to the case, where an A-LWE sample is drawn once for every fresh secret $\mathbf{s}$ resulting in essentially unrelated A-LWE instances. Hence, the secret $\mathbf{s}$ provides the sufficient randomness required as input to H.

It is not hard to see that Theorem 1 immediately entails the following statement.

Theorem 2. *Let n, m, m_1, m_2, k, q be integers with $k = \log q$ and $m = m_1 + m_2$. Let H be a random oracle as defined in Theorem 1. Let $\alpha q \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$ for a real $\epsilon > 0$. Furthermore, denote by χ_s and χ_{e_1} the distributions of the random vectors $\mathbf{s}$ and $\mathbf{e}_1$ involved in each A-LWE sample. If $\mathbb{H}_\infty(\mathbf{s}, \mathbf{e}_1) > \lambda$, then the following statements hold.*

1. *If $\mathsf{search\ LWE}_{n,m,\alpha q}$ is hard, then $\mathsf{search\text{-}s\ A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ is hard.*
2. *If $\mathsf{decision\ LWE}_{n,m,\alpha q}$ is hard, then $\mathsf{decision\ A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ is hard.*
3. *If $\mathsf{decision\ LWE}_{n,m,\alpha q}$ is hard, then $\mathsf{search\text{-}m\ A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ is hard.*

Proof. As per Theorem 1, $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ is computationally indistinguishable from $L^{\mathsf{LWE}}_{n,m,\alpha q}$. This proves the hardness of $\mathsf{decision\ A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ and $\mathsf{search\text{-}m\ A\text{-}LWE}_{n,m_1,m_2,\alpha q}$. And by essentially the same arguments we also deduce the hardness of $\mathsf{search\text{-}s\ A\text{-}LWE}_{n,m_1,m_2,\alpha q}$, because solving the search problem implies distinguishability of A-LWE instances from uniform due to the knowledge of $(\mathbf{s}, \mathbf{e})$ and by Theorem 1 we obtain distinguishability of LWE instances from uniform, hence a contradiction. □

3.3 Generic Encryption Scheme from A-LWE

In what follows we provide a generic construction of an A-LWE based encryption scheme. Due to our new feature of embedding messages in the error term, we can employ any trapdoor function that allows for error-term recovery. We restrict to the case, where function H takes only $\mathbf{s}$ as input (i.e., $m_1 = 0$) as discussed above.

Let $\mathsf{TDF} = (\mathsf{KeyGen}, g, g^{-1})$ be a trapdoor function with $g_{\mathbf{A}}(\mathbf{x}, \mathbf{y}) := \mathbf{x}^\top \mathbf{A} + \mathbf{y}^\top \in \mathbb{Z}^m$. The algorithm KeyGen outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, that is close to uniform, with an associated trapdoor $\mathbf{T}$ used to invert $g_{\mathbf{A}}$. The trapdoor function satisfies $g_{\mathbf{A}}^{-1}(\mathbf{T}, \mathbf{c}) = (\mathbf{x}, \mathbf{y})$ with $\mathbf{c} = g_{\mathbf{A}}(\mathbf{x}, \mathbf{y})$ for arbitrary $\mathbf{x} \in \mathbb{Z}_q^n$ and properly chosen $\mathbf{y} \in \mathbb{Z}^m$.

Our generic encryption scheme from A-LWE is constructed as follows:

$\mathsf{KGen}(1^n)$: Generate public key $\mathsf{pk} := \mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with trapdoor $\mathsf{sk} := \mathbf{T}$ where $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TDF.KeyGen}(1^n)$.

$\mathsf{Enc}(\mathsf{pk}, \mathbf{m} \in \{0,1\}^l$ with $0 \leq l \leq m)$: Sample $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ and compute $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m}) \in \mathbb{Z}_q^{m/k}$. Then, sample $\mathbf{e} \leftarrow_R \mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),\alpha q}$. The ciphertext is given by $\mathbf{c} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$.

$\mathsf{Dec}(\mathsf{sk}, \mathbf{c})$: Compute $g_{\mathbf{A}}^{-1}(\mathbf{T}, \mathbf{c}) = (\mathbf{s}, \mathbf{e})$. Return $\mathbf{m} = \mathsf{decode}(\mathbf{Ge} \bmod q) \oplus H(\mathbf{s})$.

The generic construction is mainly based on the capability of the scheme to recover the error vector. Thus, the underlying trapdoor construction acts as a black box granting full access to the secret $\mathbf{s}$ and the error term $\mathbf{e}$, when applying the secret trapdoor on a corresponding A-LWE instance. Once having revealed the error term, the message is recovered via the last step of the scheme involving the simple matrix $\mathbf{G}$ and the function $H(\cdot)$. Improving the quality of the trapdoor and its inversion algorithm directly impacts the efficiency of the encryption scheme, since decoding of the message from $\mathbf{e}$ is performed very efficiently.

Theorem 3. *The generic encryption scheme above is secure assuming the hardness of* decision A-LWE$_{n,0,m,\alpha q}$ *for* $\alpha q \geq 2\sqrt{n} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi} \geq \eta_\epsilon(\Lambda_q^{\perp}(\mathbf{G}))$.

Proof. Ciphertexts generated according to the generic encryption scheme from above correspond to plain A-LWE samples with $m_1 = 0$. By assumption decision A-LWE$_{n,0,m,\alpha q}$ is hard, and consequently, an adversary is not able to distinguish a challenge ciphertext from uniformly chosen samples. □

One can apply Theorem 1 and Theorem 2 to have a direct reduction from traditional LWE.

4 New Chosen-Ciphertext Secure Encryption Schemes

Due to the new functionality of embedding messages in error vectors, we are able to propose a novel secure encryption scheme providing full CCA security when adopting the tagging approach presented in [Kil06, CHK04]. In fact, we get this feature for free, if we instantiate our generic construction from Section 3.3 with the trapdoor provided in [MP12]. More specifically, the authors add a tag u to the matrix $\mathbf{A}$ such that the modified matrix $\mathbf{A}_u$ keeps changing for every encryption query.

Originally, in almost all previous encryption schemes ciphertexts are build in a one-time pad manner by adding the message to a random-looking vector coming from an LWE instance. By our modifications, we omit the way of encoding messages and the restrictions made to the parameters. Our aim is to let the ciphertexts resemble an ordinary A-LWE instance such that the hardness of the scheme can be directly reduced to the plain A-LWE problem. Indeed, the error term hides the message while following the required distribution. This allows for more flexibility, efficiency and larger messages per ciphertext at no costs. Even more, this greatly simplifies the security proof. As we show later, we can even lift up the security to publicly-detectable RCCA (pd-RCCA) with a simple trick ensuring non-malleability of ciphertexts. When applying these functionalities to the error term in the CCA1-secure scheme due to [MP12], the message throughput is at least twice as large while simultaneously providing pd-RCCA security instead of CCA1, as before. In addition to that, we give an intuition of how to get a CCA2-secure encryption scheme involving only minor modifications.

4.1 CCA1-Secure Encryption Scheme

We start with a detailed description of the CCA1 secure encryption scheme and the involved algorithms. Let $H : \mathbb{Z}_q^n \to \{0,1\}^m$ be some function. Let $\mathcal{R} = \mathbb{Z}_q[x]/(f(x))$ be a ring as constructed in [MP12], where $f(x)$ denotes a monic irreducible polynomial of degree n. Furthermore, let $h : \mathcal{R} \to \mathbb{Z}_q^{n \times n}$ be an injective ring homomorphism mapping elements $a \in \mathcal{R}$ to the matrix $h(a)$. By $\mathcal{U} = \{u_1, \ldots, u_\ell\}$ we denote a large set with "unit differences" property. That is, for any two ring elements a_i and $a_j \in \mathcal{R}^*$ with $i \neq j$ we have $a_i - a_j \in \mathcal{R}^*$ and $h(a_i - a_j) = h(a_i) - h(a_j)$ is invertible. By $\mathbf{G}_m$ we denote the matrix $\mathbf{I}_{m/k} \otimes \mathbf{g}^\top$. Our encryption scheme works as follows.

$\mathsf{KGen}(1^n)$: Let $k = \log q$ and $m, \bar{m} > 0$ with $k \mid m$ and $m = \bar{m} + nk$. Invoking $\mathsf{TDF.KeyGen}(1^n)$ outputs keys $(\mathbf{A}, \mathbf{R})$, where $\mathbf{A} = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R}\ ]$ for randomly selected matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times \bar{m}}$ and $\mathbf{R} \leftarrow_R \mathcal{D}$ is sampled from a desired distribution $\mathcal{D}$, typically the discrete Gaussian distribution. For instance, one chooses $\bar{m} = nk$ and $\mathcal{D} = D_{\mathbb{Z},t}^{\bar{m} \times nk}$ for $t \in \omega(\sqrt{\log n})$. The public and secret key are given by $\mathsf{pk} = \mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathsf{sk} = \mathbf{R} \in \mathbb{Z}_q^{\bar{m} \times nk}$.

$\mathsf{Enc}(\mathsf{pk}, \mathbf{m} \in \{0,1\}^l$ with $0 < l < m)$: Select a nonzero $u \in \mathcal{U}$. Set $\mathbf{A}_u = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ]$ with $\mathbf{G}_{nk} = \mathbf{I}_n \otimes \mathbf{g}^\top$. Then, select $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow_R \mathcal{D}_{\Lambda_{\mathbf{v}}^\perp(\mathbf{G}_m), \alpha q}$ where $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m}) \in \mathbb{Z}_q^{m/k}$ and $\alpha q \geq 2\sqrt{n} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$. Output the ciphertext

$$\mathbf{c} = (u, \mathbf{b}) \in \mathcal{U} \times \mathbb{Z}_q^m \text{ with } \mathbf{b}^\top = g_{\mathbf{A}_u}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^\top \mathbf{A}_u + \mathbf{e}^\top \bmod q\ .$$

$\mathsf{Dec}(\mathsf{sk}, \mathbf{c})$: Determine $\mathbf{A}_u = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ]$.

