Lattice Cryptography for the Internet

Chris Peikert[*]

January 31, 2014

Abstract

In recent years, *lattice-based* cryptography has been recognized for its many attractive properties, such as strong provable security guarantees and apparent resistance to quantum attacks, flexibility for realizing powerful tools like fully homomorphic encryption, and high asymptotic efficiency. Indeed, several works have demonstrated that for basic tasks like encryption and authentication, lattice-based primitives can have performance competitive with (or even surpassing) those based on classical mechanisms like RSA or Diffie-Hellman. However, there still has been relatively little work on developing lattice cryptography for deployment in *real-world* cryptosystems and protocols.

In this work we take a step toward that goal, by giving efficient and practical lattice-based protocols for key transport, encryption, and authenticated key exchange that are suitable as "drop-in" components for proposed Internet standards and other open protocols. The security of all our proposals is provably based (sometimes in the random-oracle model) on the well-studied "learning with errors over rings" problem, and hence on the conjectured worst-case hardness of problems on ideal lattices (against quantum algorithms).

One of our main technical innovations (which may be of independent interest) is a simple, low-bandwidth *reconciliation* technique that allows two parties who "approximately agree" on a secret value to reach *exact* agreement, a setting common to essentially all lattice-based encryption schemes. Our technique reduces the ciphertext length of prior (already compact) encryption schemes nearly twofold, at essentially no cost.

1 Introduction

Recent progress in lattice cryptography, especially the development of efficient *ring-based* primitives, puts it in excellent position for use in practice. In particular, the *short integer solution over rings* (ring-SIS) problem [Mic02, PR06, LM06] (which was originally inspired by the NTRU cryptosystem [HPS98]) has served as a foundation for practical collision-resistant hash functions [LMPR08, ADL$^+$08] and signature schemes [LM08, GPV08, Lyu12, MP12, DDLL13], while the *learning with errors over rings* (ring-LWE) problem [LPR10, LPR13] is at the heart of many kinds of encryption schemes. Much like their less efficient integer-based counterparts SIS [Ajt96, MR04, GPV08] and LWE [Reg05, Pei09, BLP$^+$13], both ring-SIS and ring-LWE enjoy strong provable hardness guarantees: they are hard on the average as long as the Shortest

[*] School of Computer Science, College of Computing, Georgia Institute of Technology. This material is based upon work supported by the National Science Foundation under CAREER Award CCF-1054495, by DARPA under agreement number FA8750-11-C-0096, and by the Alfred P. Sloan Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, DARPA or the U.S. Government, or the Sloan Foundation. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

Vector Problem is hard to approximate on so-called *ideal* lattices in the corresponding ring, *in the worst case*. These results provide good theoretical evidence that ring-SIS and ring-LWE are a solid foundation on which to design cryptosystems, which is reinforced by concrete cryptanalytic efforts (e.g., [CN11, LP11, LN13]). (We refer the reader to [Mic02, PR06, LM06, LPR10, LPR13] for further details on these problems' attractive efficiency and security properties.)

By now there is a great deal of theoretical work constructing a broad range of powerful cryptographic objects from (ring-)SIS and (ring-)LWE. However, far less attention has been paid to lower-level, "workhorse" primitives like key exchange and key transport protocols, which are widely used on real-world networks like the Internet. Indeed, almost all asymmetric cryptography standards are still designed around traditional mechanisms like Diffie-Hellman [DH76] and RSA [RSA78].

1.1 Our Contributions

Toward the eventual goal of broader adoption and standardization of efficient lattice-based cryptography, in this work we give efficient and practical lattice-based protocols for central asymmetric tasks like encryption, key encapsulation/transport, and authenticated key exchange (AKE). Our proposals can all be proved secure (in some cases, in the random oracle model [BR93b]) in strong, commonly accepted attack models, based on the presumed hardness of the ring-LWE problem plus other generic assumptions (e.g., signatures and message authentication codes).

Because our goal is to obtain primitives that are suitable for *real-world* networks like the Internet, we seek designs that adhere *as closely as possible* to the abstract protocols underlying existing proposed standards, e.g., IETF RFCs like [Hou03, RKBT10, HC98, Kau05, KHNE10]. This is so that working code and other time-tested solutions to engineering challenges can be reused as much as possible. Existing proposals are built around classical mechanisms like Diffie-Hellman and RSA, and ideally we would just be able to substitute those mechanisms with lattice-based ones without affecting the protocols' surrounding structure. However, lattices have very different mathematical properties than RSA and Diffie-Hellman, and many protocols are not easily adapted to use lattice-based mechanisms, or can even become *insecure* if one does so. Fortunately, we are able to show that in certain cases, existing protocols can be generalized so as to yield secure lattice-based instantiations, without substantially affecting their overall form or security analysis.

In the rest of this introduction we give an overview of our proposals.

Encryption and key transport. We first consider the task of asymmetric *key encapsulation* (also known as key transport), where the goal is for a sender to transmit a random cryptographic key K using the receiver's public key, so that K can be recovered only by the intended receiver. This task is central to the use of "hybrid" encryption, in which the parties later encrypt and/or authenticate bulk data under K using symmetric algorithms. Of course, one way to accomplish this goal is for the sender to choose K and simply encrypt it under the receiver's public encryption key. However, it is conceptually more natural (and can offer better efficiency and security bounds) to use a *key encapsulation mechanism* (KEM), in which the key K is produced as an *output* of the sender's "encapsulation" algorithm, which is run on the receiver's public key alone.

Our first technical contribution is a new ring-LWE-based KEM that has better bandwidth (i.e., ciphertext length) than prior compact encryption/KEM schemes [LPR10, LPR13] by nearly a factor of two, at essentially no cost in security or other efficiency measures. The improvement comes from our new, simple "reconciliation" technique that allows the sender and receiver to (noninteractively) reach *exact* agreement from their *approximate* or "noisy" agreement on a ring element. (See Section 3 for details.) Compared to the encryption schemes of [LPR10, LPR13], this technique allows us to replace one of the two ring elements modulo $q = \mathrm{poly}(n)$ in the ciphertext with a *binary* string of the same dimension n, thus nearly halving the ciphertext length. (See Section 4 for details.) We remark that approximate agreement is common to essentially all lattice-based encryption and key-agreement protocols, and our reconciliation technique is general enough to apply to all of them.

The KEM described above is *passively* secure, i.e., secure against passive eavesdroppers that see the public keys and ciphertexts, but do not create any of their own. Many applications require a much stronger form of security against *active* attackers, or more formally, security against adaptive chosen-ciphertext attacks. The literature contains several actively secure encryption/KEM schemes (sometimes in the random oracle model), obtained either via generic or semi-generic transformations from simpler objects (e.g., [DDN91, BR94, Sho01, FO99a, FO99b]), or more directly from particular algebraic structures and assumptions (e.g., [CS98, CS02, BCHK07, PW08]). For various reasons, most of these construction paradigms turn out to be unsuitable for obtaining highly efficient, actively secure lattice-based KEM/encryption schemes (see Section 5.3 for discussion). One method that does work well, however, is the Fujisaki-Okamoto transformation [FO99b]. In Section 5 we apply it to obtain an actively secure encryption and KEM scheme that is essentially as efficient as our passively secure KEM. This can be used as an alternative to, e.g., RSA-based actively secure key encapsulation as in the proposed standard [RKBT10].

Authenticated key exchange (AKE). An AKE protocol allows two parties to generate a fresh, mutually authenticated secret key, e.g., for use in setting up a secure point-to-point channel. Formal attack models, security definitions, and protocols for AKE have been developed and refined in several works, e.g., [BR93a, BR95, Kra96, BCK98, Sho99, CK01, CK02b, CK02a, LMQ$^+$03, Kra05]. In this work we focus on the strong notion of "SK-security" [CK01] in the "post-specified peer" model [CK02a]. This model is particularly relevant to the Internet because it allows the identity of the peering party to be discovered during the protocol, rather than specified in advance. It also ensures other desirable properties like perfect forward secrecy.

We give a generalization of an AKE protocol of Canetti and Krawczyk [CK02a] which inherits from Krawczyk's SIGMA family of protocols [Kra03], and underlies the Internet Key Exchange (IKE) proposed standard [HC98, Kau05, KHNE10]. All these protocols are built specifically around the (unauthenticated) Diffie-Hellman key-exchange mechanism. We show that the Canetti-Krawczyk protocol can be generalized to instead use *any* passively secure KEM—in particular, our lattice-based one—with only minor changes to the proof of SK-security in the post-specified peer model. Again, we view the relative lack of novelty in our protocol and its analysis as a practical advantage, since it should eventually allow for the reuse of existing code and specialized knowledge concerning the real-world implementation of these protocols.

1.2 Organization

The rest of the paper is organized as follows.

- In Section 2 we recall the necessary cryptographic and mathematical background, including the relevant basics of cyclotomic rings and the ring-LWE problem.
- In Section 3 we give the details of our new reconciliation mechanism for transforming approximate agreement to exact agreement.
- In Section 4 we present and analyze our new passively secure KEM using the above reconciliation mechanism.
- In Section 5 we transform our passively secure KEM to an actively secure one using the Fujisaki-Okamoto transformation [FO99b], and discuss the problems with other candidate transformations.

- In Section 6 we present and analyze a generalization of the Σ_0 authenticated key exchange (AKE) protocol of Canetti and Krawczyk [CK02a], which can be instantiated with our passively secure KEM.

2 Preliminaries

For $x \in \mathbb{R}$, define $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor \in \mathbb{Z}$. For an integer $q \geq 1$, let $\mathbb{Z}_q$ denote the quotient ring $\mathbb{Z}/q\mathbb{Z}$, i.e., the ring of cosets $x + q\mathbb{Z}$ with the induced addition and multiplication operations. For any two subsets X, Y of some additive group, define $-X = \{-x : x \in X\}$ and $X + Y = \{x + y : x \in X, y \in Y\}$.

