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. 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 one 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.

Key words: lattice-based cryptography, encryption, computational assumption

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 [1–3], 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 [4, 5] and signature schemes [6–10], while the latter mainly serves as a hard underlying problem for the security of primitives from Cryptomania, such as key exchange [11–13] and oblivious transfer [14]. 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 in $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$. 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 [1] and Micciancio and Regev [3], 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 [15]. 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 $\tilde{O}(n/\alpha)$-SIVP and GapSVP in n-dimensional worst-case lattices. Later, Peikert [16] and Brakerski et al. [17] gave a classical reduction from GapSVP to LWE. In [18], Döttling and Müller-Quade proved the hardness of LWE for uniformly distributed errors. Subsequently, Micciancio and Peikert [19] show that LWE remains hard even for binary errors.

Ever since the breakthrough work of Regev [15] lattice-based cryptography emerged and novel encryption schemes have been built upon LWE such as fully homomorphic encryption [20–24] and identity-based encryption [6,25–27] besides of CPA-secure [14,15,28–30] and CCA-secure encryption schemes [7,13,16,31].

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.

1. 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.
2. 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 en-

cryption schemes one may inject the message into the error term without necessarily changing the target distributions.

3. 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 [7,32] provide mechanisms to recover the error term. The most promising trapdoor construction is proposed by Micciancio and Peikert [7].

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. 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 [7], 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 acknowledgment 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.

Conceptually, the strategy of injecting messages into the error term 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 [33], 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 [6] in combination with the trapdoor construction [7] and the practical signature schemes presented in [8,10], 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 [7]. 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 [34], which does not rely on the trapdoor construction from [7].

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 indistinguishable from 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 [7], 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 $\mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),r}$. Vectors $\mathbf{e} \in \mathbb{Z}_q^m$ distributed according to $\mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),r}$ satisfy the equation $\mathbf{G}\mathbf{e} \equiv \mathbf{v} \bmod q$ for arbitrary $\mathbf{v} \in \mathbb{Z}_q^{m/k}$. Let $H : \{0,1\}^* \rightarrow \{0,1\}^m$ 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 $\mathbf{m} = \mathsf{decode}(\mathbf{G}\mathbf{e} \mod q) \oplus H(\mathsf{seed})$. 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 decision A-LWE and hence 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 [7]. Note that while some encryption schemes like [7, 32] 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 [7, 26, 29, 31], 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 [7], 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 $\mathbf{v} = \mathsf{encode}(H(\mathbf{s}) \oplus \mathbf{m})$.[1]

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

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 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 $\mathbf{A}_u = [\ \bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ] \in \mathbb{Z}^{n \times m}$ with trapdoor $[\ \mathbf{R}^\top\ \mathbf{I}\ ]^\top \in \mathbb{Z}^{nk \times m}$ to $\mathbf{A}_u^{ext} = [\ \mathbf{A}' \mid \bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R} - h(u)\mathbf{G}_{nk}\ ] \in \mathbb{Z}^{n \times (m'+m)}$ with trapdoor $[\ \mathbf{0}\ \ \mathbf{R}^\top\ \mathbf{I}\ ]^\top \in \mathbb{Z}^{nk \times (m'+m)}$. When inverting a ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}^{m'+m}$ – that is, an A-LWE instance – only the lower part of the ciphertext $\mathbf{c}_2$ is required to recover $\mathbf{s}$ and $\mathbf{e}$. This idea does not seem to carry over to the construction of [7]. In fact, their message are fixed to nk bits and extending the public key as above cannot be applied to their scheme.