1. If parsing $\mathbf{c}$ causes an error or $u = 0$, output $\perp$. Otherwise invoke the inversion algorithm in Appendix B.2 with input parameters $(\mathbf{R}, \mathbf{A}_u, \mathbf{b})$, which outputs a failure $\perp$ or $g_{\mathbf{A}_u}^{-1}(\mathbf{b}^\top) = (\mathbf{s}', \mathbf{e}')$.
2. Check $\|\mathbf{e}'\| \leq \alpha q \sqrt{m}$. If it is satisfied, compute $\mathbf{r} = H(\mathbf{s}')$ and $\mathbf{m} = \mathbf{r} \oplus \mathsf{decode}(\mathbf{G}_m \mathbf{e}' \mod q)$.
3. Output $\mathbf{m}$ as the message.

Theorem 4. *The encryption scheme above is CCA1-secure assuming the hardness of* $\mathsf{decision\ A\text{-}LWE}_{n,0,m,\alpha q}$ *for* $\alpha q \geq 2\sqrt{n} \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$.

Proof. The proof is greatly simplified as compared to [MP12], since we are not required to perform any transformations to the initial A-LWE samples. In fact, we draw samples $(\mathbf{A}, \mathbf{b}^\top) \leftarrow_R L_{n,0,m,\alpha,q}^{\mathsf{A\text{-}LWE}}(\mathbf{m})$ from the A-LWE distribution, where $\mathbf{b}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$, $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n \times m}$ and $\mathbf{e} \leftarrow_R \mathcal{D}_{\mathbb{Z}^{m_1}, \alpha q} \times \mathcal{D}_{\Lambda_{\mathbf{v}}^\perp(\mathbf{G}), \alpha q}$ with $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m})$ and $\alpha q \geq 2\sqrt{n} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi} \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$. Distinguishing these samples from random ones is as hard as solving $\mathsf{decision\ A\text{-}LWE}_{n,0,m,\alpha q}$ for the given parameters (see Theorem 3).
Encryption queries in our scheme are represented by ordinary A-LWE queries, thus we can give a direct reduction. Indeed, we have $\mathbf{b}_1 = \mathbf{s}^\top \bar{\mathbf{A}} + \mathbf{e}_1 \mod q$ and $\mathbf{b}_2 = \mathbf{s}^\top (h(u)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}) + \mathbf{e}_2 \mod q$, where $(\bar{\mathbf{A}}, h(u)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R})$ is statistically close to uniform by the leftover hash lemma and $h(u)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}$ is $negl(n)$-uniform for any choice of $u \in \mathcal{U}$ following essentially the same argumentation as in [MP12]. Hence, the advantage of the adversary in the CCA1 security game with our scheme from above is negligible. □

For instance, if one chooses $m = c \cdot nk$ corresponding to a statistical instantiation of the scheme – that is, $\mathbf{A}$ is statistically close to uniform – one can encrypt messages of length $c \cdot nk$ bits.

High Data Load Encryption. In certain application scenarios one wishes to encrypt huge amounts of data such as secure backups, etc. In this case, a fast encryption and decryption engine is desired. The key idea underlying this goal is to extend the public key by an arbitrary number of random vectors in order to ensure a more efficient encryption scheme at essentially the same security level. Assume, the initial public key is given by $\mathbf{A}_u = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}\ ] \in \mathbb{Z}^{n \times m}$. Extending the public key to $\mathbf{A}_u^{ext} = [\ \mathbf{A}'\ |\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}\ ] \in \mathbb{Z}^{n \times (m'+m)}$ allows one to encrypt $c = \lfloor (m' + \bar{m} + nk)/k \rfloor \cdot k$ bits of data simultaneously using the message embedding approach from above. This is obviously not possible with the CCA encryption scheme in [MP12] (see Appendix C.1), since the maximum message size is solely determined by n and k amounting to nk bits. The input to the inversion algorithm described in Appendix B.2 is the modified trapdoor $[\mathbf{0}\ \ \mathbf{R}^\top\ \ \mathbf{I}]^\top$ and the ciphertext $\mathbf{b}$, which then recovers $\mathbf{s}$ and $\mathbf{e}'' = (\mathbf{e}', \mathbf{e})$. Interestingly, we observe that the norm bound on the error term is only related to the part $\mathbf{e}$ due to $\left\|\mathbf{e}'' \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \\ \mathbf{I} \end{bmatrix}\right\| = \left\|\mathbf{e} \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix}\right\| < q/(4 \cdot \sqrt{s_1(\mathbf{R})^2 + 1})$ in our scheme. Thus, extending the public key has no impact on how to choose $\mathbf{e}'$ in the decryption routine, except for ensuring a reasonable level of

security of the underlying A-LWE instance. We now briefly explain the benefits of such a construction for a predefined amount of data.

As an advantage for encryption, we do not need to initiate so many generations of $\mathbf{s}$, u and computations of $h(u)$ as compared to the original public key, because we use the same $\mathbf{s}$ and u for a larger message size. For decryption, one notices that the inversion algorithm works almost as fast as with the original public key, since

$$(\mathbf{b}', \mathbf{b})^\top \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{s}^\top \mathbf{A}_u^{ext} \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \\ \mathbf{I} \end{bmatrix} + \mathbf{e}''^\top \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{s}^\top \mathbf{G} + \mathbf{e}^\top \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} .$$

This strategy is not directly applicable to the CCA1-secure encryption scheme presented in [MP12]. However, in combination with our technique to embed auxaliary data in the error term it is possible to furtherly take advantage of this feature.

4.2 Lifting to RCCA security

The notion of CCA2 security is a desirable feature in most encryption schemes. Many of the current solutions are generic and thus coupled to overhead and loss of efficiency. Generic solutions to CCA2-secure LWE-based encryption schemes include the usage of strongly unforgeable one-time signatures [DDN00] or message authentication codes with a weak form of bit commitments [BCHK07]. Regarding the first approach the user is required to encrypt the verification key vk together with the message yielding the ciphertext $\mathbf{c}$. Next, the signature is computed over the ciphertext and is subsequently transmitted together with $\mathbf{c}$ and vk to the receiver.

In many encryption schemes we require only RCCA security, which is strictly stronger than CCA1, but slightly weaker than CCA2 security. It has been shown in [CKN03] that RCCA security is sufficient for many of the major applications of CCA secure encryption such as authentication and key exchange etc. In particular, it ensures non-malleability in any ways that alter the message.

We introduce here an elegant way to easily modify the scheme from above in order to achieve publicly detectable RCCA security (pd-RCCA) without suffering from the disadvantages of generic solutions. In fact, this is the first lattice-based construction that allows for this feature. In our scheme, ciphertexts that decrypt to the same plaintext as the challenge ciphertext $(u^*, \mathbf{c}^*)$ can publicly be detected by any party.

As an advantage by our construction, the transmitted data is solely restricted to the ciphertext $\mathbf{c}$ and we omit to send additional verification keys and signatures as compared to the generic approach. Let $q = 2^k$, $m = m_1 + m_2$ with $k = \log q$ and $k \mid m_1, m_2$. Furthermore, let $H_1 : \{0,1\}^* \to \mathbb{Z}_q^{m_1/k}$ and $H_2 : \mathbb{Z}_q^n \to \{0,1\}^{m_2}$ be cryptographic hash functions modeled as random oracle. The key concept of the RCCA-secure encryption scheme consists in replacing the upper-part of the error term by $\mathbf{e}_1 \leftarrow \mathcal{D}_{\Lambda^\perp_{\mathbf{v}_1}(\mathbf{G}_{m_2}),\alpha q}$ with $\mathbf{v}_1 = H_1(\mathbf{s}, \mathbf{e}_2, u)$. It allows the receiver to check for modifications made to the ciphertext or its ingredients $(\mathbf{s}, \mathbf{e}_2, u)$ and is to be thought of a GPV signature for the public matrix $\mathbf{G}_{m_2}$. The lower part of the error vector, however, remains essentially unchanged $\mathbf{e}_2 \leftarrow \mathcal{D}_{\Lambda^\perp_{\mathbf{v}_2}(\mathbf{G}_{m_2}),\alpha q}$, where $\mathbf{v}_2 = \mathsf{encode}(H_2(\mathbf{s}) \oplus \mathbf{m})$. We then change the encryption and decryption routine of the CCA1-secure scheme in Section 4.1 as follows:

$\mathsf{Enc}(pk, \mathbf{m} \in \{0,1\}^{m_1})$: Select a nonzero $u \in \mathcal{U}$. And determine $\mathbf{A}_u = [\ \bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ]$, with $\mathbf{G}_{nk} = \mathbf{I}_n \otimes \mathbf{g}^\top$. Then, select $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ and $\mathbf{e}_2 \leftarrow \mathcal{D}_{\Lambda^\perp_{\mathbf{v}_2}(\mathbf{G}_{m_2}),\alpha q}$ where $\mathbf{v}_2 = \mathsf{encode}(H_2(\mathbf{s}) \oplus \mathbf{m})$ and $\alpha q \geq 2\sqrt{n} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$. Subsequently, sample $\mathbf{e}_1 \leftarrow \mathcal{D}_{\Lambda^\perp_{\mathbf{v}_1}(\mathbf{G}_{m_2}),\alpha q}$ with $\mathbf{v}_1 = H_1(\mathbf{s}, \mathbf{e}_2, u)$. The ciphertext is given by

$$\mathbf{c} = (u, \mathbf{b}) \in \mathcal{U} \times \mathbb{Z}_q^m \text{ with } \mathbf{b}^\top = \mathbf{s}^\top \mathbf{A}_u + (\mathbf{e}_1, \mathbf{e}_2)^\top \mod q\,.$$

$\mathsf{Dec}(\mathsf{sk}, \mathbf{c})$: Determine $\mathbf{A}_u = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ]$.

