Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices

Jonathan Katz[1]* and Vinod Vaikuntanathan[2]

[1] University of Maryland
jkatz@cs.umd.edu
[2] IBM Research
vinodv@alum.mit.edu

Abstract. We describe a public-key encryption scheme based on lattices — specifically, based on the hardness of the *learning with error* (LWE) problem — that is secure against chosen-ciphertext attacks while admitting (a variant of) smooth projective hashing. This encryption scheme suffices to construct a protocol for password-based authenticated key exchange (PAKE) that can be proven secure based on the LWE assumption in the standard model. We thus obtain the first PAKE protocol whose security relies on a lattice-based assumption.

1 Password-Based Authenticated Key Exchange

Protocols for password-based authenticated key exchange (PAKE) enable two users to generate a common, cryptographically-strong key based on an initial, low-entropy, shared secret (i.e., a password). The difficulty in this setting is to prevent *off-line* dictionary attacks where an adversary exhaustively enumerates potential passwords on its own, attempting to match the correct password to observed protocol executions. Roughly, a PAKE protocol is "secure" if off-line attacks are of no use and the best attack is an *on-line* dictionary attack where an adversary must actively try to impersonate an honest party using each possible password. On-line attacks of this sort are inherent in the model of password-based authentication; more importantly, they can be detected by the server as failed login attempts and (at least partially) defended against.

Due to the widespread use of passwords, a significant amount of research has focused on designing PAKE protocols. Early work [13] (see also [14]) considered a "hybrid" model where users share public keys in addition to a password. In the more challenging "password-only" setting clients and servers are required to share *only* a password. Bellovin and Merritt [4] initiated research in this direction, and presented a PAKE protocol with heuristic arguments for its security. It was not until several years later that formal models for PAKE were developed [3, 5, 11], and provably secure PAKE protocols were shown in the random oracle/ideal cipher models [3, 5, 18].

* Work done while visiting IBM. Research supported by NSF grants #0627306 and #0716651, and NSF CAREER award #0447075.

Goldreich and Lindell [11] constructed the first PAKE protocol without random oracles, and their approach remains the only one for the plain model where there is no additional setup. Unfortunately, their protocol is inefficient in terms of communication, computation, and round complexity. (Nguyen and Vadhan [19] show efficiency improvements, but achieve a weaker notion of security. In any case, their protocol is similarly impractical.) The Goldreich-Lindell protocol also does not tolerate concurrent executions by the same party.

Katz, Ostrovsky, and Yung [17] demonstrated the first *efficient* PAKE protocol with a proof of security in the standard model; extensions and improvements of this protocol were given in [9, 6, 16, 8]. In contrast to the work of Goldreich and Lindell, these protocols are secure even under concurrent executions by the same party. On the other hand, these protocols all require a *common reference string* (CRS). While this may be less appealing than the "plain model," reliance on a CRS does not appear to be a serious drawback in the context of PAKE since the CRS can be hard-coded into the protocol implementation. A different PAKE protocol in the CRS model is given by Jiang and Gong [15].

PAKE based on lattices? Cryptographic primitives based on lattices are appealing because of known worst-case/average-case connections between lattice problems, as well as because several lattice problems are currently immune to quantum attacks. Also, the best-known algorithms for several lattice problems require exponential time (in contrast to sub-exponential algorithms for, e.g., factoring). None of the existing PAKE constructions (in either the random oracle or standard models), however, can be instantiated with lattice-based assumptions.[1] The barrier to constructing a lattice-based PAKE protocol using the KOY/GL approach [17, 9] is that this approach requires a CCA-secure encryption scheme (more generally, a non-malleable commitment scheme) with an associated *smooth projective hash system* [7, 9]. (See Section 2.) Until recently, the existence of CCA-secure encryption schemes based on lattices (even ignoring the additional requirement of smooth projective hashing) was open. Peikert and Waters [22] gave the first constructions of CCA-secure encryption based on lattices, but the schemes they propose are not readily amenable to the smooth projective hashing requirement. Subsequent constructions [24, 20, 12] do not immediately support smooth projective hashing either.

1.1 Our Results

Building on ideas of [24, 20, 12], we show a new construction of a CCA-secure public-key encryption scheme based on the hardness of the *learning with error* (LWE) problem [23]. We then demonstrate (a variant of) a smooth projective hash system for our scheme. This is the most technically difficult aspect of our work, and is of independent interest as the first construction of a smooth projective hash system (for a conjectured hard-on-average language) based on lattice

[1] To the best of our knowledge this includes the protocol of Goldreich and Lindell [11], which requires a one-to-one one-way function on an infinite domain (in addition to oblivious transfer, which can be based on lattice assumptions [21]).

assumptions. (Instantiating the smooth projective hash framework using lattice assumptions is stated as an open question in [21].) Finally, we show that our encryption scheme can be plugged into a modification of the Katz-Ostrovsky-Yung/Gennaro-Lindell framework [17, 9] to give a PAKE protocol based on the LWE assumption.