2.1 Cryptographic Definitions

As is standard in cryptography, in this work all algorithms are implicitly given the same *security parameter* λ (represented in unary), which indicates the desired level of security. Every defined algorithm must be *efficient*, i.e., it must run in time polynomial in λ. A function $f(\lambda)$ is *negligible* if it decreases faster than any inverse polynomial, i.e., $f(\lambda) = o(n^{-c})$ for every constant $c \geq 0$. We say that two distributions X, Y (or more accurately, two ensembles $\{X_\lambda\}, \{Y_\lambda\}$ of distributions indexed by λ) are *computationally indistinguishable* if for all efficient algorithms $\mathcal{A}$, $|\Pr[\mathcal{A}(X_\lambda)] - \Pr[\mathcal{A}(Y_\lambda)]|$ is negligible (in λ). This asymptotic treatment can be made more concrete by quantifying runtimes, probabilities, etc. more precisely.

We define encryption and key encapsulation in a model in which the algorithms have access to some public parameters, which are generated according to a setup algorithm by a trusted party, and which can be used by all parties. If no trusted party is available, then the setup algorithm can be run first as part of the key-generation algorithm, and the public parameters included in the resulting public key.

A public-key cryptosystem (PKC) with ciphertext space $\mathcal{C}$ and (finite) message space $\mathcal{M}$ is given by four efficient algorithms: Setup, Gen, Enc (which may be randomized) and Dec (which must be deterministic) having the following syntax:

- $\mathsf{Setup}()$ outputs a public parameter pp.
- $\mathsf{Gen}(pp)$ outputs a public encryption key pk and secret decryption key sk.
- $\mathsf{Enc}(pp, pk, \mu)$ takes a public key pk and a message $\mu \in \mathcal{M}$, and outputs a ciphertext $c \in \mathcal{C}$.
- $\mathsf{Dec}(sk, c)$ takes a decryption key sk and a ciphertext c, and outputs some $\mu \in \mathcal{M} \cup \{\perp\}$, where $\perp$ is some distinguished symbol denoting decryption failure.

For correctness, we require that for all $\mu \in \mathcal{M}$, if $pp \leftarrow \mathsf{Setup}()$, $(pk, sk) \leftarrow \mathsf{Gen}(pp)$ and $c \leftarrow \mathsf{Enc}(pk, \mu)$, then $\mathsf{Dec}(sk, c) = \mu$ with all but negligible probability over all the randomness in the experiment.

A *key encapsulation mechanism* (KEM), sometimes also called a *key transport mechanism* (KTM), is a one-message protocol for transmitting an ephemeral secret key to a receiver, using the receiver's public key. Unlike with encryption, the ephemeral key is an *output* of the sender's algorithm, not an explicit input. Formally, a KEM with ciphertext space $\mathcal{C}$ and (finite) key space $\mathcal{K}$ is given by efficient algorithms Setup, Gen, Encaps (which may be randomized) and Decaps (which must be deterministic) having the following syntax:

- $\mathsf{Setup}()$ outputs a public parameter pp.
- $\mathsf{Gen}(pp)$ takes the public parameter and outputs a public encapsulation key pk and secret decapsulation key sk.
- $\mathsf{Encaps}(pp, pk)$ take the public parameter and an encapsulation key pk, and outputs a ciphertext $c \in \mathcal{C}$ and a key $k \in \mathcal{K}$.

- $\mathsf{Decaps}(sk, c)$ takes a decapsulation key sk and a ciphertext c, and outputs some $k \in \mathcal{K} \cup \{\perp\}$, where $\perp$ is some distinguished symbol denoting decapsulation failure.

For correctness, we require that if $pp \leftarrow \mathsf{Setup}()$, $(pk, sk) \leftarrow \mathsf{Gen}(pp)$, and $(c, k) \leftarrow \mathsf{Encaps}(pk)$, we have $\mathsf{Decaps}(sk, c) = k$ with all but negligible probability over all the randomness in the experiment.

A KEM is *passively* secure, or more formally, satisfies IND-CPA security, if the outputs of the following "real" and "ideal" experiments are computationally indistinguishable:

$$\begin{array}{l|l}
pp \leftarrow \mathsf{Setup}() & pp \leftarrow \mathsf{Setup}() \\
(pk, sk) \leftarrow \mathsf{Gen}(pp) & (pk, sk) \leftarrow \mathsf{Gen}(pp) \\
(c, k) \leftarrow \mathsf{Encaps}(pp, pk) & (c, k) \leftarrow \mathsf{Encaps}(pp, pk) \\
 & k^* \leftarrow \mathcal{K} \\
\text{Output } (pp, pk, c, k) & \text{Output } (pp, pk, c, k^*)
\end{array}$$

Any passively secure KEM can be converted into a passively secure encryption scheme for message space $\mathcal{M} = \mathcal{K}$ by having the sender use the ephemeral key $k \in \mathcal{K}$ as a one-time pad to conceal the message (this assumes that $\mathcal{K}$ has a group structure); this lengthens the ciphertext by an element of $\mathcal{K}$. In the other direction, any passively secure encryption scheme can be converted into an passively secure KEM with key space $\mathcal{K} = \mathcal{M}$ by having the sender choose the ephemeral key and encrypt it.

A public-key cryptosystem is *actively* secure (against adaptive chosen-ciphertext attacks), or more formally, satisfies IND-CCA security, if the following experiments for $b = 0, 1$ are computationally indistinguishable: generate $pp \leftarrow \mathsf{Setup}()$ and $(pk, sk) \leftarrow \mathsf{Gen}(pp)$, and give the adversary (pp, pk) and oracle access to $\mathsf{Dec}(sk, \cdot)$; then, when the adversary returns two messages $\mu_0, \mu_1 \in \mathcal{M}$, generate $c^* \leftarrow \mathsf{Enc}(pk, \mu_b)$ and give the adversary oracle access to $\mathsf{Dec}(sk, \cdot)$, with the restriction that it may not query the oracle on c^*.

2.2 Subgaussian Random Variables

In our constructions we analyze the behavior of "error" terms using the standard notion of *subgaussian* random variables, relaxed slightly as in [MP12]. (For further details and full proofs, see [Ver12].) For any $\delta \geq 0$, we say that a random variable X (or its distribution) over $\mathbb{R}$ is δ*-subgaussian* with parameter $r > 0$ if for all $t \in \mathbb{R}$, the (scaled) moment-generating function satisfies

$$\mathbb{E}[\exp(2\pi t X)] \leq \exp(\delta) \cdot \exp(\pi r^2 t^2).$$

By Markov's inequality, X has Gaussian tails, i.e., for all $t \geq 0$,

$$\Pr[|X| \geq t] \leq 2\exp(\delta - \pi t^2/r^2). \tag{2.1}$$

A standard fact is that any B-bounded centered random variable X (i.e., $\mathbb{E}[X] = 0$ and $|X| \leq B$ always) is 0-subgaussian with parameter $B\sqrt{2\pi}$.

We extend the notion of subgaussianity to vectors. Specifically, we say that a random real vector X is δ-subgaussian with parameter r if for all real unit vectors $\mathbf{u}$, the random variable $\langle \mathbf{u}, X \rangle \in \mathbb{R}$ is δ-subgaussian with parameter r. More generally, X and $\mathbf{u}$ may be taken from any real inner product space.

Fact 2.1. *If X_1 is δ_1-subgaussian with parameter r_1, and X_2 is δ_2 subgaussian with parameter r_2 conditioned on* any *value of X_1 (in particular, if X_1 and X_2 are independent), then $X_1 + X_2$ is $(\delta_1 + \delta_2)$-subgaussian with parameter $\sqrt{r_1^2 + r_2^2}$.*

2.3 Cyclotomic Rings

Here we briefly recall the relevant background on cyclotomic rings as they relate to ring-LWE. Many more details can be found in [LPR10, LPR13]. The latter reference especially covers many computational aspects of cyclotomics and ring-LWE.

For a positive integer index m, let $K = \mathbb{Q}(\zeta_m)$ and $R = \mathbb{Z}[\zeta_m] \subset K$ denote the mth cyclotomic number field and ring (respectively), where ζ_m denotes an abstract element having order m. The minimal polynomial of ζ_m over the rationals, i.e., the unique monic $f(X) \in \mathbb{Q}[X]$ of minimal degree having ζ_m as a root, is an integer polynomial called the mth *cyclotomic polynomial* $\Phi_m(X) \in \mathbb{Z}[X]$. Its complex roots are all the primitive mth roots of unity ω_m^i for $i \in \mathbb{Z}_m^*$, where $\omega_m = \exp(2\pi\sqrt{-1}/m) \in \mathbb{C}$. Therefore, R has degree $n = \varphi(m)$ as a ring extension of $\mathbb{Z}$, and is isomorphic to the quotient ring $\mathbb{Z}[X]/(\Phi_m(X))$ by identifying ζ_m with X, and similarly for K over $\mathbb{Q}$. In particular, $\{1, \zeta_m, \zeta_m^2, \ldots, \zeta_m^{n-1}\}$ is a $\mathbb{Z}$-basis of R, called the *power basis*. However, the power basis is just one particular basis of R, and it turns out that other bases are better suited to the kinds of computational operations and analysis used in ring-LWE cryptography (see [LPR13] for details). For the most part, in this work we remain agnostic to the choice of basis used for representing and operating upon ring elements, except when analyzing the coefficients of error terms, in which case we use the *decoding basis* of R, described in Section 2.3.2 below.

For any integer modulus $q \geq 1$, let R_q denote the quotient ring R/qR. For a given basis of R, we can represent any element of R_q by an n-dimensional vector of $\mathbb{Z}_q$-coefficients with respect to that basis. Fixing a basis of R and a set of distinguished representatives of $\mathbb{Z}_q$ therefore induces a set of distinguished representatives of R_q. For security, implementations must represent R_q-elements using their distinguished representatives with respect to some public basis, the choice of which must not depend on any secret information.

For any $p|m$, let $\zeta_p = \zeta_m^{m/p} \in R$ (which has order p), and define

$$g = \prod_{\text{odd prime } p|m} (1 - \zeta_p) \in R.$$

Also define $\hat{m} = m/2$ if m is even, and $\hat{m} = m$ otherwise. We recall a standard fact about these elements (see, e.g., [LPR13, Section 2.5.4]).