1. If parsing $\mathbf{c}$ causes an error or $u = 0$, output $\perp$. Otherwise invoke the inversion algorithm Invertin Appendix B with input parameters $(\mathbf{R}, \mathbf{A}_u, \mathbf{b})$, which outputs the values $\mathbf{s}'$ and $\mathbf{e}'$ or a failure $\perp$.
2. (Non-malleability) Check $\mathbf{G}_{m_1}\mathbf{e}'_1 \stackrel{?}{=} H_1(\mathbf{s}, \mathbf{e}'_2, u)$.
3. (Message Recovery) Check $\|\mathbf{e}'\| \leq \alpha q\sqrt{m}$. If it is satisfied, compute $\mathbf{r} = H(\mathbf{s}')$ and $\mathbf{m} = \mathbf{r} \oplus \mathsf{decode}(\mathbf{G}_{m_2}\mathbf{e}'_1 \bmod q)$.
4. Output $\mathbf{m}$ as the message.

Theorem 5. *The scheme from above is pd-RCCA secure assuming the hardness of* decision A-LWE$_{n,0,m,\alpha q}$ *for* $\alpha q \geq 2\sqrt{n} \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$ *in the random oracle model.*

The proof of Theorem 5 can be found in Appendix D.3. Similarly, one can apply this technique to [MP12] in order to obtain pd-RCCA security. The construction of [MP12] can be found in Appendix C.1. One requires to compute the hash value on all possible malleable variables that could be changed when calling the encryption routine.

4.3 Upgrading RCCA to CCA2

There exist different approaches to turn any RCCA-secure encryption scheme into an CCA2-secure scheme. A generic way of doing this, is presented in [CKN03], which does not involve any further computational assumptions. Specifically, it aims at combining an RCCA-secure public key encryption scheme with an CCA-secure symmetric key encryption scheme in order to attain CCA2-security. It is well-known that an CCA-secure symmetric encryption scheme can be build from any secure encryption scheme. Hence, the resulting CCA2-secure scheme is said to be efficient if the underlying encryption schemes are efficient.

The second approach, that we suggest, requires to append the hash value $H'(\mathbf{e}_1)$ to the ciphertext output by the RCCA-secure scheme presented in the previous section. We note here that it suffices if hash function H' is merely collision-resistance. Following this, the new challenge ciphertext consists of the tuple $(u^*, \mathbf{b}^*, H(\mathbf{e}_1^*))$. We now analyze the behavior of the decryption oracle, in case the ciphertext has been modified. Assuming RCCA-security we only focus on the last case in the security game $\mathbf{Exp}^{\mathsf{ind-cca2}}_{\mathcal{E},\mathcal{A}}(n)$ (see Appendix A). Thus, we need only to prove that the adversary cannot modify the error vector $\mathbf{e}_1$, since all other cases are covered due to RCCA-security.

1. If $\mathbf{b}_1 \neq \mathbf{b}_1^*$,$\mathbf{b}_2 = \mathbf{b}_2^*$ and $H(\mathbf{e}_1) = H(\mathbf{e}_1^*)$, then the attacker must have found a collision.
2. If $\mathbf{b}_1 \neq \mathbf{b}_1^*$, $\mathbf{b}_2 = \mathbf{b}_2^*$ and $H(\mathbf{e}_1) \neq H(\mathbf{e}_1^*)$, then $\mathsf{Dec}(\cdot)$ outputs $\perp$, because otherwise the attacker must have known $\mathbf{e}_1 = \mathbf{b}_1 - \mathbf{s}^{*\top}\bar{\mathbf{A}}$ and hence $\mathbf{s}^*$, which is equivalent to solving A-LWE.

This proofs that the attacker is not able to derive new information about the encrypted message, even when he is given access to the decryption oracle after issuance of the challenge ciphertext.

4.4 Asymmetric Authenticated Encryption

Theoretically, one can embed to our encryption schemes as second message – the one which is embedded to the error term – a digital signature authenticating the plaintext. One could use any signatures which satisfy the length requirements to be a bit string of length m (if not satisfied, one expands the public key). However, many of the well-known lattice-based signature schemes produce signatures that are already distributed just like discrete Gaussians. Such a choice causes short signatures to be selected with higher probability and more importantly it allows to decouple the distribution of the signatures from the secret keys. Considering the fact that the error term is sampled from a discrete Gaussian distribution as well, we can build an authenticated encryption scheme, where parts of the error term are directly replaced

by signatures. Doing this, we achieve security guarantees similar to CCA2 and moreover the receiver is able to identify the originator of the message through his embedded signature.

Lattice-based signatures due to [DDLL13, Lyu12, GPV08, MP12] are qualified for this task. It has recently been shown that these schemes are very efficient and perform very well in practice. Thus, the efficiency directly impacts our schemes allowing for fast encryption in conjunction with an authentication subroutine. Hence, there is no need for the usage of strongly unforgeable one-time signatures as before, where the verification keys have to be encrypted before transmitting the ciphertext together with the signature and the verification keys to the receiver. This obviously increases the data size to be sent and further does not allow identifying the originator of the message. Indeed, this functionality is also achieved by the more efficient non-malleability mechanisms presented in the previous sections. Notably, all of our proposals can be applied essentially without increasing the ciphertext size.

Let $q = 2^k$, $m = m_1 + m_2$ with $k = \log q$ and $k \mid m_2$. Furthermore, let $H : \{0,1\}^* \to \mathbb{Z}_q^{m_2/k}$ be a hash function and $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a signature scheme where a part of the resulting signature is distributed following the discrete Gaussian distribution such as the efficient constructions given in [DDLL13, Lyu12, GPV08, MP12]. For instance, in [DDLL13] signatures are represented by the tuple $(\mathbf{c}, \mathbf{z})$, where $\mathbf{z}$ is distributed as $\mathcal{D}_{\mathbb{Z}^{m_1},s}$ and $\mathbf{c}$ occupies very few bits ($\approx 100 - 200$ bits) and can thus be appended to the ciphertext. Essentially, the same approach is taken for the GPV signature scheme [GPV08, MP12].

$\mathsf{Enc}(\mathsf{pk}, \mathbf{m} \in \{0,1\}^{m_1})$: Select a nonzero $u \in \mathcal{U}$. And determine $\mathbf{A}_u = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ]$, with $\mathbf{G}_{nk} = \mathbf{I}_n \otimes \mathbf{g}^\top$. Then, select $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ and $\mathbf{e}_2 \leftarrow \mathcal{D}_{\Lambda^\perp_{\mathbf{v}_2}(\mathbf{G}_{m_2}),\alpha q}$ where $\mathbf{v}_2 = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m})$ and $\alpha q \geq 2\sqrt{n} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$. Subsequently, sample $\mathbf{e}_1 = \mathsf{Sign}(\mathbf{s}, \mathbf{e}_2, u)$. The ciphertext is given by

$$\mathbf{c} = (\mathbf{u}, \mathbf{b}) \in \mathcal{U} \times \mathbb{Z}_q^m \text{ with } \mathbf{b}^\top = \mathbf{s}^\top \mathbf{A}_u + (\mathbf{e}_1, \mathbf{e}_2)^\top \mod q\,.$$

$\mathsf{Dec}(sk, \mathbf{c})$: Determine $\mathbf{A}_u = [\ \bar{\mathbf{A}}\ |\bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ]$.

1. If parsing $\mathbf{c}$ causes an error or $u = 0$, output $\perp$. Otherwise invoke the inversion algorithm in Appendix B.2 with input parameters $(\mathbf{R}, \mathbf{A}_u, \mathbf{b})$, which outputs the values $\mathbf{s}'$ and $\mathbf{e}'$ or a failure $\perp$.
2. (Signature Verification) Check signature, $\mathsf{Verify}(\mathbf{s}, \mathbf{e}'_2, u) \stackrel{?}{=} 1$.
3. (Message Recovery) Check $\|\mathbf{e}'\| \leq \alpha q\sqrt{m}$. If it is satisfied, compute $\mathbf{r} = H(\mathbf{s}')$ and $\mathbf{m} = \mathbf{r} \oplus \mathsf{decode}(\mathbf{G}_{m_2}\mathbf{e}'_1 \bmod q)$.
4. Output $\mathbf{m}$ as the message.