Organization of the paper. In Section 2 we define a variant of smooth projective hashing (SPH) that we call *approximate* SPH. We then show in Section 3 that a CCA-secure encryption scheme having an approximate SPH system suffices for our desired application to PAKE.

The main technical novelty of our paper is in the sections that follow. In Section 4 we review the LWE problem and some preliminaries. As a prelude to our main construction, we show in Section 5 a CPA-secure encryption scheme based on the LWE problem, with an associated approximate SPH system. In Section 6 we describe how to extend this initial scheme to obtain CCA-security.

Throughout the paper, we denote the security parameter by n.

2 Approximate Smooth Projective Hash Functions

Smooth projective hash functions were introduced by Cramer and Shoup [7]; we follow (and adapt) the treatment of Gennaro and Lindell [9], who extend the original definition. Rather than aiming for utmost generality, we tailor the definitions to our eventual application.

Roughly speaking, the differences between our definition and that of Gennaro-Lindell are as follows. (This discussion assumes familiarity with [9]; for the reader not already familiar with that work, a self-contained description is given below.) In [9] there are sets X and $L \subset X$; *correctness* is guaranteed for $x \in L$, while *smoothness* is guaranteed for $x \in X \setminus L$. Here, we require only *approximate* correctness, and moreover only for elements in a subset $\bar{L} \subseteq L$. Details follow.

Fix a CCA-secure (labeled) public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an efficiently recognizable message space $\mathcal{D}$ (which will correspond to the dictionary of passwords in our application to PAKE). We assume the encryption scheme defines a notion of *ciphertext validity* such that (1) validity of a ciphertext (with respect to pk) can be determined efficiently using pk alone, and (2) all honestly generated ciphertexts are valid. We also assume no decryption error.

For the rest of the discussion, fix a key pair (pk, sk) as output by $\mathsf{Gen}(1^n)$ and let $\mathcal{C}$ denote the set of valid ciphertexts with respect to pk. Define sets $X, \{\bar{L}_m\}_{m \in \mathcal{D}}$, and $\bar{L}$ as follows. First, set

$$X = \{(\mathsf{label}, C, m) \mid \mathsf{label} \in \{0,1\}^n;\ C \in \mathcal{C};\ m \in \mathcal{D}\}\,.$$

Next, for $m \in \mathcal{D}$ let $\bar{L}_m = \{(\mathsf{label}, \mathsf{Enc}_{pk}(\mathsf{label}, m), m) \mid \mathsf{label} \in \{0,1\}^n\} \subset X$; i.e., $\bar{L}_m$ is the set of honestly generated encryptions of m (using any label). Let $\bar{L} = \cup_{m \in \mathcal{D}} \bar{L}_m$. Finally, define

$$L_m = \{(\mathsf{label}, C, m) \mid \mathsf{label} \in \{0,1\}^n;\ \mathsf{Dec}_{sk}(\mathsf{label}, C) = m\}\,,$$

and set $L = \cup_{m \in \mathcal{D}} L_m$. (Recall we assume no decryption error, and so L_m depends only on pk.) Note that $\bar{L}_m \subseteq L_m$ for all m. Furthermore, for any ciphertext C and $\mathsf{label} \in \{0,1\}^n$ there is at most one $m \in \mathcal{D}$ for which $(\mathsf{label}, C, m) \in L$.

Approximate smooth projective hash functions. An *approximate smooth projective hash function* is a collection of keyed functions $\{H_k : X \to \{0,1\}^n\}_{k \in K}$, along with a *projection function* $\alpha : K \times (\{0,1\}^* \times \mathcal{C}) \to S$, satisfying notions of (approximate) *correctness* and *smoothness*:

Approximate correctness: If $x = (\mathsf{label}, C, m) \in \bar{L}$ then the value of $H_k(x)$ is approximately determined by $\alpha(k, \mathsf{label}, C)$ and x (in a sense we will make precise below).

Smoothness: If $x \in X \setminus L$ then the value of $H_k(x)$ is statistically close to uniform given $\alpha(k, \mathsf{label}, C)$ and x (assuming k was chosen uniformly in K).

We stress that, in contrast to [9], we require nothing for $x \in L \setminus \bar{L}$; furthermore, even for $x \in \bar{L}$ we require only approximate correctness. We highlight also that, as in [9], the projection function α should be a function of label, C only.