Fact 2.2. *The element g divides $\hat{m}$ in R, and is coprime in R with all integer primes* except *the odd primes p dividing m.*

2.3.1 Canonical Embedding

Here we recall the classical notion of the *canonical embedding* of K, which endows it (and hence $R \subset K$) with a geometry. There are $n = \varphi(m)$ ring embeddings (i.e., injective ring homomorphisms) $\sigma_i \colon K \to \mathbb{C}$ that fix $\mathbb{Q}$ pointwise. They are defined by $\sigma_i(\zeta_m) = \omega_m^i$ for each $i \in \mathbb{Z}_m^*$, where again $\omega_m = \exp(2\pi\sqrt{-1}/m) \in \mathbb{C}$. Note that these embeddings map ζ_m to each of the primitive mth roots of unity in $\mathbb{C}$, and that the embeddings come in conjugate pairs, i.e., $\sigma_{m-i} = \overline{\sigma_i}$.

The canonical embedding $\sigma \colon K \to \mathbb{C}^n$ is simply the concatenation of the above ring embeddings, i.e.,

$$\sigma(e) = \big(\sigma_i(e)\big)_{i \in \mathbb{Z}_m^*}.$$

Due to the conjugate pairs of embeddings, σ actually maps into the subspace $H \subseteq \mathbb{C}^n$ characterized by this conjugate symmetry. It is straightforward to verify that H is an n-dimensional real inner product space under the standard inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i x_i \overline{y_i}$ on $\mathbb{C}^n$.

We extend geometric notions, such as norms and subgaussianity, to K by identifying its elements with their canonical embeddings in H. In particular, the ℓ_2 (Euclidean) and ℓ_∞ norms on K are defined by

$$\|e\|_2 := \|\sigma(e)\|_2 = \left(\sum_{i \in \mathbb{Z}_m^*} |\sigma_i(e)|^2\right)^{1/2} \quad \text{and} \quad \|e\|_\infty := \|\sigma(e)\|_\infty = \max_{i \in \mathbb{Z}_m^*} |\sigma_i(e)|,$$

respectively. (For example, $\|1\|_2 = \sqrt{n}$ and $\|1\|_\infty = 1$.) Similarly, we say that $e \in K$ is δ-subgaussian with parameter r if $\sigma(e) \in H$ is.

Note that because $\sigma(e + e') = \sigma_i(e) + \sigma_i(e')$ and $\sigma_i(e \cdot e') = \sigma_i(e) \cdot \sigma_i(e')$, for $p \in \{2, \infty\}$ we have the triangle inequality and "expansion bound"

$$\begin{aligned} \|e + e'\|_p &\leq \|e\|_p + \|e'\|_p \\ \|e \cdot e'\|_p &\leq \|e\|_p \cdot \|e'\|_\infty. \end{aligned}$$

2.3.2 Decoding Basis

Here we recall a certain $\mathbb{Q}$-basis of K that plays an important role in the generation of error terms and in decryption for ring-LWE-based cryptosystems. As background, a central object in the definition and usage of ring-LWE is the fractional "codifferent" ideal $R^\vee = (\hat{m}/g)^{-1}R \subset K$. In [LPR13, Section 6] it is shown that a certain $\mathbb{Z}$-basis of $R^\vee$ (and hence $\mathbb{Q}$-basis of K), called the *decoding basis*, has essentially optimal error tolerance (e.g., for decryption) and admits fast sampling of error terms from appropriate distributions. We will not need the formal definition of the decoding basis in this work, but will only rely on a few of its important properties as demonstrated in [LPR13].

In this work, for convenience we avoid the codifferent ideal $R^\vee = (\hat{m}/g)^{-1}R$, and instead give an alternative (but equivalent) definition of the decoding basis, by multiplying by $\hat{m}/g \in R$ where appropriate to map $R^\vee$ to R. More specifically, we replace any $e^\vee \in R^\vee$ that would normally belong to the codifferent by $e = (\hat{m}/g)e^\vee \in R$. Similarly, we define the decoding basis of R to be $\hat{m}/g$ times the elements of the decoding basis of $R^\vee$. Then by linearity, the coefficient vector of any $e^\vee \in K$ with respect to the "true" decoding basis (of $R^\vee$) is identical to that of $e = (\hat{m}/g)e^\vee$ with respect to the decoding basis of R. In particular, any algorithm for sampling an error term $e^\vee$ as represented in the decoding basis of $R^\vee$, such as the one given in [LPR13, Section 6.3], is also (without any modification) an algorithm for sampling $e = (\hat{m}/g)e^\vee$ as represented in the decoding basis of R.

We caution that multiplying an element by $\hat{m}/g$ can significantly distort its canonical geometry. More precisely, the ratio of the lengths (or subgaussian parameters) of $e = (\hat{m}/g)e^\vee$ versus $e^\vee$ can vary widely depending on $e^\vee$, because the embeddings $\sigma_i(1/g)$ can vary widely in magnitude. However, multiplying by $g \in R$ undoes this distortion, because $\sigma(g \cdot e) = \hat{m} \cdot \sigma(e^\vee)$. So, we typically deal with error terms $e \in R$ where $g \cdot e$ is subgaussian, and analyze the coefficients of e itself with respect to the decoding basis of R. The following is a reformulation of [LPR13, Lemma 6.6] to our definition of decoding basis.

Lemma 2.3. *Let $e \in K$ be such that $g \cdot e$ is δ-subgaussian with parameter $\hat{m} \cdot r$, and let $e' \in K$ be arbitrary. Then every decoding-basis coefficient of $e \cdot e'$ is δ-subgaussian with parameter $r \cdot \|e'\|_2$.*

2.3.3 Error Distributions

In the context of ring-LWE we work with certain Gaussian-like error distributions over the number field K, and discretized to R. For $r > 0$, the Gaussian distribution D_r over $\mathbb{R}$ with parameter r has probability distribution function $\exp(-\pi x^2/r^2)/r$. More generally, the Gaussian distribution D_r over a real inner product space V (e.g., $\mathbb{R}^n$ or $H \subset \mathbb{C}^n$) is such that the marginal distributions $\langle \mathbf{u}, D_r \rangle = D_r$ (over $\mathbb{R}$) for all unit vectors $\mathbf{u} \in V$. In particular, D_r over the subspace $H \subset \mathbb{C}^n$ (as defined above in Section 2.3.1) outputs an element whose real and complex coordinates are all Gaussian with parameter $r/\sqrt{2}$, and are independent up to the conjugate symmetry of H. For convenience, but with a slight abuse of formality, we also define the Gaussian distribution D_r over the number field K to output an element $a \in K$ for which $\sigma(a) \in H$ has distribution D_r.[1]

In our applications we use error distributions of the form $\psi = (\hat{m}/g) \cdot D_r$ over K; the extra $\hat{m}/g$ factor corresponds to the translation from $R^\vee$ to R as described above in Section 2.3.2. We also *discretize* such distributions to the ring R, denoting the resulting distribution by $\chi = \lfloor \psi \rceil$, by sampling an element $a \in K$ from ψ and then rounding each of its rational decoding-basis coefficients to their nearest integers. (Other discretization methods are described in [LPR13, Section 2.4.2]; our applications can use any of them with only minor changes to the analysis.) We rely on the following facts from [LPR13].

Fact 2.4. *Let $e \leftarrow \chi$ where $\chi = \lfloor \psi \rceil$ for $\psi = (\hat{m}/g) \cdot D_r$. Then:*

1. *$g \cdot e$ is δ-subgaussian with parameter $\hat{m} \cdot \sqrt{r^2 + 2\pi \operatorname{rad}(m)/m}$ for some $\delta \leq 2^{-n}$.*
2. *$\|g \cdot e\|_2 \leq \hat{m} \cdot (r + \sqrt{\operatorname{rad}(m)/m}) \cdot \sqrt{n}$ except with probability at most 2^{-n}.*

For computational purposes, [LPR13, Section 6.3] gives a fast algorithm for sampling from $(\hat{m}/g) \cdot D_r$, where the output is represented as a vector of rational coefficients with respect to the decoding basis of R; this allows for immediate discretization.

2.4 Ring-LWE

We now recall the ring-LWE probability distribution and (decisional) computational problem. For simplicity and convenience for our applications, we present the problem in its discretized, "normal" form, where all quantities are from R or $R_q = R/qR$, and the secret is drawn from the (discretized) error distribution. (See [LPR10] for a more general form.)

Definition 2.5 (Ring-LWE Distribution). For an $s \in R$ and a distribution χ over R, a sample from the ring-LWE distribution $A_{s,\chi}$ over $R_q \times R_q$ is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(a, b = a \cdot s + e)$.

Definition 2.6 (Ring-LWE, Decision). The *decision* version of the ring-LWE problem, denoted $R\text{-}\mathsf{DLWE}_{q,\chi}$, is to distinguish with non-negligible advantage between independent samples from $A_{s,\chi}$, where $s \leftarrow \chi$ is chosen once and for all, and the same number of *uniformly random* and independent samples from $R_q \times R_q$.

Theorem 2.7 ([LPR10]). *Let R be the mth cyclotomic ring, having dimension $n = \varphi(m)$. Let $\alpha = \alpha(n) < \sqrt{\log n/n}$, and let $q = q(n)$, $q = 1 \bmod m$ be a $\operatorname{poly}(n)$-bounded prime such that $\alpha q \geq \omega(\sqrt{\log n})$. There is a $\operatorname{poly}(n)$-time quantum reduction from $\tilde{O}(\sqrt{n}/\alpha)$-approximate SIVP (or SVP) on ideal lattices in R to solving $R\text{-}\mathsf{DLWE}_{q,\chi}$ given only $\ell - 1$ samples, where $\chi = \lfloor \psi \rceil$ and ψ is the Gaussian distribution $(\hat{m}/g) \cdot D_{\xi q}$ for $\xi = \alpha \cdot (n\ell/\log(n\ell))^{1/4}$.*

[1] This is an abuse because $\sigma(K)$ is not equal to H, but is merely dense in it. To be formal, we would define the Gaussian distribution over the field tensor product $K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R}$, which is isomorphic to H (as a real inner product space) under the natural extension of σ. Since in practice Gaussians can only be sampled with finite precision, in this work we ignore such subtleties.