We shortly discuss how to instantiate the scheme. If the signature scheme in [MP12] is applied, one requires to use the probabilistic GPV signature scheme, which outputs signatures $(\mathbf{r}, \mathbf{z})$ being distributed as $\mathcal{D}_{\Lambda^\perp_{H(\mathbf{s},\mathbf{e}_2,u,\mathbf{r})}(\mathbf{B}),s} \sim \mathcal{D}_{\mathbb{Z}^{m_1},s}$ with $\mathbf{r} \leftarrow_R \{0,1\}^l$ according to Lemma 5. This is due to the random seed that is appended before generating the signature. As described in Section 4.2, the signature $\mathbf{e}_1 \leftarrow \mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{B}),s}$ has to be computed on all malleable variables $\mathbf{v} = H(u, \mathbf{e}_2, \mathbf{s}, \mathbf{r})$. Embedding signatures in the error vector has a flavor of CCA2 security, when considering the last case in the proof of Theorem 5. In this particular case, building a replay is equivalent to $\mathbf{e}_1 - \mathbf{e}_1^* \in \Lambda_q^\perp(\mathbf{B})$ and the length of $\mathbf{e}_1 - \mathbf{e}_1^*$ is short, which is obviously a difficult task assuming the hardness of SIS. Note, that the Gaussian parameter $s = \tilde{O}(n)$ is also used for sampling the second part of the error term $\mathbf{e}_1 \leftarrow \mathcal{D}_{\Lambda^\perp_{H(\mathbf{s})\oplus\mathbf{m}}(\mathbf{G}_{m_1}),s}$ containing the message, since $s \geq 2\sqrt{\ln(2n(1+1/\epsilon))/\pi} \geq \eta_\epsilon(\Lambda^\perp(\mathbf{B}))$ is always satisfied. It is also noteworthy to mention, that it can be reasonable to adopt the high data load approach from Section 4.1, where $\mathbf{e}_1$ is suitable for signatures. Similar steps are taken when $\mathbf{e}_1$ is replaced by a signature generated according to the signature schemes [DDLL13, Lyu12] (see Appendix C.2).

4.5 Enhancing the Symmetric-Key Encryption Scheme from [BV11b]

In this section, we show that our message embedding technique can be applied to the symmetric key encryption scheme as provided in [BV11b]. The authors propose a symmetric key encryption scheme that

entails the properties of a somewhat homomorphic encryption scheme. Based on this construction they build a fully homomorphic public key encryption scheme that is KDM-secure, meaning that the scheme remains secure even when encrypting polynomial functions of the secret key. In fact, the symmetric key encryption scheme is a very efficient construction that is directly based on ring-LWE (or PLWE [BV11b]). Ciphertexts are built in a one-time pad manner. The secret key shared among the participants is the secret $\mathbf{s} \in \mathcal{R}_q = \mathbb{Z}[X]/\langle f(X)\rangle$ coming from ring-LWE. One typically chooses $f(X) = X^m + 1$ for $m = 2^l$ and $l \in \mathbb{N}$. We now present the encryption scheme that is enriched with our message embedding technique required to increase the message throughput. But for the sake of simplicity, we ignore the homomorphic properties of the scheme with regard to the message $\mathbf{m}_1$. Interested readers are referred to the respective descriptions in [BV11b].

$\mathsf{KGen}(1^n)$: Let $n_1, n_2 > 0$ such that $n_1 + n_2 = n$, $k \mid n_2$ and let t be coprime to q. Sample the coefficients of $\mathbf{s} \leftarrow_R \mathcal{R}_q$ and set the secret key $sk := \mathbf{s}$

$\mathsf{Enc}(sk, \mathbf{m}_1 \in \mathcal{R}_t = \mathbb{Z}_t[X]/\langle f(X)\rangle, \mathbf{m}_2 \in \{0,1\}^l$, with $0 < l < n_2$): The first message is encoded to be a polynomial in $\mathcal{R}_t$. Sample $(\mathbf{a}, \mathbf{as} + t\mathbf{e})$, where $\mathbf{a} \leftarrow_R \mathcal{R}_q$. The coefficients of $\mathbf{e}_1$ and $\mathbf{e}_2$ are sampled from $\mathcal{D}_{\mathbb{Z}^{n_1},r}$ respectively $\mathcal{D}_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}_{n_2}),\alpha q}$ for $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m}_2) \in \mathbb{Z}_q^{n_2/k}$ with $\alpha q \geq 2\sqrt{n} \geq 2 \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$. Set $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$. Output the ciphertext

$$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \text{ with } \mathbf{c}_1 = \mathbf{a}, \mathbf{c}_2 = \mathbf{as} + t\mathbf{e} + \mathbf{m}_1$$

$\mathsf{Dec}(sk, \mathbf{c})$: Compute the first message $\mathbf{m}_1 = \mathbf{c}_2 - \mathbf{c}_1\mathbf{s} \bmod t$. We then have $t\mathbf{e} = \mathbf{c}_2 - \mathbf{c}_1\mathbf{s} - \mathbf{m}_1$. If we divide t out, we obtain $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ and finally recover the second message $\mathbf{m}_2 = \mathsf{decode}\left((\mathbf{G}_{n_2} \cdot \mathbf{e}_2) \oplus H(\mathbf{e}_1)\right)$.

A quick view to this construction shows, that the error vector is efficiently obtained, because $\mathbf{c}_1\mathbf{s}$ has already been computed during decryption. And multiplication by a constant t^{-1} is performed very fast. We omit a formal proof here. Essentially, the same arguments of the proof from the original construction [BV11b] work here as well with the difference, that we reduce the security to the ring variant of the A-LWE problem. After this modification, we can additionally embed messages of size n_2 bits into the error term. However, we stress that the homomorphic property does not carry over to this additional slot of message. We basically obtain an encryption scheme working on message pairs (m_1, m_2) which is homomorphic only with respect to m_1. This scheme may be of independent interest for novel applications.

4.6 Embedding Approach Applied to [MP12]

All the features provided in the previous sections can be utilized in order to optimize the encryption scheme MP12 (see Appendix C.1). We particularly focus on the error term $(\bar{\mathbf{e}}, \mathbf{e}_1)$ that is recovered by the LWE inversion algorithm. The other scheme ingredients remain unchanged. One observes that the error term is unused and offers additional space for messages of size m bits as compared to nk bits in the original scheme. This allows us to increase the message throughput per ciphertext by a factor of $m/nk = c$, where $m = c \cdot nk$ and $c \in \mathbb{Q}_{\geq 2}$. The way RCCA-security is guaranteed resembles the message embedding approach. Hence, we can use a small part of the error term to lift up the scheme to RCCA (or CCA2). Due to $\alpha q, \alpha' q \geq 2\sqrt{n} \geq \eta_\epsilon(\Lambda^{\perp}_q(\mathbf{G}))$ the modified parts of the error term always follow the correct distributions. All these properties can be included at essentially the same efficiency. But one has to take care about the parameters in case the error term is augmented with authentication data as described in Appendix 4.4. Current state-of-the art signature schemes such as [DDLL13] are characterized by very small parameters and efficient instantiations. This allows one to realize the desired authentication subroutine without violating any conditions.

5 Parameter selection

We consider a statistical instantiation of the schemes with $\bar{m} = O(nk)$ such that the public key $\mathbf{A} = [\ \bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R}]$ is statistically close to uniform. The corresponding ring variant and the involving algorithms are presented in [MP12, BB13]. In both schemes from above (see Section 4.1 and Appendix C.1), we sample the trapdoor $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z},t}^{\bar{m} \times nk}$ for $t \in \omega(\sqrt{\log n})$. For instance, the parameter t is chosen to satisfy $t \geq 2\eta_\epsilon(\mathbb{Z})$ and $\bar{m} = nk + \omega(\sqrt{\log n})$, but other choices are also possible (see [MP12]). When setting the parameters, one has to take care of the maximum error length in order to correctly invert the underlying LWE instance in the decryption routine. By [MP12, Theorem 5.4], the maximum length of the error term is bounded by $\|\mathbf{e}\| < q/(4 \cdot \sqrt{s_1(\mathbf{R})^2 + 1})$ for $s_1(\mathbf{R}) \leq (\sqrt{\bar{m}} + \sqrt{nk}) \cdot t$, where $s_1(\mathbf{R})$ denotes the largest singular value of $\mathbf{R}$. As a consequence, selecting $\alpha \leq 1/(4 \cdot \sqrt{s_1(\mathbf{R})^2 + 1})$ for the error term allows for correct inversion, except with negligible probability. In case it is required to embed discrete Gaussians with larger parameters such as signatures from other schemes, one could apply the encryption scheme for high data load from Section 4.1. To this end m' is chosen in such a way that $\mathbf{e}'$ is completely replaced by the signature and one chooses arbitrary $\alpha q \geq 2\sqrt{n}$ such that the corresponding A-LWE instance is hard.

Below you find a candidate set of parameters to instantiate our proposed encryption schemes.

Parameter Selection	
n	e.g. $n \geq 256$ (cf. Table 1)
q	e.g. power of 2 with $q \geq 2^{19}$
k	$\lceil \log_2(q) \rceil$
m	$\bar{m} + nk = c \cdot nk,\ c \geq 2$
t	Parameter for matrix $\mathbf{R}$, e.g. $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z}^{\bar{m} \times nk},t}$
$s_1(\mathbf{R})$	$\approx 1/\sqrt{2\pi} \cdot (\sqrt{n \cdot k} + \sqrt{\bar{m}}) \cdot t$
r	$r \geq 2 \cdot \sqrt{\ln(2n(1 + \frac{1}{\epsilon}))/\pi}$
α	$1/(4 \cdot \sqrt{s_1(\mathbf{R})^2 + 1}) \geq \alpha > 2\sqrt{n}/q$

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, May 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, August 2010.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *29th Annual ACM Symposium on Theory of Computing*, pages 284–293. ACM Press, May 1997.

[ADL+08] Yuriy Arbitman, Gil Dogon, Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFTX: A proposal for the SHA-3 standard, 2008. In The First SHA-3 Candidate Conference.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108. ACM Press, May 1996.