Nonetheless, we show that message injection into the error term can directly enhance the CCA-secure scheme in [7] yielding a decrease of the message expansion factor. As a result, one obtains a scheme that follows the one-time-pad approach while encapsulating further messages in the error vector. Put it differently, with message embedding one could choose smaller parameters for the scheme in [7] when encrypting the same message length. In terms of security the original proof in [7] gets through without any major modifications. Table 1 gives an overview of parameters and the corresponding sizes for various lattice-based encryption schemes where we, for simplicity, fix the ciphertext size. Note that we have $c \in \mathbb{Q}_{\geq 2}$ for a matrix statistically close to uniform, and consequently the message throughput in our scheme is at least twice as the one from [7]. The ring setting, however, allows for smaller key sizes and more efficient implementations. In Table 1 we mainly focus on the most efficient ones including the CPA-secure encryption scheme from [29]. Due to space reasons, Table 1 does not include the less efficient schemes from [16, 26, 31], which are characterized by large public keys or small LWE error-rates beside of high message expansion factors. For instance, in [31] the LWE error rate $\alpha = \tilde{O}(1/n^4)$ is quite small (yielding to an easier LWE instance) with public keys of size $\tilde{O}(n^2)$ bits. In [16], Peikert improved the LWE error rate to $\alpha = \tilde{O}(1/n)$ but with the cost of an increased public key of size $\tilde{O}(n^3)$. The CCA-secure encryption scheme [26] provides a trade-off of the previous proposals with an LWE error rate of $\tilde{O}(1/n^2)$ and public key size of $\tilde{O}(n^2)$ bits.

Replayable Chosen-Ciphertext Secure Encryption. The notion of replayable CCA-security, which constitutes a relaxed version of CCA2-security, was firstly introduced by Canetti et al. [33] and addresses the ability of an adversary to replay ciphertexts that decrypt to the same message. An RCCA-secure encryption scheme

$m = c \cdot nk$, $k = \log q$	CCA1 **[7]**	CCA1 Constr. **4.1**	CCA1 Constr. **4.1** + **[7]**	CPA **[29]**
Ciphertext size	$m \cdot k$	$m \cdot k$	$m \cdot k$	$m \cdot k$
Message size	nk	$c \cdot nk$	$(c+1) \cdot nk$	$cnk - n$
Message Exp.	$c \cdot k$	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	$n \cdot m$	$n \cdot m$	$n \cdot m$	$n \cdot m$

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

detects modifications carried out on the ciphertext that alter the message. Valid encryptions of the same ciphertexts, however, are allowed. Canetti et al. have shown that RCCA is sufficient for most practical applications. There exist a series of RCCA-secure encryption schemes [35–39]. However, to our knowledge, we are the first realizing an RCCA-secure encryption scheme based on lattice problems, and hence relying on the worst-case hardness of lattice problems. We show that RCCA security comes essentially through our message embedding technique with only minor modifications. Our construction resembles GPV signatures generated for the public matrix $\mathbf{G}$. Just as for standard GPV signatures, it is required to hash all sensible (random) variables such as the tag u, the secret $\mathbf{s}$ and the lower part of the error term $\mathbf{e}_2$ containing the message to $\mathbf{v} = H(u, \mathbf{s}, \mathbf{e}_2)$ using a random oracle H. Subsequently, we sample a preimage $\mathbf{e}_1 \leftarrow \mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G})}$ 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 $\Lambda_q^{\perp}(\mathbf{G})$ 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 [33]. 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 [40–43] that ensure CCA2-security. For instance, one can use strongly unforgeable one-time signature schemes [40], 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 [33] 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 [7, 8, 10].

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 approximately $m \cdot \log(\alpha q) = O(m \log n)$ 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. For instance, we can embed signatures of size $c \log(\alpha q) nk$ bits as compared to nk bits following [7]. 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 $\mathbf{A}_u^{ext}$ 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 [7]. 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 [7], 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 [34], 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 $\mathbf{b} = \mathbf{a}\mathbf{s} + t\mathbf{e} \in \mathcal{R}_q = \mathbb{Z}[X]/\left\langle f(X) \right\rangle$ 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.

Due to space limitations, we detail the application of our technique to obtain a CCA-secure encryption scheme and refer the reader to our full version [44] for the further aforementioned applications.

2 Preliminaries

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., $\mathbf{A}$).

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

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.