Note that the above worst-case hardness result deteriorates with the number of samples ℓ; fortunately, all our applications require only a small number of samples.

In addition to the above theorem, a plausible conjecture is that the *search* version of ring-LWE is hard for the fixed error distribution $\psi = (\hat{m}/g) \cdot D_{\alpha q}$, where $\alpha q \geq \omega(\sqrt{\log n})$.[2] (Informally, the search problem is to *find* the secret s given arbitrarily many ring-LWE samples; see [LPR10] for a precise definition.) Unfortunately, for technical reasons it is not known whether this is implied by the worst-case hardness of ideal lattice problems in R, except for impractically large q and small α. However, it is proved in [LPR10, Theorem 5.3] that the decision version with error distribution ψ (or its discretization $\lfloor \psi \rceil$) is at least as hard as the search version. Note that unlike Theorem 2.7, this results avoids the extra $(n/\log n)^{1/4}$ factor in the error distribution for the decision version, which leads to better parameters in applications.

3 New Reconciliation Mechanism

As mentioned in the introduction, one of our contributions is a more bandwidth-efficient method for two parties to agree on a secret bit, assuming they "approximately agree" on a (pseudo)random value modulo q. This is based on a new reconciliation mechanism that we describe in this section.

For an integer p that divides q (where typically $p = 2$), we define the modular rounding function $\lfloor \cdot \rceil_p \colon \mathbb{Z}_q \to \mathbb{Z}_p$ as $\lfloor x \rceil_p := \lfloor \frac{p}{q} \cdot x \rceil$, and similarly for $\lfloor \cdot \rfloor_p$. Note that the function is well-defined on the quotient rings because $\frac{p}{q} \cdot q\mathbb{Z} = p\mathbb{Z}$. For now we have restricted to the case $p|q$ so that the rounding function is unbiased. In Section 3.2 below we lift this restriction, using randomness to avoid introducing bias.

3.1 Even Modulus

In this subsection let the modulus $q \geq 2$ be even, and define disjoint intervals $I_0 := \{0, 1, \ldots, \lfloor \frac{q}{4} \rceil - 1\}$, $I_1 := \{-\lfloor \frac{q}{4} \rfloor, \ldots, -1\} \bmod q$ consisting of $\lfloor \frac{q}{4} \rceil$ and $\lfloor \frac{q}{4} \rfloor$ (respectively) cosets in $\mathbb{Z}_q$. Observe that these intervals form a partition of all the elements $v \in \mathbb{Z}_q$ such that $\lfloor v \rceil_2 = 0$ (where we identify 0 and 1 with their residue classes modulo two). Similarly, $\frac{q}{2} + I_0$ and $\frac{q}{2} + I_1$ partition all the v such that $\lfloor v \rceil_2 = 1$.

Now define the *cross-rounding* function $\langle \cdot \rangle_2 \colon \mathbb{Z}_q \to \mathbb{Z}_2$ as

$$\langle v \rangle_2 := \lfloor \tfrac{4}{q} \cdot v \rfloor \bmod 2.$$

Equivalently, $\langle v \rangle_2$ is the $b \in \{0, 1\}$ such that v belongs to the disjoint union $I_b \cup (\frac{q}{2} + I_b)$; hence the name "cross-rounding." If v is uniformly random, then $\langle v \rangle_2$ is uniformly random if and only if $q/2$ is even; otherwise, $\langle v \rangle_2$ is biased toward zero. Regardless of this potential bias, however, the next claim shows that $\langle v \rangle_2$ hides $\lfloor v \rceil_2$ perfectly.

Claim 3.1. *For even q, if $v \in \mathbb{Z}_q$ is uniformly random, then $\lfloor v \rceil_2$ is uniformly random given $\langle v \rangle_2$.*

Proof. For any $b \in \{0, 1\}$, if we condition on $\langle v \rangle_2 = b$, then v is uniform over $I_b \cup (\frac{q}{2} + I_b)$. As already observed, if $v \in I_b$ then $\lfloor v \rceil_2 = 0$, whereas if $v \in (\frac{q}{2} + I_b)$ then $\lfloor v \rceil_2 = 1$, so $\lfloor v \rceil_2$ is uniformly random given $\langle v \rangle_2$. □

[2] The conjecture seems plausible even for the weaker bound $\alpha q \geq 1$. However, when $\alpha q = o(1)$, the search problem can be solved in subexponential $2^{o(n)}$ time, given a sufficiently large number of samples [AG11].

We now show that if $v, w \in \mathbb{Z}_q$ are sufficiently close, we can recover $\lfloor v \rceil_2$ given w and $\langle v \rangle_2$. Define the set $E := [-\frac{q}{8}, \frac{q}{8}) \cap \mathbb{Z}$, and define the *reconciliation* function $\mathrm{rec}\colon \mathbb{Z}_q \times \mathbb{Z}_2 \to \mathbb{Z}_2$ as

$$\mathrm{rec}(w, b) := \begin{cases} 0 & \text{if } w \in I_b + E \pmod q \\ 1 & \text{otherwise.} \end{cases}$$

Claim 3.2. *For even q, if $w = v + e \bmod q$ for some $v \in \mathbb{Z}_q$ and $e \in E$, then $\mathrm{rec}(w, \langle v \rangle_2) = \lfloor v \rceil_2$.*

Proof. Let $b = \langle v \rangle_2 \in \{0, 1\}$, so $v \in I_b \cup (\frac{q}{2} + I_b)$. Then $\lfloor v \rceil_2 = 0$ if and only if $v \in I_b$. This in turn holds if and only if $w \in I_b + E$, because $((I_b + E) - E) \subseteq I_b + (-\frac{q}{4}, \frac{q}{4})$ and $(\frac{q}{2} + I_b)$ are disjoint (modulo q). The claim follows. □

3.2 Odd Modulus

All of the above applies when q is even, but in applications of ring-LWE this is often not the case. (For instance, it is often desirable to let q be a sufficiently large prime, for efficiency and security reasons.) When q is odd, while it is possible to use the above methods to agree on a bit derived by rounding a uniform $v \in \mathbb{Z}_q$, the bit will be *biased*, and hence not wholly suitable as key material. Here we show how to avoid such bias by temporarily "scaling up" to work modulo $2q$, and introducing a small amount of extra randomness.

Define the randomized function $\mathrm{dbl}\colon \mathbb{Z}_q \to \mathbb{Z}_{2q}$ that, given a $v \in \mathbb{Z}_q$, outputs $\bar{v} = 2v - \bar{e} \in \mathbb{Z}_{2q}$ for some random $\bar{e} \in \mathbb{Z}$ that is uniformly random modulo two and independent of v, and small in magnitude (e.g., bounded by one).[3] The first of these properties imply that if v is uniformly random in $\mathbb{Z}_q$, then so is $\bar{v}$ in $\mathbb{Z}_{2q}$, and hence the following extension of Claim 3.1 holds:

Claim 3.3. *For odd q, if $v \in \mathbb{Z}_q$ is uniformly random and $\bar{v} \leftarrow \mathrm{dbl}(v) \in \mathbb{Z}_{2q}$, then $\lfloor \bar{v} \rceil_2$ is uniformly random given $\langle \bar{v} \rangle_2$.*

Moreover, if $w, v \in \mathbb{Z}_q$ are close, then so are $2w, \mathrm{dbl}(v) \in \mathbb{Z}_{2q}$, i.e., if $w = v + e \pmod q$ for some (small) e, then $2w = \bar{v} + (2e + \bar{e}) \pmod{2q}$. Therefore, to (cross-)round from $\mathbb{Z}_q$ to $\mathbb{Z}_2$, we simply apply dbl to the argument and then apply the appropriate rounding function from $\mathbb{Z}_{2q}$ to $\mathbb{Z}_2$. Similarly, to reconcile some $w \in \mathbb{Z}_q$ we apply rec to $2w \in \mathbb{Z}_{2q}$; note that this process is still deterministic.

3.3 Extending to Cyclotomic Rings

We extend (cross-)rounding and reconciliation to cyclotomic rings R using the decoding basis. For even q, the rounding functions $\lfloor \cdot \rceil_2, \langle \cdot \rangle_2\colon R_q \to R_2$ are obtained by applying their integer versions (from $\mathbb{Z}_q$ to $\mathbb{Z}_2$) coordinate-wise to the input's decoding-basis $\mathbb{Z}_q$-coefficients. Formally, if $D = \{d_j\} \subset R$ denotes the decoding basis and $v = \sum_j v_j \cdot d_j \in R_q$ for coefficients $v_j \in \mathbb{Z}_q$, then $\lfloor v \rceil_2 := \sum_j \lfloor v_j \rceil_2 \cdot d_j \in R_2$, and similarly for $\langle \cdot \rangle_2$. The reconciliation function $\mathrm{rec}\colon R_q \times R_2 \to R_2$ is obtained from its integer version as $\mathrm{rec}(w, b) = \sum_j \mathrm{rec}(w_j, b_j) \cdot d_j$, where $w = \sum_j w_j \cdot d_j$ and $b = \sum_j b_j \cdot d_j$.

For odd q, we define the randomized function $\mathrm{dbl}\colon R_q \to R_{2q}$ which applies its (randomized) integer version *independently* to each of the input's decoding-basis coefficients. The (cross-)rounding functions from R_q to R_2 are defined to first apply dbl to the argument, then (cross-)round the result from R_{2q} to R_2. To reconcile $w \in R_q$ we simply reconcile $2w \in R_{2q}$.

[3] For example, we could simply take $\bar{e}$ to be uniform over $\{0, 1\}$. However, it is often more analytically convenient for $\bar{e}$ to be zero-centered and hence subgaussian. To achieve this we can take $\bar{e} = 0$ with probability $1/2$, and $\bar{e} = \pm 1$ each with probability $1/4$.