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *STACS*, volume 3 of *LIPIcs*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

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

[BB13] Rachid El Bansarkhani and Johannes Buchmann. Improvement and efficient implementation of a lattice-based signature scheme. In Lange Tanja, Kristin Lauter, and Petr Lisonek, editors, *Selected Areas in Cryptography*, LNCS. Springer, 2013.

[BCHK07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584, 2013.

[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, August 2012.

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

[CHJ+02] Jean-Sébastien Coron, Helena Handschuh, Marc Joye, Pascal Paillier, David Pointcheval, and Christophe Tymen. GEM: A generic chosen-ciphertext secure encryption method. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 263–276. Springer, February 2002.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, May 2004.

[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, May 2010.

[CKN03] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 565–582. Springer, August 2003.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO (1)*, pages 40–56, 2013.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DMQ13] Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In Thomas Johansson and PhongQ. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 18–34. Springer Berlin Heidelberg, 2013.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, August 1987.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. ACM, 2009.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 105–111. Springer, August 1997.

[GH11] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 107–109. IEEE Computer Society Press, October 2011.

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547. Springer, September 2012.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008.

[Gro04] Jens Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 152–170. Springer, February 2004.

[HLM03] Jonathan Herzog, Moses Liskov, and Silvio Micali. Plaintext awareness via key registration. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 548–564. Springer, August 2003.

[JD12] Xiaodong Lin Jintai Ding. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688, 2012. `http://eprint.iacr.org/`.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, March 2006.

[Kle00] Philip N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA 2000*, pages 937–941. ACM, 2000.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 636–652. Springer, December 2009.

[LMPR08] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. Swifft: A modest proposal for FFT hashing. In *FSE*, volume 5086 of *LNCS*, pages 54–72. Springer, 2008.

[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, February 2011.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, May 2010.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In Thomas Johansson and PhongQ. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54. Springer Berlin Heidelberg, 2013.

[LV08] Benoît Libert and Damien Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In Ronald Cramer, editor, *PKC 2008: 11th International Conference on Theory and Practice of Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 360–379. Springer, March 2008.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, April 2012.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, April 2012.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. In *CRYPTO (1)*, pages 21–39, 2013.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381. IEEE Computer Society Press, October 2004.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 333–342. ACM Press, May / June 2009.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 80–97. Springer, August 2010.

[Pei14] Chris Peikert. Lattice cryptography for the internet. Cryptology ePrint Archive, Report 2014/070, 2014. `http://eprint.iacr.org/`.

[PR07] Manoj Prabhakaran and Mike Rosulek. Rerandomizable RCCA encryption. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 517–534. Springer, August 2007.

[PSNT06] Duong Phan, Reihaneh Safavi-Naini, and Dongvu Tonien. Generic construction of hybrid public key traitor tracing with full-public-traceability. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 264–275. Springer, July 2006.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, August 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 187–196. ACM Press, May 2008.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51:899–942, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, December 2009.

[XF07] Rui Xue and Dengguo Feng. Toward practical anonymous rerandomizable RCCA secure encryptions. In Sihan Qing, Hideki Imai, and Guilin Wang, editors, *ICICS 07: 9th International Conference on Information and Communication Security*, volume 4861 of *Lecture Notes in Computer Science*, pages 239–253. Springer, December 2007.

A Chosen Ciphertext Security and Variants

We recall the definitions of (replayable) chosen ciphertext security of encryption schemes. Let $\mathcal{E} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme and consider the following experiments for $\mathsf{atk} \in \{\mathsf{cca1}, \mathsf{cca2}, \mathsf{rcca}\}$:

Experiment $\mathbf{Exp}^{\mathrm{ind}-\mathsf{atk}}_{\mathcal{E},\mathcal{A}}(n)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^k)$
$(\mu_0, \mu_1) \leftarrow \mathcal{A}^{\mathsf{Dec}(\cdot)}(\mathrm{CHOOSE}, \mathsf{pk})$
$\mathbf{c}_b \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\mu_b)$ for $b \leftarrow_R \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_2(\cdot)}(\mathrm{GUESS}, \mathbf{c}_b)$
Output 1 iff
1. $b' = b$
2. $|\mu_0| = |\mu_1|$
3. $\mathbf{c}_b$ was not queried to $\mathcal{O}_2$

If $\mathcal{A}$ queries $\mathcal{O}_2(\mathbf{c})$, and
- if $\mathsf{atk} = \mathsf{cca1}$, then return $\perp$.
- if $\mathsf{atk} = \mathsf{cca2}$, then return $\mathsf{Dec}(\mathsf{sk}, \mathbf{c})$.
- if $\mathsf{atk} = \mathsf{rcca}$ and $\mathsf{Dec}(\mathsf{sk}, \mathbf{c}) \notin \{\mu_0, \mu_1\}$, then return $\mathsf{Dec}(\mathsf{sk}, \mathbf{c})$.
- Otherwise, return $\perp$.

Definition 6 (Chosen-ciphertext secure encryption). *Let $\mathcal{E} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme.*

CCA1 security. *We say $\mathcal{E}$ is secure against non-adaptive chosen-ciphertext attacks (CCA1) if we have*

$$\Pr[\mathbf{Exp}^{\mathsf{atk}-\mathsf{cca1}}_{\mathcal{E},\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n) \ .$$

CCA2 security. *We say $\mathcal{E}$ is secure against adaptively chosen-ciphertext attacks (CCA2) if we have*

$$\Pr[\mathbf{Exp}^{\mathsf{atk}-\mathsf{cca2}}_{\mathcal{E},\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n) \ .$$

RCCA security. *We say $\mathcal{E}$ is secure against replayable chosen-ciphertext attacks (RCCA) if we have*

$$\Pr[\mathbf{Exp}^{\mathsf{atk}-\mathsf{rcca}}_{\mathcal{E},\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n) \ .$$

for all polynomial-time algorithms $\mathcal{A}$.

There exists a strict hierarchy in the security notions. That is, CCA2 security implies RCCA security which in turn implies CCA1 security. All the above security notions are formulated following the indistinguishability approach. We note that alternatively one could define the security in a non-malleability approach yielding NM-CCA1, NM-CCA2, and NM-RCCA. Here, an adversary given an encryption of μ_b is essentially not able to output an encryption of μ_{1-b}. The non-malleability notions imply the indistinguishability counterparts. However, the other direction does not hold in general, for instance for CCA1 security. Moreover, the notion of non-malleable replayable CCA is strictly stronger than the indistinguishably notion of replayable CCA for polynomial message spaces [CKN03].

B Lattice-Related Algorithms

B.1 Sampling from $\Lambda^{\perp}(\mathbf{G})$

Let $k = \lceil \log q \rceil$. We recall the primitive matrix $\mathbf{G}_{nk} = \mathbf{I}_n \otimes \mathbf{g}^\top$ with $\mathbf{g}^\top = (1, 2, \ldots, 2^{k-1})$ from [MP12], which plays a major role in our constructions due to its simple structure and the efficiency when sampling short vectors from the sets $\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}_{nk})$ for any syndrome $\mathbf{v} \in \mathbb{Z}_q^n$. We denote by $\mathbf{G}_m$ the matrix $\mathbf{I}_{m/k} \otimes \mathbf{g}^\top$, otherwise we refer to $\mathbf{G} = \mathbf{G}_{nk}$ in case we omit the index. This matrix induces a lattice $\Lambda^{\perp}_q(\mathbf{G}_{nk}) = \{\mathbf{x} \in \mathbb{Z}^{nk} \mid \mathbf{G}_{nk}\mathbf{x} \equiv 0 \bmod q\}$ with generator matrix $\mathbf{S} = \mathbf{I}_n \otimes \mathbf{S}_k$, where

$$\mathbf{S}_k = \begin{bmatrix} 2 & & & 0 \\ -1 & 2 & & \\ & \ddots & \ddots & \\ 0 & & -1 & 2 \end{bmatrix} \in \mathbb{Z}_q^{k \times k}.$$

Partitions are represented by $\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}_{nk}) = \mathbf{x} + \Lambda^{\perp}_q(\mathbf{G}_{nk})$, where $\mathbf{x}$ denotes an arbitrary vector satisfying $\mathbf{G}\mathbf{x} = \mathbf{v} \bmod q$. The corresponding smoothing parameter is bounded by $\|\mathbf{S}\| \sqrt{\ln(2n(1+1/\epsilon))/\pi} \geq \eta_\epsilon(\Lambda^{\perp}_q(\mathbf{G}_{nk}))$. The sampling algorithm for any partition $\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G})$ with $\mathbf{v} \in \mathbb{Z}_q^n$ works as desecribed in Algorithm 1. This algorithm is applied on each component of $\mathbf{v}$ yielding a sample $\mathbf{x}$ distributed as $D_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}),r}$.