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 1 (LWE Distribution). *Let n, m, q be integers and $\chi_e = \mathcal{D}_{\mathbb{Z}^m,\alpha q}$ be the discrete Gaussian distribution over $\mathbb{Z}^m$. 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 2 (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 $\mathsf{decision}\ \mathsf{LWE}_{n,m,\alpha q}$ (resp. $\mathsf{search}\ \mathsf{LWE}_{n,m,\alpha q}$) is hard if all polynomial time algorithm solves $\mathsf{decision}\ \mathsf{LWE}_{n,m,\alpha q}$ (resp. $\mathsf{search}\ \mathsf{LWE}_{n,m,\alpha q}$) only with negligible probability.

Various algorithms for different tasks such as sampling from $\Lambda^{\perp}(\mathbf{G})$ or inverting LWE instances are presented in the full version [44]. In this paper we use those algorithms in a block-box way and take them as given.

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., [7]), that enable the recovery of the error term. As a result, we take advantage of an increased message throughput per ciphertext. In the full version [44] we provide a 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 $\mathsf{encode} : \{0,1\}^m \rightarrow \mathbb{Z}_q^{m/k}$ and $\mathsf{decode} : \mathbb{Z}_q^{m/k} \rightarrow \{0,1\}^m$ 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 $\mathbf{G}\mathbf{x} \bmod q = \mathsf{encode}(\mathbf{m})$ holds. Sampling from $D_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}),r}$ is performed very efficiently (see [44]) 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 1 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 exemplarily, 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 [6, Lemma 2.10].

The message embedding functionality comes at almost no costs. Let k be a factor of m_2. One samples $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{m_1},r}$ and a preimage $\mathbf{e}_2 \leftarrow_R D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),r}$ for the syndrome $\mathbf{v} = \mathsf{encode}(\mathbf{m} \oplus H(\mathbf{e}_1))$ for some random function $H : \{0,1\}^* \rightarrow \{0,1\}^{m_2}$. 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 the following lemmata and the case of uniformly distributed error vectors is presented in the full version [44].

Lemma 1 (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 $\mathbf{v} \leftarrow_R \mathbb{Z}_q^l$ with $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ and $\epsilon = \mathsf{negl}(\lambda)$ is negligible.*

Lemma 2 (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 $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ 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. We make use of the gadget matrix $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^\top$ for $\mathbf{g}^\top = (1, \ldots, 2^{k-1})$. For simplicity, assume $q = 2^k$. For general q, the preimage sampling algorithm for $\Lambda^\perp(\mathbf{G})$ is more involved (see [7]).

Definition 3 (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 $H : \mathbb{Z}_q^n \times \mathbb{Z}_q^{m_1} \rightarrow \{0,1\}^{m_2}$ 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* $\mathbf{e}_2 \leftarrow_R \mathcal{D}_{\Lambda^{\perp}_{\mathbf{v}}(\mathbf{G}),\alpha q}$.
- *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 4 (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 $\text{A-LWE}^H_{n,m_1,m_2,\alpha q}$) *problem asks upon input $\mathbf{m} \in \{0,1\}^{m_2}$ to distinguish in polynomial time (in n) between samples $(\mathbf{A}_i, \mathbf{b}_i^\top) \leftarrow_R L^{\text{A-LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ 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 $\text{A-LWE}^H_{n,m_1,m_2,\alpha q}$) *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 $(\mathbf{A}_i, \mathbf{b}_i) \leftarrow_R L^{\text{A-LWE}}_{n,m_1,m_2,\alpha q}(\mathbf{m})$ for secret $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$.*
- *The Search-Message Augmented Learning with Errors* (search-m $\text{A-LWE}^H_{n,m_1,m_2,\alpha q}$) *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 $\text{LWE}^H_{n,m_1,m_2,\alpha q}$ *is hard if all polynomial time algorithms solve the* decision/search-s/search-m A $\text{LWE}^H_{n,m_1,m_2,\alpha q}$ *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 decision/search-s/search-m A $\text{LWE}_{n,m_1,m_2,\alpha q}$ the A-LWE 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 $\epsilon = \mathsf{negl}(\lambda) > 0$. 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* $\mathsf{search}\ \mathsf{LWE}_{n,m,\alpha q}$ *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 $\mathbf{m} \in \{0,1\}^{m_2}$.*