4 Passively Secure KEM

In this section we construct, based on ring-LWE, an efficient key encapsulation mechanism (KEM) that is secure against *passive* (i.e., eavesdropping) attacks. In later sections this will be used as a component of actively secure constructions. Specifically, we use the KEM as part of an authenticated key exchange protocol, and we use the induced passively secure encryption scheme to obtain actively secure encryption/KEM schemes via the Fujisaki-Okamoto transformation.

Our KEM is closely related to the compact ring-LWE cryptosystem from [LPR13, Section 8.2] (which generalizes the one sketched in [LPR10] to arbitrary cyclotomics), with two main changes: first, we avoid using the "codifferent" ideal $R^\vee$ using the approach described in Section 2.3.2; second, we use the reconciliation mechanism from Section 3 to improve ciphertext length. A third minor difference is that the system is constructed explicitly as a KEM (not a cryptosystem), i.e., the encapsulated key is not explicitly chosen by either party. Instead, the sender and receiver "approximately agree" on a pseudorandom value in R_q using ring-LWE, and use the reconciliation technique from Section 3 to derive the ephemeral key from it.

As compared with the previous most efficient ring-LWE cryptosystems and KEMs, the new reconciliation mechanism reduces the ciphertext length by nearly a factor of two, because it replaces one of the ciphertext's two R_q elements with an R_2 element. So the ciphertext length is reduced from $2n \log q$ bits to $n(1 + \log q)$ bits, where n is both the dimension of R and the length of the agreed-upon key. In terms of security, the reconciliation technique requires a ring-LWE error rate that is half as large as in prior schemes, but this weakens the concrete security only very slightly. (The reason for the smaller error rate is that we need the error term's decoding-basis coefficients to be bounded by $q/8$ instead of by $q/4$; see Claim 3.2.) Of course, if necessary we can compensate for this security loss by increasing the parameters (and hence the key size) very slightly. For practical purposes, the improvement in ciphertext length seems to outweigh the small loss in security or key size.

4.1 Construction

The KEM is parameterized by:

- A positive integer m specifying the mth cyclotomic ring R of degree $n = \varphi(m)$.
- A positive integer modulus q which is coprime with every odd prime dividing m, so that $g \in R$ is coprime with q (see Fact 2.2). For efficiency and provable security, we typically take q to be prime and 1 modulo m (or if necessary, a product of such primes), which implies the coprimality condition.
- A discretized error distribution $\chi = \lfloor \psi \rceil$ over R, where $\psi = (\hat{m}/g) \cdot D_r$ is over K (see Section 2.3.3), for some parameter $r > 0$.

The ciphertext space is $\mathcal{C} = R_q \times R_2$, and the key space is $\mathcal{K} = R_2$. We can identify elements in $\mathcal{K} = R_2$ with bit strings in $\{0, 1\}^n = \mathbb{Z}_2^n$ in some canonical way, e.g., the jth bit of the string is the jth decoding-basis coefficient of the element in R_2.

In what follows we assume that q is odd (since this will typically be the case in practice), and use the randomized function $\operatorname{dbl}\colon \mathbb{Z}_q \to \mathbb{Z}_{2q}$ and (deterministic) reconciliation function $\operatorname{rec}\colon \mathbb{Z}_q \times \mathbb{Z}_2 \to \mathbb{Z}_2$ from Section 3.[4] In dbl we take the random term $\bar{e}$ to be 0 with probability $1/2$, and ± 1 each with probability $1/4$,

[4] If q is even, then the Encaps and Decaps algorithms can be simplified by using the deterministic (cross-)rounding and associated reconciliation functions from Section 3.1. Lemmas 4.1 and 4.2 then remain true as stated, with essentially the same (but somewhat simpler) proofs.

so that $\bar{e}$ is uniform modulo two (as needed for security) and 0-subgaussian with parameter $\sqrt{2\pi}$. We also extend dbl and rec to cyclotomic rings as described in Section 3.3.

The algorithms of the KEM are as follows.

- $\mathsf{KEM1.Setup}()$: choose $a \leftarrow R_q$ and output $pp = a$.
- $\mathsf{KEM1.Gen}(pp = a)$: choose $s_0, s_1 \leftarrow \chi$, let $b = a \cdot s_1 + s_0 \in R_q$, and output public key $pk = b$ and secret key $sk = s_1$.
- $\mathsf{KEM1.Encaps}(pp = a, pk = b)$: choose independent $e_0, e_1, e_2 \leftarrow \chi$. Let $u = e_0 \cdot a + e_1 \in R_q$ and $v = g \cdot e_0 \cdot b + e_2 \in R_q$. Let $\bar{v} \leftarrow \mathrm{dbl}(v)$ and output the encapsulation $c = (u, v' = \langle \bar{v} \rangle_2) \in R_q \times R_2$ and key $\mu = \lfloor \bar{v} \rceil_2 \in R_2$.
- $\mathsf{KEM1.Decaps}(sk = s_1, c = (u, v'))$: compute $w = g \cdot u \cdot s_1 \in R_q$ and output $\mu = \mathrm{rec}(w, v') \in R_2$.

4.2 Security

Lemma 4.1. *KEM1 is IND-CPA secure, assuming the hardness of R-$\mathsf{DLWE}_{q,\chi}$ given two samples.*

Proof. We show that, under the assumption, adjacent games below (where Gen and Encaps are from KEM1) are computationally indistinguishable:

$a \leftarrow R_q$	$a \leftarrow R_q$	$(a, b) \leftarrow R_q \times R_q$
$(b, sk) \leftarrow \mathsf{Gen}(a)$	$b \leftarrow R_q$	$(u, v) \leftarrow R_q \times R_q$
$((u, v'), \mu) \leftarrow \mathsf{Encaps}(a, b)$	$((u, v'), \mu) \leftarrow \mathsf{Encaps}(a, b)$	$\bar{v} \leftarrow \mathrm{dbl}(v), v' = \langle v \rangle_2$
		$\mu^* \leftarrow R_2$
Output $(a, b, (u, v'), \mu)$	Output $(a, b, (u, v'), \mu)$	Output $(a, b, (u, v'), \mu^*)$

Notice that the first game is the "real" IND-CPA attack game on KEM1 (as defined in Section 2.1). The last game is not actually the "ideal" IND-CPA game, because it does not involve Gen or Encaps, but at the end of the proof we argue that it is computationally indistinguishable from the ideal game.

First consider the first pair of adjacent games above. It is immediate that they are computationally indistinguishable (under the assumption) because (a, b) is either one sample from $A_{s_1,\chi}$ where $s_1 \leftarrow \chi$, or uniform over $R_q \times R_q$. A formal reduction from the R-$\mathsf{DLWE}_{q,\chi}$ problem is straightforward.

We now show that the second and third games above are computationally indistinguishable. To do so, we construct an efficient reduction $\mathcal{S}$. It takes as input two pairs $(a, u), (b', v) \in R_q \times R_q$, lets $\bar{v} \leftarrow \mathrm{dbl}(v)$, and outputs

$$(a, \quad b = g^{-1} \cdot b' \in R_q, \quad (u, v' = \langle \bar{v} \rangle_2), \quad \mu = \lfloor \bar{v} \rceil_2).$$

Note that $g^{-1} \in R_q$ is well-defined because g, q are coprime in R. Now first suppose that the inputs to $\mathcal{S}$ are drawn from $A_{e_0,\chi}$ where $e_0 \leftarrow \chi$. Then the output of $\mathcal{S}$ is distributed exactly as in the second game, because $a, b \in R_q$ are uniformly random and independent (since a, b' are, and g is a unit modulo q), because $u = e_0 \cdot a + e_1, v = e_0 \cdot b' + e_2 = g \cdot e_0 \cdot b + e_2$ for independent $e_1, e_2 \leftarrow \chi$, and because v', μ are computed exactly as in $\mathsf{Encaps}(a, b)$. Now suppose that the inputs given to $\mathcal{S}$ are uniformly random over $R_q \times R_q$ and independent. Then the output of $\mathcal{S}$ is distributed exactly as in the third game, because a, b, u, v are uniform and independent, and hence by Claim 3.3, $\mu = \lfloor \bar{v} \rceil_2$ is uniformly random conditioned on $v' = \langle \bar{v} \rangle_2$. Since $\mathcal{S}$ is efficient and the two types of inputs to $\mathcal{S}$ are computationally indistinguishable (by assumption), so are the second and third games.

Finally, we claim that the third game above is computationally indistinguishable from the "ideal" game in the IND-CPA attack on KEM1. This follows by modifying the first and second games above to additionally choose $\mu^* \leftarrow R_2$ and output it in place of μ. Then the modified first game is exactly the ideal game, and computational indistinguishability of adjacent games follows by the same reasoning as above. This completes the proof. □

4.3 Correctness

We now analyze the parameters of the scheme that suffice for correctness of decapsulation.

Lemma 4.2. *Suppose $\|g \cdot s_i\|_2 \leq \ell$ for $i = 0, 1$ (where s_i are the secret values chosen by KEM1.*Gen*), and*

$$(q/8)^2 \geq (r'^2 \cdot (2\ell^2 + n) + \pi/2) \cdot \omega^2$$

for some $\omega > 0$, where $r'^2 = r^2 + 2\pi \operatorname{rad}(m)/m$. Then KEM$1$.$\mathsf{Decaps}$ *decrypts correctly except with probability at most $2n \exp(3\delta - \pi\omega^2)$ over the random choices of KEM1.*Encaps*, for some $\delta \leq 2^{-n}$.*

Proof. On an encapsulation $(u = e_0 \cdot a + e_1, v = g \cdot e_0 \cdot b + e_2)$ produced under public key $(a, b = a \cdot s_1 + s_0)$, the decapsulation algorithm computes

$$w = g \cdot u \cdot s_1 = v + g(e_0 \cdot s_0 + e_1 \cdot s_1) - e_2 \in R_q.$$

Let $e = g(e_0 \cdot s_0 + e_1 \cdot s_1) - e_2 \in R$, and let $\bar{e} \in R$ be the random element chosen by $\bar{v} \leftarrow \operatorname{dbl}(v)$ in KEM1.Encaps, so $\bar{v} = 2v - \bar{e} \in R_{2q}$. Then by Claim 3.2, it suffices to show that the decoding-basis coefficients of $2e + \bar{e}$ are all in $[-\frac{2q}{8}, \frac{2q}{8})$ with the claimed probability. We do so by showing that the coefficients are all 3δ-subgaussian with parameter $2(r'^2 \cdot (2\ell^2 + n) + \pi/2)^{1/2}$. The lemma then follows by Equation (2.1) and the union bound over all n coefficients.