Algorithm 1: Algorithm Sample to sample from $\Lambda^{\perp}(G)$

Input: $\mathbf{G}, \mathbf{v}$
Output: $\mathbf{t} = (t_0, \ldots, t_{k-1})^T$

▷ *The algorithm initializes a_0 with an entry $v_i \in \mathbb{Z}_q$ of the syndrome $\mathbf{v}$ and outputs a vector $\mathbf{t} \in \Lambda^{\perp}_{v_i}(\mathbf{g}^\top)$ distributed as $D_{\Lambda^{\perp}_{v_i}(\mathbf{g}^\top),r}$.*

for $i = 0, \ldots, k-1$ **do**
  $t_i \leftarrow D_{2\mathbb{Z}+a_i,r}$
  $a_{i+1} = (a_i - t_i)/2$

B.2 LWE Inversion (Invert Algorithm)

We briefly describe the inversion algorithm for the trapdoor function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top$ as provided in [MP12], which represents the core decryption subroutine in the generic encryption scheme from Section 3.3. Specifically, it employs a trapdoor matrix $\mathbf{R}$ to solve LWE instances $\mathbf{b}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top \bmod q$ generated by a handcrafted random matrix of the form $\mathbf{A} = [\bar{\mathbf{A}} \mid \mathbf{H}\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}]$ for an invertible matrix $\mathbf{H}$ and the primitive matrix $\mathbf{G}$. Without the knowledge of the trapdoor it is hard to find a solution $\mathbf{s}$ and $\mathbf{e}$

for a given $\mathbf{b}$ such that $\mathbf{b} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ assuming the hardness of LWE. For $q = 2^k$, the inversion algorithm works as follows. Suppose we are given a vector $\mathbf{b}^\top$ corresponding to $\mathbf{A}$. Applying $\mathbf{R}$, we obtain

$$\mathbf{b}^\top \cdot \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{s}^\top \mathbf{H}\mathbf{G} + \mathbf{e}^\top \cdot \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{s}'^\top \mathbf{G} + \mathbf{e}'^\top .$$

Now, for every scalar s'_i in $\mathbf{s}'$, we have to solve the equation $\mathbf{b}^\top_{(i)} = s'_i \cdot \mathbf{g}^\top + \mathbf{e}^\top_{(i)}$, where $\mathbf{e}'_{(i)} = (e'_{i\cdot k}, \ldots, e'_{i\cdot k+(k-1)})$. Considering the binary expansion $s'_i = \sum_{j=0}^{k-1} c_j \cdot 2^j$ one starts from the last entry in $\mathbf{b}^\top_{(i)}$, which is $b_{i\cdot k+(k-1)} = 2^{k-1} \cdot c_0 + e_{i\cdot k+(k-1)} \mod q = (q/2)c_0 + e_{i\cdot k+(k-1)}$. One sets $c_0 = 0$ in case $b_{i\cdot k+(k-1)} \in [-\frac{q}{4}, \frac{q}{4})$, else $c_0 = 1$. Subtracting $2^{k-1} \cdot c_{k-1}$ from s'_i one recovers c_1 by the same testing procedure etc. This is successively applied on all entries s'_i, which subsequently returns $\mathbf{s}'$. Finally, one recovers $\mathbf{e}^\top = \mathbf{b}^\top - \mathbf{s}'^\top \mathbf{G}$ and $\mathbf{s} = \mathbf{s}'\mathbf{H}^{-1}$.

B.3 Message Embedding for Uniform Random Errors

In this subsection we briefly explain how to embed messages into error vectors that are uniformly distributed over a set $[0, 2^l - 1]^m$ with $l \geq 2$. This set can subsequently be transformed into a desired representation. In fact, by slightly modifying the strategy from Section 3.1 one obtains the required distributions. Therefore, let again $\mathbf{G} = \mathbf{I}_{m/l} \otimes \mathbf{g}^\top$ and $p = 2^l$ with $m = l \cdot r$ and $\mathbf{g}^\top = (1, 2, \ldots, 2^{l-1})$ as shown in Appendix B.1. We know from Appendix D.1 that $\mathbb{Z}^m = \dot{\bigcup}_{\mathbf{v} \in \mathbb{Z}_q^n} \Lambda^\perp_{\mathbf{v}}(\mathbf{A})$ holds for any full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and so for $\mathbf{G}$. It then follows $\mathcal{U}(\mathbb{Z}^m) \sim \mathcal{U}(\Lambda^\perp_{\mathbf{v}}(\mathbf{G}))$ for $\mathbf{v} \leftarrow_R \mathbb{Z}_q^n$, where $\mathcal{U}(B)$ denotes the uniform distribution over the set B. Our aim is to let $\mathbf{v}$ be statistically (resp. computationally) indistinguishable from uniform similar to Lemma 5 and Lemma 6. We slightly change the algorithm from Section B.1 and explain how to sample a vector uniformly at random from the set $[0, 2^l - 1]^m \cap \Lambda^\perp_{\mathbf{v}}(\mathbf{G})$ for any fixed choice of $\mathbf{v}$.

Algorithm 2: Algorithm SampleU to sample from $\Lambda^\perp(G)$

Input: $\mathbf{G}, \mathbf{v}$
Output: $\mathbf{t} = (t_0, \ldots, t_{l-1})^T$

▷ *The algorithm initializes a_0 with an entry $v_i \in \mathbb{Z}_q$ of the syndrome $\mathbf{v}$ and outputs a vector $\mathbf{t} \in \Lambda^\perp_{v_i}(\mathbf{g}^\top)$ distributed as $D_{\Lambda^\perp_{v_i}(\mathbf{g}^\top), r}$.*

for $i = 0, \ldots, l-1$ **do**
  $t_i \leftarrow_R \{2\mathbb{Z} + a_i\} \cap [0, 2^l - 1]$
  $a_{i+1} = (a_i - t_i)/2$

In each step t_i is chosen from a set containing 2^{l-1} odd or even elements dependent on a_i. Thus, the algorithm outputs uniform random elements from $[0, 2^l - 1]^m \cap \Lambda^\perp_{\mathbf{v}}(\mathbf{G})$. As a consequence, we can prove the statistical or computational indistinguishability in essentially the same way as in Lemma 5 and 6.

Following this approach, we sample the error vector as follows. Let $\mathbf{m} \in \{0,1\}^m$ be the message and $\mathbf{v} = \mathsf{encode}(\mathbf{m} \oplus H(\mathbf{s})) \in \mathbb{Z}_q^n$. Then, we obtain a vector that is computationally indistinguishable from uniform. Subsequently, we sample $\mathbf{e} \leftarrow_R [0, 2^l - 1]^m \cap \Lambda^\perp_{\mathbf{v}}(\mathbf{G})$ using Algorithm 2. Here, we have assumed that vector $\mathbf{s}$ is freshly sampled uniformly from $\mathbb{Z}_q^n$ in each sample. If this is not the case, one takes also some part of the error term as input to the function H, in order to provide sufficient entropy.

C Lattice-Based Cryptosystems

C.1 CCA1-Secure Encryption Scheme by Micciancio and Peikert

Recently, Micciancio and Peikert [MP12] introduced new methods to derive trapdoors in cryptographic lattices with higher quality compared with previous constructions [Bab86, Kle00, AP09, Pei10]. One application is a new CCA1-secure encryption scheme, denoted by $\mathsf{MP12}$, whose security is based on the hardness of $\mathrm{LWE}_{n,\alpha q}$ for $\alpha q \geq 2\sqrt{n}$. The construction makes use of the gadget matrix $\mathbf{G}$ with $k = O(\log q) = O(\log n)$. Further, let $\bar{m} = O(nk)$ and $\mathcal{D} = D_{\mathbb{Z}^{\bar{m}\times nk},\omega(\sqrt{\log n})}$. The public key and ciphertext size are determined by $m = \bar{m} + nk$. The error rate α for LWE is given by $1/\alpha = O(nk)\cdot\omega(\sqrt{\log n})$.

For realizing the tag-based approach, we recall the construction from [MP12]. Let $\mathcal{R} = \mathbb{Z}_q[x]/(f(x))$ be a ring as constructed in [MP12], where $f(x)$ denotes a monic irreducible polynomial of degree n. Furthermore, let $h : \mathcal{R} \to \mathbb{Z}_q^{n\times n}$ be an injective ring homomorphism mapping elements $a \in \mathcal{R}$ to the matrix $h(a)$. By $\mathcal{U} = \{u_1, \ldots, u_\ell\}$ we denote a large set with "unit differences" property. That is, for any two ring elements a_i and $a_j \in \mathcal{R}^*$ with $i \neq j$ we have $a_i - a_j \in \mathcal{R}^*$ and $h(a_i - a_j) = h(a_i) - h(a_j)$ is invertible. Adopting the notation from [MP12] the map encode allows to bijectively switch between $\{0,1\}^{nk}$ and the cosets of $\Lambda/2\Lambda$ with encode^{-1} being its inverse, where $\Lambda = \Lambda(\mathbf{G}^t)$.

The encryption scheme $\mathsf{MP12} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ works as follows:

$\mathsf{KGen}(1^n)$: Upon input the security parameter n, choose $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n\times m'}$ and $\mathbf{R} \leftarrow_R \mathcal{D}$. Let $\mathbf{A}_1 = -\bar{\mathbf{A}}\mathbf{R} \bmod q$. The public key is $\mathsf{pk} = \mathbf{A} = [\bar{\mathbf{A}}|\mathbf{A}_1] \in \mathbb{Z}_q^{n\times m}$ while the secret key is determined by $\mathsf{sk} = \mathbf{R}$.

$\mathsf{Enc}(\mathsf{pk}, \mu)$: Upon input public key $\mathsf{pk} = [\bar{\mathbf{A}}|\mathbf{A}_1]$ and message $\mu \in \{0,1\}^{nk}$, do the following:

- Set $\mathbf{A}_u = [\bar{\mathbf{A}}|\mathbf{A}_1 + h(u)\mathbf{G}]$ where $u \leftarrow_R \mathcal{U}$ is nonzero.
- Choose $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{e}_1 \leftarrow_R \mathcal{D}_{\mathbb{Z},\alpha q}^{m'}$, and $\mathbf{e}_2 \leftarrow_R \mathcal{D}_{\mathbb{Z},s}^{nk}$ where $s^2 = (\|\mathbf{e}_1\|^2 + m'(\alpha q)^2)\cdot\omega(\sqrt{\log n})^2$.
- Set $\mathbf{b}^t = 2(\mathbf{s}^t\mathbf{A}_u \bmod q) + \mathbf{e}^t + (0, \mathsf{encode}(\mu))^t \bmod 2q$, where $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2) \in \mathbb{Z}^m$ and $\mathbf{0}$ is of dimension m'.