Proof. 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 $\mathsf{search}\ \mathsf{LWE}_{n,m,\alpha,q}$ 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 $\mathbb{H}_\infty(\mathbf{s}, \mathbf{e}_1) > \lambda$ and the same input will be queried with negligible probability. 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 drawing elements from the output range uniformly at random. Moreover, we have $\mathbb{H}_\infty(\mathbf{s}, \mathbf{e}_1) > \lambda$ such that the same input element $(\mathbf{s}, \mathbf{e}_1)$ will not be sampled again except with negligible probability. 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 sufficient entropy such that all pairs $(\mathbf{s}, (\mathbf{e}_1)_i)$ are distinct with overwhelming probability.
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 1 implies that $\mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{A}),r}$ is statistically indistinguishable from $\mathcal{D}_{\mathbb{Z}^{m_2},r}$ for $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$, if H has not been queried on input $(\mathbf{s}, \mathbf{e}_1)$ before. 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$_2$** 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 $\mathbf{Hybrid}_1$ 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 by q^* the query $(\mathbf{s}, \mathbf{e}_1)$ on H made by $\mathcal{A}$, 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 from $L^{\mathsf{LWE}}_{n,m,\alpha,q}$, if $(\mathbf{s}, \mathbf{e}_1)$ 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 $(\mathbf{s}, \mathbf{e}_1)$ even if given LWE samples. We stress that after returning the hash value of $(\mathbf{s}, \mathbf{e}_1)$ 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 $(\mathbf{s}, \mathbf{e}_1)$ 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 $\mathbf{Hybrid}_1$ 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.

□

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 [7, 16, 28, 29] 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.

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 $\epsilon = \mathsf{negl}(\lambda) > 0$. 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.*

- *If* search $\mathsf{LWE}_{n,m,\alpha q}$ *is hard, then* search-s $\mathsf{A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ *is hard.*
- *If* decision $\mathsf{LWE}_{n,m,\alpha q}$ *is hard, then* decision $\mathsf{A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ *resp.* search-m $\mathsf{A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ *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 decision $\mathsf{A\text{-}LWE}_{n,m_1,m_2,\alpha q}$ and search-m $\mathsf{A\text{-}LWE}_{n,m_1,m_2,\alpha q}$. And by essentially the same arguments we also deduce the hardness of search-s $\mathsf{A\text{-}LWE}_{n,m_1,m_2,\alpha q}$, 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 inputs only $\mathbf{s}$ (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 $\mathbf{e} \leftarrow_R \mathcal{D}_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),\alpha q}$. 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 $\mathbf{m} = \mathsf{decode}(\mathbf{G}\mathbf{e} \bmod q) \oplus H(\mathbf{s})$.

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 $\mathsf{A\text{-}LWE}_{n,0,m,\alpha q}$ *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 $\mathsf{A\text{-}LWE}_{n,0,m,\alpha q}$ 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. □

Remark. We like to note that one could increase the message throughput of our encryption scheme even further by embedding the message not only into the error term but also into (part of) the secret $\mathbf{s}$. This allows for an additional message of size approximately $n(k-1)$ bits. This is possible since each encryption query involves a fresh secret vector $\mathbf{s}$. One has to make sure that the hash function H is invoked on a value with sufficient entropy (e.g. the first n bits are random).

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 encryption scheme providing full CCA security when adopting the tagging approach presented in [45, 46]. In fact, we get this feature for free, if we instantiate our generic construction from Section 3.3 with the trapdoor provided in [7]. 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 [7], 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.

In this paper, we assume the reader is familiar with the various security models for encryption schemes. We refer to the full version [44] for a description of the CCA1, CCA2, and RCCA models.