To prove the above claim on $2e + \bar{e}$, first recall from Item 1 of Fact 2.4 that $g \cdot e_i$ (for $i = 0, 1, 2$) is δ-subgaussian with parameter $\hat{m} \cdot r'$. Then because $\|g \cdot s_i\|_2 \leq \ell$ for $i = 0, 1$, by Lemma 2.3 the decoding-basis coefficients of $g \cdot e_i \cdot s_i$ are all δ-subgaussian with parameter $r' \cdot \ell$. Also, by Lemma 2.3 (with $e' = 1$ and $\|e'\|_2 = \sqrt{n}$), the decoding-basis coefficients of e_2 are all δ-subgaussian with parameter $r' \cdot \sqrt{n}$. Finally, by assumption the decoding-basis coefficients of $\bar{e}$ are all 0-subgaussian with parameter $\sqrt{2\pi}$. Since the e_i and $\bar{e}$ are all mutually independent, by Fact 2.1 the decoding-basis coefficients of $2e + \bar{e}$ are all 3δ-subgaussian with parameter $2(r'^2 \cdot (2\ell^2 + n) + \pi/2)^{1/2}$, as claimed. □

4.4 Instantiating the Parameters

We now instantiate the parameters to analyze their asymptotic behavior and the underlying (worst-case) hardness guarantees. These calculations work for arbitrary choices of m and error parameter $r \geq 1$, and can therefore be slightly loose by small constant factors. Very sharp bounds can easily be obtained for particular choices of m and r using Lemma 4.2.

Since $\operatorname{rad}(m)/m \leq 1$, by Item 2 of Fact 2.4 we have that each $\|g \cdot s_i\|_2 \leq \hat{m} \cdot (r + 1) \cdot \sqrt{n}$ except with probability at most 2^{-n}. Similarly, $r'^2 \leq r^2 + 2\pi$. Therefore, by taking $\omega = \sqrt{\ln(2n/\varepsilon)/\pi}$ and

$$q \geq 8\sqrt{(r^2 + 2\pi)(2\hat{m}^2 \cdot (r + 1)^2 + 1) \cdot n} \cdot \omega = O(\hat{m} \cdot r^2 \cdot \sqrt{n}) \cdot \omega,$$

we obtain a probability of decryption failure bounded by $\approx \varepsilon$. Thus we may take $q = O(r^2 \cdot n^{3/2} \log n)$ in the typical case where $\hat{m} = O(n)$ and, say, $\varepsilon = 2^{-128}$.

To apply Theorem 2.7 for $\ell = 2$ samples, we let $r = \xi q$ and $\xi = \alpha \cdot (3n/\log(3n))^{1/4}$, where

- $r = (3n/\log(3n))^{1/4} \cdot \omega(\sqrt{\log n})$ to guarantee $\alpha q \geq \omega(\sqrt{\log n})$, and
- $q = O(r^2 \cdot n^{3/2} \log n) = \tilde{O}(n^2)$ is a sufficiently large prime congruent to one modulo m.

Then we obtain that $R\text{-}\mathsf{DLWE}_{q,\chi}$ is hard (and hence the KEM is IND-CPA secure, by Lemma 4.1) assuming that SVP on ideal lattices in R is hard to approximate to within $\tilde{O}(\sqrt{n}/\alpha) = \tilde{O}(\sqrt{n} \cdot q) = \tilde{O}(n^{5/2})$ factors for quantum algorithms.

Alternatively, we may conjecture that the search version of ring-LWE with error distribution $\psi = D_r$ is hard for $r \geq \omega(\sqrt{\log n})$ (or even $r \geq 1$), which by [LPR10, Theorem 5.3] implies that $R\text{-}\mathsf{DLWE}_{q,\chi}$ is hard as well. This lets us use a modulus as small as $q = \tilde{O}(n^{3/2})$, and implies a smaller modulus-to-noise ratio of $q/r = \tilde{O}(n^{3/2})$, rather than $\tilde{O}(n^{7/4})$ as when invoking Theorem 2.7 above. A smaller modulus-to-noise ratio provides stronger concrete security against known attacks, so this parameterization may be preferred in practice.

5 Actively Secure KEM

In this section construct an actively secure (i.e., secure under chosen-ciphertext attack) encryption scheme, using the passively secure encryption derived from KEM1 as a component. As noted in the preliminaries, actively secure encryption immediately yields an actively secure KEM or *key transport* protocol. Our construction may be seen as an alternative to proposed Internet standards for RSA-based key transport, such as [Hou03, RKBT10].

5.1 Overview

The literature contains many constructions of actively secure encryption, both in the standard and random-oracle models, and from both general assumptions and specific algebraic or structural ones (including lattices and LWE), e.g., [DDN91, BR93b, BR94, BR97, FO99a, FO99b, OP01, ABR01, CS98, CS02, BCHK07, PW08, Pei09, MP12]. Since our focus here is on efficiency, we allow for the use of the random-oracle heuristic as well as potentially strong (but plausible) non-standard assumptions. However, even with this permissive approach, it turns out that most known approaches for obtaining active security are either *insecure* when applied to our KEM (and other lattice-based encryption schemes more generally), or are unsuitable for other reasons. See Section 5.3 for further discussion on this point.

Considering all the options from the existing literature, we conclude that the best choice appears to be the second Fujisaki-Okamoto transformation [FO99b], which converts any passively secure encryption scheme into one which is provably actively secure, in the random-oracle model. (Note that the transformation requires an *encryption* scheme, and cannot be applied directly to a KEM.) Among the reasons for our choice are that the original passively secure scheme can have a minimally small plaintext space, and the resulting scheme is a "hybrid" one, i.e., it *symmetrically* encrypts a plaintext of arbitrary length. However, the transformation does have one important efficiency and implementation disadvantage in our setting: the random oracle's output is used as the *randomness* for asymmetric encryption, and the decryption algorithm re-runs the encryption algorithm with the same randomness to check ciphertext validity. This is somewhat unnatural in the (ring-)LWE setting, where encryption uses many random bits to generate high-precision Gaussians.[5]

[5] This disadvantage could be mitigated by using *uniformly random* error terms from a small interval, rather than Gaussians. When appropriately parameterized, the (ring-)LWE problem does appear to be hard with such errors, and there is some theoretical evidence of hardness as well [DMQ13, MP13]. However, the theoretical bounds are rather weak, and more investigation of concrete security is certainly needed.

We therefore slightly modify the construction so that the random oracle's output is used as the seed of a cryptographic pseudorandom generator (sometimes also referred to as a stream cipher), which produces the randomness for asymmetric encryption.

We remark that another approach is to use a different transformation, such as one like OAEP [BR94, Sho01] or REACT [OP01], in which the asymmetric encryption randomness is "freely chosen." In our context, these transformations require the use of an injective trapdoor function. Such functions can be constructed reasonably efficiently based on (ring-)LWE [PW08, GPV08, MP12], but it is not clear whether they can offer efficiency and bandwidth comparable to that of our passively secure KEM. An interesting open problem is to devise a passive-to-active security transformation that does not suffer any of the above-discussed drawbacks.

5.2 Construction

Our (actively secure) encryption scheme PKC2 is parameterized by:

- an integer $N \geq 0$, the bit length of the messages that PKC2 will encrypt;
- an asymmetric encryption scheme PKC with message space $\{0,1\}^n$, where PKC.Enc uses at most L uniformly random bits (i.e., PKC.$\mathsf{Enc}(\cdot; r)$ is a deterministic function for any coins $r \in \{0,1\}^L$), e.g., the encryption scheme induced by KEM1;
- a cryptographic pseudorandom generator $\mathsf{PRG} \colon \{0,1\}^\ell \to \{0,1\}^L$, for some seed length ℓ;
- hash functions $G \colon \{0,1\}^n \to \{0,1\}^N$ and $H \colon \{0,1\}^{n+N} \to \{0,1\}^\ell$, modelled as independent random oracles.

PKC2 is defined as follows:

- PKC2.$\mathsf{Setup}()$: let $pp \leftarrow$ PKC.$\mathsf{Setup}()$ and output pp.
- PKC2.$\mathsf{Gen}(pp)$: let $(pk, sk) \leftarrow$ PKC.$\mathsf{Gen}(pp)$ and output public key pk and secret key sk.
- PKC2.$\mathsf{Enc}(pp, pk, \mu)$: choose $\sigma \leftarrow \{0,1\}^n$, let $c =$ PKC.$\mathsf{Enc}_{pk}(pp, \sigma; \mathsf{PRG}(H(\sigma \| \mu)))$ and $w = G(\sigma) \oplus \mu$, and output the ciphertext $c \| w$.
- PKC2.$\mathsf{Dec}(sk, (c, w))$: compute $\sigma =$ PKC.$\mathsf{Dec}(sk, c)$ and $\mu = G(\sigma) \oplus w$, and check whether $c \overset{?}{=}$ PKC.$\mathsf{Enc}_{pk}(pp, \sigma; \mathsf{PRG}(H(\sigma \| \mu)))$. If so, output μ, otherwise output $\perp$.

Theorem 5.1. *PKC2 is IND-CCA secure, assuming that PKC is passively* one-way *secure,* PRG *is a secure pseudorandom generator, and G and H are modeled as random oracles.*

The proof of Theorem 5.1 is essentially the same as the one given in [FO99b], with minor changes to deal the use of the pseudorandom generator. Since the generator is invoked only on uniformly random bits (from the random oracle H) that are independent of everything else, the changes are straightforward.