The ciphertext $\mathbf{c}$ consists of $\mathbf{c} = (u, \mathbf{b}) \in U \times \mathbb{Z}_{2q}^m$.

$\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \mathbf{c})$: Upon input public key $\mathsf{pk} = [\bar{\mathbf{A}}|\mathbf{A}_1]$, secret key $\mathsf{sk} = \mathbf{R}$, and ciphertext $\mathbf{c} = (u, \mathbf{b})$, do the following:

- Set $\mathbf{A}_u = [\bar{\mathbf{A}}|h(u)\mathbf{G} - \mathbf{A_1}\mathbf{R}]$.
- Call $\mathsf{Invert}(\mathbf{R}, \mathbf{A}_u, b \bmod q)$ from Section B.2 and obtain $\mathbf{z} \in \mathbb{Z}_q^n$ and $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2) \in \mathbb{Z}^{m'} \times \mathbb{Z}^{nk}$ for which $\mathbf{b}^t = \mathbf{z}\mathbf{A}_u + \mathbf{e}^t \bmod q$.
- Let $\mathbf{v} = \mathbf{b} - \mathbf{e} \bmod 2q$, parsed as $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_{2q}^{m'} \times \mathbb{Z}_{2q}^{nk}$.

If none of the following conditions is true, output message $\mu = \mathsf{encode}^{-1}(\mathbf{v}^t \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix} \bmod 2q) \in \mathbb{Z}^{nk}$.

Otherwise, output $\perp$.

1. $\mathbf{c}$ does not parse or $u = 0$,
2. the function Invert failed for any reason,
3. $\|\mathbf{e}_1\| \geq \alpha q\sqrt{m'}$ or $\|\mathbf{e}_2\| \geq \alpha q\sqrt{2m'nk}\cdot\omega(\sqrt{\log n})$,
4. $\mathbf{v}_1 \notin 2\Lambda(\bar{\mathbf{A}}^t)$, or
5. the function $\mathsf{encode}^{-1}(\cdot)$ does not exist.

C.2 Lattice-based Signatures by Ducas et al.

Here, we recall the lattice-based signature scheme by Ducas et al. [DDLL13] which qualifies as the state-of-the-art in terms of lattice signatures. It follows the Fiat-Shamir paradigm [FS87], and its security is based on SIS problem and proven in the random oracle model. The idea is as follows. Let D_σ^m define a discrete Gaussian distribution for standard deviation $\sigma \in \mathbb{R}$. To sign a message μ, first sample a short vector $\mathbf{y} \in D_\sigma^m$ and set $c = H(\mathbf{A}\mathbf{y} \bmod 2q, \mu)$ where $\mathbf{A} \in \mathbb{Z}_{2q}^{n\times m}$ is the public key. Now, choose a random bit $b \in \{0,1\}$. The signature consists of $(\mathbf{c}, \mathbf{z} = \mathbf{y} + (-1)^b\mathbf{S}\mathbf{c})$ where $\mathbf{S} \in \mathbb{Z}_{2q}^{n\times m}$ is a secret key, which

satisfies $\mathbf{AS} = q\mathbf{I}_n (\mathrm{mod}\ 2q)$. However, the security of this scheme holds only, if vector $\mathbf{z}$ output by the signature algorithm is distributed according to D_σ^m. This is achieved by rejection sampling, i.e., $(\mathbf{c}, \mathbf{z})$ is output only with probability $1/(M \exp(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}) \cosh(-\frac{\langle \mathbf{z}, \mathbf{Sc} \rangle}{\sigma^2}))$.

D Missing Proofs

D.1 Proofs for Message Embedding

Proof (of Lemma 5). Consider the statistical distance between $\mathcal{D}_{\mathbb{Z}^m,r}$ and $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$, where $\mathbf{v} \in \mathbb{Z}_q^n$ is chosen at random. Since $\mathbf{A}$ is a full-rank matrix, we have $\mathbb{Z}^m = \dot{\bigcup_{\mathbf{b} \in \mathbb{Z}_q^n}} \Lambda^\perp_{\mathbf{b}}(\mathbf{A})$ and $\rho_r(\mathbb{Z}^m) = \sum_{\mathbf{b} \in \mathbb{Z}_q^n} \rho_r(\Lambda^\perp_{\mathbf{b}}(\mathbf{A})) \in [\frac{1-\epsilon}{1+\epsilon}, 1] \cdot q^n \cdot \rho_r(\Lambda^\perp_q(\mathbf{A}))$. In the latter distribution $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ the process of sampling $\mathbf{z} \in \mathbb{Z}^m$ can be reduced to the tasks of selecting the correct partition $\Lambda^\perp_{\mathbf{v}}(\mathbf{A})$ with probability $\frac{1}{q^n}$ and subsequently sampling $\mathbf{z}$ from $\Lambda^\perp_{\mathbf{v}}(\mathbf{A})$ with probability $\frac{\rho_r(\mathbf{z})}{\rho_r(\Lambda^\perp_{\mathbf{Az}}(A))}$. Following this, $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ outputs a sample $\mathbf{z}$ with probability $P[\mathbf{X} = \mathbf{z}] = \frac{1}{q^n} \cdot \frac{\rho_r(\mathbf{z})}{\rho_r(\Lambda^\perp_{\mathbf{Az}}(A))}$.

$$
\begin{aligned}
\Delta(\mathcal{D}_{\mathbb{Z}^m,r}, \mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}) &= \sum_{\mathbf{z} \in \mathbb{Z}^m} \left| \frac{\rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^m)} - \frac{1}{q^n} \cdot \frac{\rho_r(\mathbf{z})}{\rho_r(\Lambda^\perp_{\mathbf{Az}}(A))} \right| \\
&\overset{Lemma\ 2}{\in} \sum_{\mathbf{z} \in \mathbb{Z}^m} \left| \frac{\rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^m)} - \frac{\rho_r(\mathbf{z})}{q^n \cdot [\frac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_r(\Lambda^\perp_q(\mathbf{A}))} \right| \\
&\overset{Lemma\ 2}{\in} \sum_{\mathbf{z} \in \mathbb{Z}^m} \left| \frac{\rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^m)} - \frac{\rho_r(\mathbf{z})}{[\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}] \cdot \sum_{\mathbf{b} \in \mathbb{Z}_q^n} \rho_r(\Lambda^\perp_{\mathbf{b}}(\mathbf{A}))} \right| \\
&= \sum_{\mathbf{z} \in \mathbb{Z}^m} \left| \frac{\rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^m)} - \frac{[\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}] \cdot \rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^m)} \right| \\
&\in [0, \frac{2\epsilon}{1-\epsilon}] \cdot \sum_{\mathbf{z} \in \mathbb{Z}^m} \left| \frac{\rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^m)} \right| \\
&\leq \frac{2\epsilon}{1-\epsilon}
\end{aligned}
$$

□

Here you find the proof of Lemma 6 for the computational counterpart.

Proof (of Lemma 6). Let $\mathbf{v}' \sim \mathcal{U}(\mathbb{Z}_q^l)$ be a vector chosen at random. By contradiction, we assume that $\mathbf{e} \sim \mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ is distinguishable from $\mathbf{e}' \sim \mathcal{D}_{\Lambda^\perp_{\mathbf{v}'}(\mathbf{A}),r} \overset{Lemma\ 5}{\approx_s} \mathcal{D}_{\mathbb{Z}^m,r}$ in polynomial time for the given parameters and $\mathbf{v}$ chosen as above. Then, $\mathbf{v}$ is computationally distinguishable from $\mathbf{v}'$ by Lemma 1 with $M(\mathbf{v}_i) = \mathcal{D}_{\Lambda^\perp_{\mathbf{v}_i}(\mathbf{A}),r}$. Hence, we have a contradiction. Therefore, the distribution $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ is computationally indistinguishable from $\mathcal{D}_{\mathbb{Z}^m,r}$. □

D.2 Proof of Theorem 1

Proof (of Theorem 1). We need to show that samples from $L^{\mathsf{LWE}}_{n,m,\alpha,q}$ are indistinguishable from $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha,q}(\mathbf{m})$ if we assume that the search $\mathsf{LWE}_{n,m,\alpha,q}$ problem is hard to solve in polynomial time and tuples $(\mathbf{s}, (\mathbf{e}_1)_i)$

for each sample i have sufficient entropy . That is,

$$L^{\mathsf{LWE}}_{n,m,\alpha,q} \approx_c L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha,q}(\mathbf{m})$$

for arbitrary $\mathbf{m} \in \{0,1\}^{m_2}$.

We consider a series of intermediate hybrid experiments. In the first hybrid, we modify the A-LWE samples in such a way that we replace $H(\mathbf{s}, \mathbf{e}_1)$ with a uniformly sampled value $\mathbf{u}$. Here, we use the fact, that $\mathbb{H}_\infty(\mathbf{s}, \mathbf{e}_1) > \lambda$. Consequently, $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}, \mathbf{e}_1) \oplus \mathbf{m})$ becomes uniformly distributed. The next hybrid replaces $\mathbf{e}_2$ by value $\mathbf{e}^*_2$ which is sampled according to $\mathcal{D}_{\mathbb{Z}^{m_2},r}$. The final distribution is identically distributed as the original LWE. In the following we describe the hybrids more formally.