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 [7], 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 LWE inversion algorithm as provided in [7, 44] 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 $\mathbf{m} = \mathbf{r} \oplus \mathsf{decode}(\mathbf{G}_m \mathbf{e}' \bmod q)$.
3. Output $\mathbf{m}$ as the message.

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

Proof. The proof is greatly simplified as compared to [7], 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 decision $\mathsf{A\text{-}LWE}_{n,0,m,\alpha q}$ 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 $\mathbf{b}_1 = \mathbf{s}^\top \bar{\mathbf{A}} + \mathbf{e}_1 \bmod q$ and $\mathbf{b}_2 = \mathbf{s}^\top (h(u)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}) + \mathbf{e}_2 \bmod q$, 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 [7]. 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. In combination with the one-time-pad approach from [7] and message injection into the secret vector $\mathbf{s}$, we can embed approximately $(c+2)nk - n$ message bits.

Further Applications. In the full version [44] we show how to use this cryptosystem as a main building block for the first lattice-based RCCA-secure encryption scheme and provide the schemes with an optional mode for high data load encryption. Moreover, we propose an asymmetric authenticated encryption scheme (amongst others) by exploiting the full entropy of the error vector for signatures and give a more efficient variant of the somewhat homomorphic encryption scheme initially proposed by Brakerski and Vaikuntanathan [34].

Acknowledgments. The work presented in this paper was performed within the context of the Software Cluster project Sinnodium and Software Campus project *IT-GiKo* by Rachid El Bansarkhani. It was funded by the German Federal Ministry of Education and Research (BMBF). Özgür Dagdelen and Johannes Buchmann are supported by BMBF within EC-SPRIDE. The authors thank Steven Galbraith for useful comments.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC, ACM Press (1996) 99–108
2. Regev, O.: New lattice-based cryptographic constructions. J. ACM **51** (2004) 899–942
3. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: 45th FOCS, IEEE Computer Society Press (2004) 372–381
4. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: Swifft: A modest proposal for FFT hashing. In: FSE. Volume 5086 of LNCS., Springer (2008) 54–72
5. Arbitman, Y., Dogon, G., Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFTX: A proposal for the SHA-3 standard (2008) In The First SHA-3 Candidate Conference.
6. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In Ladner, R.E., Dwork, C., eds.: 40th ACM STOC, ACM Press (2008) 197–206
7. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval, D., Johansson, T., eds.: EUROCRYPT 2012. Volume 7237 of LNCS., Springer (2012) 700–718
8. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: CRYPTO (1). (2013) 40–56
9. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In Prouff, E., Schaumont, P., eds.: CHES 2012. Volume 7428 of LNCS., Springer (2012) 530–547
10. Lyubashevsky, V.: Lattice signatures without trapdoors. In Pointcheval, D., Johansson, T., eds.: EUROCRYPT 2012. Volume 7237 of LNCS., Springer (2012) 738–755

11. Katz, J., Vaikuntanathan, V.: Smooth projective hashing and password-based authenticated key exchange from lattices. In Matsui, M., ed.: ASIACRYPT 2009. Volume 5912 of LNCS., Springer (2009) 636–652
12. Jintai Ding, X.L.: 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/.
13. Peikert, C.: Lattice cryptography for the internet. Cryptology ePrint Archive, Report 2014/070 (2014) http://eprint.iacr.org/.
14. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In Wagner, D., ed.: CRYPTO 2008. Volume 5157 of LNCS., Springer (2008) 554–571
15. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In Gabow, H.N., Fagin, R., eds.: 37th ACM STOC, ACM Press (2005) 84–93
16. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Mitzenmacher, M., ed.: 41st ACM STOC, ACM Press (2009) 333–342
17. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC. (2013) 575–584
18. Döttling, N., Müller-Quade, J.: Lossy codes and a new variant of the learning-with-errors problem. In Johansson, T., Nguyen, P., eds.: Advances in Cryptology – EUROCRYPT 2013. Volume 7881 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 18–34
19. Micciancio, D., Peikert, C.: Hardness of sis and lwe with small parameters. In: CRYPTO (1). (2013) 21–39
20. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, ACM (2009) 169–178
21. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In Ostrovsky, R., ed.: 52nd FOCS, IEEE Computer Society Press (2011) 97–106
22. Gentry, C., Halevi, S.: Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In Ostrovsky, R., ed.: 52nd FOCS, IEEE Computer Society Press (2011) 107–109
23. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In Safavi-Naini, R., Canetti, R., eds.: CRYPTO 2012. Volume 7417 of LNCS., Springer (2012) 868–886
24. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: ITCS. (2012) 309–325
25. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In Gilbert, H., ed.: EUROCRYPT 2010. Volume 6110 of LNCS., Springer (2010) 523–552
26. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In Gilbert, H., ed.: EUROCRYPT 2010. Volume 6110 of LNCS., Springer (2010) 553–572
27. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Rabin, T., ed.: CRYPTO 2010. Volume 6223 of LNCS., Springer (2010) 98–115
28. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In Gilbert, H., ed.: EUROCRYPT 2010. Volume 6110 of LNCS., Springer (2010) 1–23

29. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In Kiayias, A., ed.: CT-RSA 2011. Volume 6558 of LNCS., Springer (2011) 319–339
30. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In Johansson, T., Nguyen, P., eds.: Advances in Cryptology – EUROCRYPT 2013. Volume 7881 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 35–54
31. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In Ladner, R.E., Dwork, C., eds.: 40th ACM STOC, ACM Press (2008) 187–196
32. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In Matsui, M., ed.: ASIACRYPT 2009. Volume 5912 of LNCS., Springer (2009) 617–635
33. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In Boneh, D., ed.: CRYPTO 2003. Volume 2729 of LNCS., Springer (2003) 565–582
34. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Rogaway, P., ed.: CRYPTO 2011. Volume 6841 of LNCS., Springer (2011) 505–524
35. Groth, J.: Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In Naor, M., ed.: TCC 2004. Volume 2951 of LNCS., Springer (2004) 152–170
36. Phan, D., Safavi-Naini, R., Tonien, D.: Generic construction of hybrid public key traitor tracing with full-public-traceability. In Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., eds.: ICALP 2006, Part II. Volume 4052 of LNCS., Springer (2006) 264–275
37. Prabhakaran, M., Rosulek, M.: Rerandomizable RCCA encryption. In Menezes, A., ed.: CRYPTO 2007. Volume 4622 of LNCS., Springer (2007) 517–534
38. Xue, R., Feng, D.: Toward practical anonymous rerandomizable RCCA secure encryptions. In Qing, S., Imai, H., Wang, G., eds.: ICICS 07. Volume 4861 of LNCS., Springer (2007) 239–253
39. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In Cramer, R., ed.: PKC 2008. Volume 4939 of LNCS., Springer (2008) 360–379
40. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Journal on Computing **30** (2000) 391–437
41. Coron, J.S., Handschuh, H., Joye, M., Paillier, P., Pointcheval, D., Tymen, C.: GEM: A generic chosen-ciphertext secure encryption method. In Preneel, B., ed.: CT-RSA 2002. Volume 2271 of LNCS., Springer (2002) 263–276
42. Herzog, J., Liskov, M., Micali, S.: Plaintext awareness via key registration. In Boneh, D., ed.: CRYPTO 2003. Volume 2729 of LNCS., Springer (2003) 548–564
43. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing **36** (2007) 1301–1328
44. El Bansarkhani, R., Dagdelen, O., Buchmann, J.: Augmented learning with errors: The untapped potential of the error term. Cryptology ePrint Archive, Report 2014/733 (2014) `http://eprint.iacr.org/`.
45. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In Halevi, S., Rabin, T., eds.: TCC 2006. Volume 3876 of LNCS., Springer (2006) 581–600
46. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In Cachin, C., Camenisch, J., eds.: EUROCRYPT 2004. Volume 3027 of LNCS., Springer (2004) 207–222