5.2.1 Variants

Removing PRG. Of course, by setting $\ell = L$ we can let $\mathsf{PRG} \colon \{0,1\}^L \to \{0,1\}^L$ be the identity function, which is a trivial (and trivially secure) pseudorandom generator. The possible disadvantage is that the random oracle H must then have output length L, which in practice would correspond to many invocations of a hash function on distinct but related inputs. How this compares to invoking PRG would depend on the choice of instantiations.

Alternative symmetric encryption. In PKC2.Enc, letting $w = G(\sigma) \oplus \mu$ is just a particular method of encrypting μ under symmetric key $G(\sigma)$, namely, the one-time pad. As in [FO99b], we may instead use any symmetric encryption which is *one-time* passively secure (i.e., the adversary cannot make any chosen-plaintext or chosen-ciphertext queries). With this generalization, it is important to use the variant transformation from [AGKS05], in which H is applied to $\sigma \| w$ rather than to $\sigma \| \mu$, and so w must be computed before c (and H's input length should be adjusted accordingly). This is because the symmetric encryption may not be a bijection between μ and w for every key $G(\sigma)$, and such a property is required by the security proof for the original Fujisaki-Okamoto transformation (but not for the variant from [AGKS05]).

5.3 Alternatives

We considered several other known methods for obtaining active security. Unfortunately, most of them are either *insecure* when instantiated with our KEM1, or suffer from other costly drawbacks. For example:

- Constructions in the spirit of "hashed ElGamal," such as DHIES [BR97, ABR01] or variants [CS98, Section 10], in which the key from the passively secure KEM (and possibly ciphertext as well) are hashed by a random oracle to derive the final output key, are *not* actively secure when instantiated with our KEM1 or others like it. Briefly, the reason is related to the search/decision equivalence for (ring-)LWE: the adversary can query the decryption oracle on specially crafted ciphertexts for which the random oracle input is one of only a small number of possibilities (and depends on only a small portion of the secret key), and can thereby learn the entire secret key very easily.

- For similar reasons, applying the REACT transformation [OP01] to our KEM does not yield an actively secure scheme, because the KEM is not one-way under a "plaintext checking attack" (OW-PCA) due to the search/decision equivalence.

- The Bellare-Rogaway [BR93b] and OAEP transformations [BR94, Sho01] cannot be applied to our KEM, because they require a trapdoor permutation (or an injective trapdoor function). We remark that injective trapdoor functions can be constructed from (ring-)LWE [PW08, GPV08, MP12], and the most recent constructions are even reasonably efficient. However, it is not clear whether they can compete with the efficiency and bandwidth of our KEM.

- The first Fujisaki-Okamoto transformation [FO99a] *does* yield actively secure encryption when instantiated with our KEM's associated encryption scheme. However, it has the big disadvantage that the message length of the resulting scheme is substantially shorter than that of the original one, by (say) at least 128 bits for reasonable security bounds. Since our KEM's plaintext-to-ciphertext expansion is somewhat large, it is important to keep the size of the plaintext as small as possible.

6 Authenticated Key Exchange

In this section we give a protocol for authenticated key exchange which may be instantiated using our passively secure KEM from Section 4, together with other generic cryptographic primitives like signatures, which may also be instantiated with efficient ring-based constructions, e.g., [GPV08, MP12, Lyu12, DDLL13].

6.1 Overview

Informally, a key-exchange protocol allows two parties to establish a common secret key over a public network. The first such protocol was given by Diffie and Hellman [DH76]. However, it is well-known that this protocol can only be secure against a *passive* adversary who only reads the network traffic, but does not modify it or introduce messages of its own. An *authenticated key exchange* (AKE) protocol authenticates the parties' identities to each other, and provides a "consistent view" of the completed protocol to the peers, even in the presence of an active adversary who may control the network entirely (e.g., it may delete, delay, inject, or modify messages at will). Moreover, an AKE protocol may provide various security properties even if the adversary compromises some of the protocol participants and learns their local secrets. For example, "perfect forward secrecy" ensures the security of secret keys established by prior executions of the protocol, even if the long-term secrets of one or both parties are exposed later on. An excellent in-depth (yet still informal) discussion of these issues, and of the design considerations for AKE protocols, may be found in [Kra03].

Formal attack models, security definitions, and abstract protocols for AKE have been developed and refined in several works, e.g., [BR93a, BR95, Kra96, BCK98, Sho99, CK01, CK02b, CK02a, LMQ$^+$03, Kra05]. Of particular relevance to this work is the notion of "SK-security" due to Canetti and Krawczyk [CK01], which was shown to be sufficient for the prototypical application of constructing secure point-to-point channels. However, this model is not entirely appropriate for networks like the Internet, where peer identities are not necessarily known at the start of the protocol execution, and where identity concealment may be an explicit security goal. With this motivation in mind, Canetti and Krawczyk then gave an alternative formalization of SK-security which is more appropriate in such settings, called the "post-specified peer" model [CK02a], and gave a formal analysis of an instance of the "SIGn-and-MAc" (SIGMA) family of protocols due to Krawczyk [Kra03]. (In [CK02b] they also investigated the relationship between SK-security, key exchange, and secure channels in Canetti's "universal composability" model [Can01, Can00].) The formal definitions of SK-security and the post-specified peer models are somewhat lengthy and we will not need them here, so we refer the reader to [CK01, CK02a] for the details.

Regarding real-world protocols, the Internet Key Exchange (IKE) protocols [HC98, Kau05, KHNE10] define an open standard for authenticated key exchange as part of the Internet Protocol Security (IPsec) suite [KA98, KS05]. IKE's signature-based authentication mode follows the design of the SIGMA protocols from [Kra03] that were formally analyzed in [CK02a].

Our contribution. In the next subsection we give a protocol, called Σ_0', which is a slight generalization of the Σ_0 protocol from [CK02a], which itself follows the SIGMA design [Kra03] underlying the IKE protocol. The only difference between Σ_0' and Σ_0 is that we replace the (unauthenticated) Diffie-Hellman key-agreement steps in Σ_0 with an abstract IND-CPA-secure KEM (which can be instantiated by our lattice-based KEM1 from Section 4). Such a replacement is possible because the Diffie-Hellman steps in Σ_0 are used only to establish the common secret key (whereas the other steps provide authentication), and because the protocol has designated "initiator" and "responder" roles. In particular, the responder gets the initiator's start message before having to prepare its response, so the start message can contain a (fresh) KEM public key and the responder can run the encapsulation algorithm using this key. The security proof for Σ_0' is also just a slight variant of the one for Σ_0, because the latter proof uses only the KEM-like features of Diffie-Hellman, and not any of its other algebraic properties.

As mentioned in the introduction, from a practical perspective we believe the relatively minor differences between Σ_0' and Σ_0 (and their security proofs) to be an advantage: it should lessen the engineering burden required to implement the protocol correctly and securely, and may facilitate migration from, and co-existence with, existing Diffie-Hellman-based implementations.

6.2 The Protocol

The protocol Σ_0' is parameterized by an (IND-CPA-secure) a digital signature scheme SIG, a key-encapsulation mechanism KEM with key space $\mathcal{K}$, a pseudorandom function $F\colon \mathcal{K} \times \{0,1\} \to \mathcal{K}'$, and a message authentication code MAC with key space $\mathcal{K}'$ and message space $\{0,1\}^*$. A successful execution of the protocol outputs a secret key in $\mathcal{K}'$.

As in [CK02a], we assume that each party has a long-term signing key for SIG whose corresponding verification key is registered and bound to its identity ID, and is accessible to all other parties. This may be achieved in a standard way using certificate authorities. We also assume that trusted public parameters pp for KEM have been generated by a trusted party using KEM.Setup, and are available to all parties. As noted in the preliminaries, if no trusted party is available then KEM.Setup can be folded into KEM.Gen.

1. **Start message** ($I \to R$): (sid, pk_I).

   The protocol is activated by the initiator ID_I with a session identifier sid, which must be distinct from all those of prior sessions initiated by ID_I. The initiator generates a fresh keypair $(pk_I, sk_I) \leftarrow$ KEM.$\mathsf{Gen}(pp)$, stores it as the state of the session $(\mathsf{ID}_I, \mathsf{sid})$, and sends the above message to the responder.

2. **Response message** ($R \to I$): $(\mathsf{sid}, c, \mathsf{ID}_R, \text{SIG.}\mathsf{Sign}_R(1, \mathsf{sid}, pk_I, c), \text{MAC.}\mathsf{Tag}_{k_1}(1, \mathsf{sid}, \mathsf{ID}_R))$.

   When a party ID_R receives a start message (sid, pk_I), if the session identifier sid was never used before at ID_R, the party activates session sid as responder. It generates an encapsulation and key $(c, k) \leftarrow$ KEM.$\mathsf{Encaps}(pp, pk_I)$, derives $k_0 = F_k(0)$ and $k_1 = F_k(1)$, and erases the values pk_I and k from its memory, storing (k_0, k_1) as the state of the session. It generates and sends the above response message, where SIG.Sign_R is computed using its long-term signing key, and MAC.Tag is computed using key k_1.

3. **Finish message** ($I \to R$): $(\mathsf{sid}, \mathsf{ID}_I, \text{SIG.}\mathsf{Sign}_I(0, \mathsf{sid}, c, pk_I), \text{MAC.}\mathsf{Tag}_{k_1}(0, \mathsf{sid}, \mathsf{ID}_I))$.

   When party ID_I receives the (first) response message $(\mathsf{sid}, c, \mathsf{ID}_R, \sigma_R, \tau_R)$ having session identifier sid, it looks up the state (pk_I, sk_I) associated with session sid and computes $k =$ KEM.$\mathsf{Decaps}(sk_I, c)$ and $k_0 = F_k(0), k_1 = F_k(1)$. It then retrieves the signature verification key of ID_R and uses that key to verify the signature σ_R on the message tuple $(1, \mathsf{sid}, pk_I, c)$, and also verifies the MAC tag τ_R on the message tuple $(1, \mathsf{sid}, \mathsf{ID}_R)$ under key k_1. If either verification fails, the session is aborted, its state is erased, and the session output is $(\mathsf{abort}, \mathsf{ID}_I, \mathsf{sid})$. If both verifications succeed, then ID_I completes the session as follows: it generates and sends the above finish message where SIG.Sign_I is computed using its long-term signing key, and MAC.Tag is computed using key k_1. It then produces public session output $(\mathsf{ID}_I, \mathsf{sid}, \mathsf{ID}_R)$ and session secret output k_0, and erases the session state.