Hybrid$_1$. In the first hybrid, in each A-LWE sample we replace the value $H(\mathbf{s}, \mathbf{e}_1)$ by a uniformly sampled value $\mathbf{u} \in \{0,1\}^{m_2}$. We argue that a (polynomial-time) distinguisher notices the difference only if it queries the random oracle on input $(\mathbf{s}, \mathbf{e}_1)$. Otherwise, if $(\mathbf{s}, \mathbf{e}_1)$ has not been queried before, the distribution of $H(\mathbf{s}, \mathbf{e}_1)$ is statistically close to the uniform distribution in $\{0,1\}^{m_2}$ due to the property of a random oracle and given sufficient entropy. This holds, in particular, if many samples are given to the distinguisher and all $H(\mathbf{s}, (\mathbf{e}_1)_i)$ have been replaced because by assumption we have that every pair $(\mathbf{s}, (\mathbf{e}_1)_i)$ in sample i has fresh entropy, more than λ.
We comment on a distinguisher which queries the random oracle at a certain point on $(s, \mathbf{e}_1)$ below in the proof, and assume for now, that no such distinguisher exists.

Hybrid$_2$. In the next hybrid, we replace the error term $\mathbf{e}_2$ by value $\mathbf{e}*_2$ which is sampled according to $\mathcal{D}_{\mathbb{Z}^{m_2},r}$. Note that A-LWE samples from **Hybrid$_1$** satisfy that $\mathbf{v} = \mathsf{encode}(\mathbf{u} \oplus \mathbf{m})$ is uniformly distributed since $\mathbf{u}$ is uniformly picked (even if $\mathbf{m}$ is chosen by the distinguisher). Now, Lemma 6 implies that $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ is computationally indistinguishable from $\mathcal{D}_{\mathbb{Z}^{m_2},r}$ for $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$. For this reason, replacing $\mathbf{e}_2$, which is distributed according to $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$, by vector $\mathbf{e}^*_2$ is unnoticeable to a distinguisher.

We argue that A-LWE samples from hybrid **Hybrid$_2$** are indistinguishable from LWE samples. This follows from the fact that the error term in A-LWE is now identically distributed as LWE which is the only difference between A-LWE and LWE samples. We still need to argue that it is very unlikely that a distinguisher queries the random oracle H on input $(\mathbf{s}, \mathbf{e}_1)$ for some $\mathbf{e}_1$ used in an A-LWE sample.

Suppose that there exists an algorithm $\mathcal{A}$ which distinguishes in polynomial time original A-LWE samples from A-LWE samples from **Hybrid$_1$** with non-negligible probability. We then construct an adversary $\mathcal{A}_{LWE}$ with black-box access to algorithm $\mathcal{A}$ that solves the $\mathsf{search\ LWE}_{n,m,\alpha,q}$ problem in polynomial time with non-negligible probability. This contradicts the theorem assumption that $\mathsf{search\ LWE}_{n,m,\alpha,q}$ is hard.

Adversary $\mathcal{A}_{LWE}$ is given samples from $L^{\mathsf{LWE}}_{n,m,\alpha,q}$ and is asked to find the secret vector $\mathbf{s}$. Let us denote the query $(s, \mathbf{e}_1)$ on H made by $\mathcal{A}$ by q^* where q^* is polynomially bounded by the security parameter. Whenever algorithm $\mathcal{A}$ asks for new samples, $\mathcal{A}_{LWE}$ asks for samples in her challenge and forwards them to $\mathcal{A}$. That is, $\mathcal{A}$ obtains samples from $L^{\mathsf{LWE}}_{n,m,\alpha,q}$ instead of either version of $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha,q}(\mathbf{m})$. We have already shown via hybrids that $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha,q}(\mathbf{m})$ is indistinguishable to $L^{\mathsf{LWE}}_{n,m,\alpha,q}$ if $(s, \mathbf{e}_1)$ was not sent to oracle H. This means that before $\mathcal{A}$ makes query q^* to H, those samples are indistinguishable. As a result, $\mathcal{A}$ must query H on input $(s, \mathbf{e}_1)$ even if given LWE samples. We stress that after returning the hash value of $(s, \mathbf{e}_1)$ to $\mathcal{A}$ it may be noticing that $\mathcal{A}_{LWE}$ has tricked her. However, eavesdropping the input to oracle H suffices to $\mathcal{A}_{LWE}$ to break her $\mathsf{search\ LWE}_{n,m,\alpha,q}$ problem independently whether $\mathcal{A}$ aborts at this time. Hence, if $\mathcal{A}$ queries H on input $(s, \mathbf{e}_1)$ with non-negligible probability, so does $\mathcal{A}_{LWE}$ solve the $\mathsf{search\ LWE}_{n,m,\alpha,q}$ problem with the very same probability. By assumption there does not exist such a successful algorithm.

We conclude that the step from the original A-LWE samples to **Hybrid$_1$** will be unnoticeable to a distinguisher if $\mathsf{search\ LWE}_{n,m,\alpha,q}$ is hard, and both distributions $L^{\mathsf{LWE}}_{n,m,\alpha,q}$ and $L^{\mathsf{A\text{-}LWE}}_{n,m_1,m_2,\alpha,q}(\mathbf{m})$ are computationally indistinguishable.

□

D.3 Proof of Theorem 5

Proof (of Theorem 5). First, we have to show that the error term $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ has the proper distribution. Since the coset selection for $\mathbf{e}_2$ is based on the entropy of $\mathbf{s}$ and hence the randomness of $H(\mathbf{s})$, the distribution of $\mathbf{e}_2$ is negligible close to $\mathcal{D}_{\mathbb{Z}^{m_1},\alpha q}$ according to Lemma 5. The same argument also holds for $\mathbf{e}_1$, which uses $\mathbf{e}_2$ as the source for entropy, when computing the random coset $H(\mathbf{s}, \mathbf{e}_2, u)$. Thus, it follows that the vector $\mathbf{b}$ is then distributed from a distribution which is indistinguishable from the $L^{\mathsf{A\text{-}LWE}}_{n,0,m,\alpha,q}(\mathbf{m})$ distribution.

Hence, the scheme above satisfies CCA1 security following the reasonings in Section 4.1. Suppose $(u^*, \mathbf{b}^*)$ is the challenge ciphertext. We will show that any decryption oracle query cannot further help the attacker in guessing the correct message according to the security model $\mathbf{Exp}^{\mathrm{ind-rcca}}_{\mathcal{E},\mathcal{A}}(n)$ (see Appendix A).

If the attacker queries $(u, \mathbf{b}^*)$ to the decryption oracle for u other than u^*, the oracle will respond with $\perp$, since $\mathbf{G}_{m_1}\mathbf{e}_1^* = H(\mathbf{s}^*, \mathbf{e}_2^*, u^*)$ does not hold anymore. In case $\mathbf{b} \neq \mathbf{b}^*$ for arbitrary selected u, we differentiate 3 cases:

1. $\mathbf{b}_1 = \mathbf{b}_1^*$ and $\mathbf{b}_2 \neq \mathbf{b}_2^*$, then $\mathsf{Dec}(\cdot)$ outputs $\perp$ since $\mathbf{G}_{m_1}\mathbf{e}_1^* = H(\mathbf{s}^*, \mathbf{e}_2^*, u)$, but either $\mathbf{s} \neq \mathbf{s}^*$ or $\mathbf{e}_2 \neq \mathbf{e}_2^*$.
2. $\mathbf{b}_1 \neq \mathbf{b}_1^*$ and $\mathbf{b}_2 \neq \mathbf{b}_2^*$, then $\mathbf{s}$, $\mathbf{e}$ or both must have been changed. In case $\mathbf{s}$ changed and $\mathbf{e} = \mathbf{e}^*$, $\mathsf{Dec}(\cdot)$ outputs $\perp$. If $\mathbf{e}$ altered while $\mathbf{s} = \mathbf{s}^*$ is still the same, the adversary must have known $\mathbf{s}^*$ in order to obtain $H(\mathbf{s}^*, \mathbf{e}_2, u)$ – and hence a preimage for a random oracle which we can use to solve the underlying A-LWE problem. Finally, if both values $\mathbf{s} \neq \mathbf{s}^*$ and $\mathbf{e} \neq \mathbf{e}^*$ and $\mathsf{Dec}(\cdot)$ outputs a message other than $\perp$, the attacker must have selected $\mathbf{s}$ and $\mathbf{e}$ or the message such that $\mathbf{G}_{m_1}\mathbf{e}_1 = H(\mathbf{s}, \mathbf{e}_2, u)$ is satisfied.
3. $\mathbf{b}_1 \neq \mathbf{b}_1^*$ and $\mathbf{b}_2 = \mathbf{b}_2^*$, then the decryption oracle outputs replay in case $\mathbf{e}_2 - \mathbf{e}_2^* \in \Lambda_q^{\perp}(\mathbf{G}_{m_1})$ with $u = u^*$ and short $\mathbf{e}_2 - \mathbf{e}_2^*$ indicating that $\mathbf{b}$ and $\mathbf{b}^*$ decrypt to the same plaintext. Otherwise $\mathsf{Dec}(\cdot)$ outputs $\perp$.

In all cases, the attacker learns nothing about the message concealed in $\mathbf{b}^*$. Thus we have an IND-RCCA secure encryption scheme, which is equivalent to NM-RCCA due to the large message space. The last case implies a publicly detectable RCCA (pd-RCCA) scheme, where an arbitrary party is able to detect modified ciphertexts that decrypt to the same plaintext. Based on the relation CCA2 $\Rightarrow$ pd-RCCA $\Rightarrow$ sd-RCCA $\Rightarrow$ RCCA [CKN03], we even have a stronger security guaranty than plain RCCA. $\square$