4. **Responder completion**: when party ID_R receives the (first) finish message $(\mathsf{sid}, \mathsf{ID}_I, \sigma_I, \tau_I)$ having session identifier sid, it looks up the state (k_0, k_1) associated with session sid. It then retrieves the signature verification key of ID_I and uses that key to verify the signature σ_I on the message tuple $(0, \mathsf{sid}, c, pk_I)$, and also verifies the MAC tag τ_I on the message tuple $(0, \mathsf{sid}, \mathsf{ID}_I)$ under key k_1. If either verification fails, the session is aborted, its state is erased, and the session output is $(\mathsf{abort}, \mathsf{ID}_r, \mathsf{sid})$. If both verifications succeed, then ID_R completes the session with public session output $(\mathsf{ID}_R, \mathsf{sid}, \mathsf{ID}_I)$ and secret session output k_0, and erases the session state.

6.3 Security

Theorem 6.1. *The Σ_0' protocol is SK-secure in the post-specified peer model of [CK02a], assuming that SIG and MAC are existentially unforgeable under chosen-message attack, that KEM is IND-CPA secure, and that F is a secure pseudorandom function.*

The proof of Theorem 6.1 follows by straightforwardly adapting the one from [CK02a]. Because the changes are simple and affect only small parts of the proof, we do not duplicate the whole proof here, but only describe the differences.

According to the definition of SK-security in the post-specified peer model from [CK02a], we need to show two properties: property P1 is essentially "correctness," or more precisely, equality of the secret outputs when two uncorrupted parties $\mathsf{ID}_I, \mathsf{ID}_R$ complete matching sessions with respective public outputs $(\mathsf{ID}_I, \mathsf{sid}, \mathsf{ID}_R), (\mathsf{ID}_R, \mathsf{sid}, \mathsf{ID}_I)$. Property P2 is essentially "secrecy," or more precisely, that no efficient attacker (in the post-specified peer model) can distinguish a real response to a test-session query from a uniformly random response, with non-negligible advantage.

Property P1 follows by adapting the proof in [CK02a, Section 4.2, full version]. It suffices to show that both parties compute the same decapsulation key k. This is guaranteed by the correctness of KEM and the security of the signature scheme, which ensures that the key k is obtained by decapsulating the appropriate ciphertext. (Security of MAC or the PRF is not needed for this property.)

Property P2 follows by adapting the proof in [CK02a, Section 4.3, full version]. While the proof is several pages long, very little of it is specific to the Diffie-Hellman mechanism or the DDH assumption. For example, the proof does not use any algebraic properties of the Diffie-Hellman problem beyond its assumed pseudorandomness. In the proof from [CK02a], a distinguisher for the DDH problem is constructed, i.e., it gets as input a tuple (g, g^x, g^y, g^z) where either $z = xy$ or z is uniformly random modulo the order of the group generated by g. In our setting, we instead construct a distinguisher for the IND-CPA security of KEM, i.e., it gets as input a tuple (pp, pk, c, k) where either k is the decapsulation of ciphertext c, or is uniformly random in the key space $\mathcal{K}$. To modify the proof from [CK02a], throughout it we syntactically replace the components of the DDH tuple with the corresponding ones of the KEM tuple (replacing g^{xy} by the real decapsulation key k, and g^z for uniform and independent z by a uniformly random and independent key $k^* \in \mathcal{K}$). With these and corresponding other syntactic changes to the component lemmas, the proof from [CK02a] remains valid.

6.4 Variants and IKE

As in [CK02a], we can consider variants of Σ_0' that extend its functionality or security properties, and also some important differences in the real IKE protocol that affect the analysis.

Perhaps most importantly, the signatures modes of the IKE protocol do not actually include the special distinguishing values 0,1 in the signed/MAC-tagged response and finish messages. (These values were included in Σ_0 for "symmetry breaking," to ease the analysis.) The Σ_0 protocol remains secure even without these values, as shown in [CK02a, Section 5.1, full version] via a more involved analysis. The analysis also carries over to the corresponding Σ_0' variant, based on the negligible "collision" probabilities of two uncorrupted parties generating the same KEM public key pk, or an equal public key and KEM ciphertext. Passive security immediately implies that such collision probabilities are negligible.

Another important difference with the IKE signature mode is that in the response and finish messages of the latter, the MAC tag is not sent separately, but instead is treated as the message to be signed. (Because of this, the MAC tag is computed on a tuple of *all* the values that are either signed or tagged in Σ_0.) In order to handle this, we need the MAC.Tag algorithm to be deterministic, which is standard. Then the analysis in [CK02a, Section 5.2, full version] goes through unchanged, as it relies only on the security of the MAC and signature schemes. The resulting protocol (also without the 0,1 values) essentially corresponds to IKE's "aggressive mode of signature authentication."

Other changes include offering identity concealment via encryption; a protocol corresponding to IKE's "main mode with signature authentication;" and more. These are all analyzed in [CK02a, Sections 5.3-5.4], and that analysis also goes through essentially unchanged for Σ_0'.

References

[ABR01] M. Abdalla, M. Bellare, and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In *CT-RSA*, pages 143–158. 2001.

[ADL+08] Y. Arbitman, G. Dogon, V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFTX: A proposal for the SHA-3 standard, 2008. Submitted to NIST SHA-3 competition.

[AG11] S. Arora and R. Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, pages 403–415. 2011.

[AGKS05] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In *EUROCRYPT*, pages 128–146. 2005.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

[BCHK07] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

[BCK98] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In *STOC*, pages 419–428. 1998.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584. 2013.

[BR93a] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *CRYPTO*, pages 232–249. 1993.

[BR93b] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73. 1993.

[BR94] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, pages 92–111. 1994.

[BR95] M. Bellare and P. Rogaway. Provably secure session key distribution: the three party case. In *STOC*, pages 57–66. 1995.

[BR97] M. Bellare and P. Rogaway. Minimizing the use of random oracles in authenticated encryption schemes. In *ICICS*, pages 1–16. 1997.

[Can00] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. `http://eprint.iacr.org/`.

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145. 2001.

[CK01] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *EUROCRYPT*, pages 453–474. 2001. Full version at `http://eprint.iacr.org/2001/040`.

[CK02a] R. Canetti and H. Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In *CRYPTO*, pages 143–161. 2002. Full version at `http://eprint.iacr.org/2002/120`.

[CK02b] R. Canetti and H. Krawczyk. Universally composable notions of key exchange and secure channels. In *EUROCRYPT*, pages 337–351. 2002. Full version at `http://eprint.iacr.org/2002/059`.

[CN11] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT*, pages 1–20. 2011.

[CS98] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167226, 2003. Preliminary version in CRYPTO 1998.

[CS02] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64. 2002.

[DDLL13] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO*, pages 40–56. 2013.

[DDN91] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000. Preliminary version in STOC 1991.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[DMQ13] N. Döttling and J. Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In *EUROCRYPT*, pages 18–34. 2013.

[FO99a] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, pages 53–68. 1999.

[FO99b] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, pages 537–554. 1999.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. 2008.

[HC98] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409 (Proposed Standard), November 1998. Obsoleted by RFC 4306, updated by RFC 4109.

[Hou03] R. Housley. Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS). RFC 3560 (Proposed Standard), July 2003.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. 1998.

[KA98] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

[Kau05] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. RFC 4306 (Proposed Standard), December 2005. Obsoleted by RFC 5996, updated by RFC 5282.

[KHNE10] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996 (Proposed Standard), September 2010. Updated by RFCs 5998, 6989.

[Kra96] H. Krawczyk. SKEME: a versatile secure key exchange mechanism for Internet. In *NDSS*, pages 114–127. 1996.

[Kra03] H. Krawczyk. SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE-protocols. In *CRYPTO*, pages 400–425. 2003. Full version at `http://webee.technion.ac.il/~hugo/sigma.html`.

[Kra05] H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In *CRYPTO*, pages 546–566. 2005.

[KS05] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), December 2005. Updated by RFC 6040.

[LM06] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP (2)*, pages 144–155. 2006.

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, pages 37–54. 2008.

[LMPR08] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE*, pages 54–72. 2008.

[LMQ$^+$03] L. Law, A. Menezes, M. Qu, J. A. Solinas, and S. A. Vanstone. An efficient protocol for authenticated key agreement. *Des. Codes Cryptography*, 28(2):119–134, 2003.

[LN13] M. Liu and P. Q. Nguyen. Solving bdd by enumeration: An update. In *CT-RSA*, pages 293–309. 2013.

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA*, pages 319–339. 2011.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60(6):43:1–43:35, November 2013. Preliminary version in EUROCRYPT '10.

[LPR13] V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-LWE cryptography. In *EUROCRYPT*, pages 35–54. 2013.

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755. 2012.

[Mic02] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Preliminary version in FOCS 2002.

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718. 2012.

[MP13] D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *CRYPTO*, pages 21–39. 2013.

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[OP01] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA*, pages 159–175. 2001.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342. 2009.

[PR06] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, pages 145–166. 2006.

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. *SIAM J. Comput.*, 40(6):1803–1844, 2011. Preliminar version in STOC 2008.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005.

[RKBT10] J. Randall, B. Kaliski, J. Brainard, and S. Turner. Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS). RFC 5990 (Proposed Standard), September 2010.

[RSA78] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[Sho99] V. Shoup. On formal models for secure key exchange. Cryptology ePrint Archive, Report 1999/012, 1999. `http://eprint.iacr.org/`.

[Sho01] V. Shoup. OAEP reconsidered. *J. Cryptology*, 15(4):223–249, 2002. Preliminary version in CRYPTO 2001.

[Ver12] R. Vershynin. *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012. Available at `http://www-personal.umich.edu/~romanv/papers/non-asymptotic-rmt-plain.pdf`.