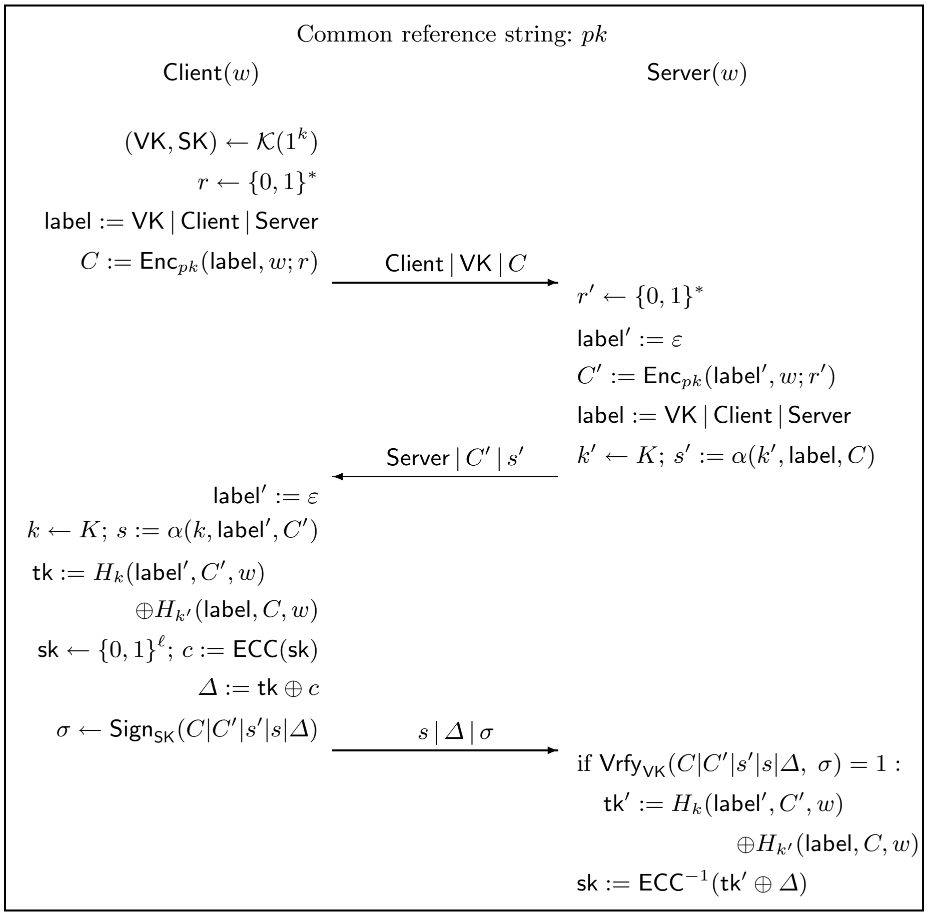
Formally, an $\varepsilon(n)$-approximate smooth projective hash function is defined by a sampling algorithm that, given pk, outputs $(K, G, \mathbb{H} = \{H_k : X \to \{0,1\}^n\}_{k \in K}, S, \alpha : K \times (\{0,1\}^* \times \mathcal{C}) \to S)$ such that:

1. There are efficient algorithms for (1) sampling a uniform $k \in K$, (2) computing $H_k(x)$ for all $k \in K$ and $x \in X$, and (3) computing $\alpha(k, \mathsf{label}, C)$ for all $k \in K$ and $(\mathsf{label}, C) \in \{0,1\}^* \times \mathcal{C}$.
2. For $x = (\mathsf{label}, C, m) \in \bar{L}$ the value of $H_k(x)$ is approximately determined by $\alpha(k, \mathsf{label}, C)$, relative to the Hamming metric. Specifically, let $\mathsf{Ham}(a, b)$ denote the Hamming distance of two strings $a, b \in \{0,1\}^n$. Then there is an efficient algorithm H' that takes as input $s = \alpha(k, \mathsf{label}, C)$ and $\bar{x} = (\mathsf{label}, C, m, r)$ for which $C = \mathsf{Enc}_{pk}(\mathsf{label}, m; r)$ and satisfies:
$$\Pr[\mathsf{Ham}(H_k(x),\ H'(s, \bar{x})) \geq \varepsilon \cdot n] = \mathsf{negl}(n),$$
where the probability is taken over choice of k.
3. For any $x = (\mathsf{label}, C, m) \in X \setminus L$, the following two distributions have statistical distance negligible in n:
$$\big\{k \leftarrow K; s = \alpha(k, \mathsf{label}, C) : \big(s, H_k(x)\big)\big\}$$
and
$$\{k \leftarrow K; s = \alpha(k, \mathsf{label}, C); v \leftarrow \{0,1\}^n : (s, v)\}.$$

3 A PAKE Protocol from Approximate SPH

We use the standard definition of security for PAKE [3, 17, 9].

Here, we show that a modification of the Gennaro-Lindell framework [9] can be used to construct a PAKE protocol from any CCA-secure encryption scheme that has associated with it an approximate smooth projective hash function as


Common reference string: pk

$\mathsf{Client}(w)$ $\qquad$ $\mathsf{Server}(w)$

$(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{K}(1^k)$

$r \leftarrow \{0,1\}^*$

$\mathsf{label} := \mathsf{VK} \mid \mathsf{Client} \mid \mathsf{Server}$

$C := \mathsf{Enc}_{pk}(\mathsf{label}, w; r)$ $\xrightarrow{\mathsf{Client} \mid \mathsf{VK} \mid C}$

$r' \leftarrow \{0,1\}^*$

$\mathsf{label}' := \varepsilon$

$C' := \mathsf{Enc}_{pk}(\mathsf{label}', w; r')$

$\mathsf{label} := \mathsf{VK} \mid \mathsf{Client} \mid \mathsf{Server}$

$\xleftarrow{\mathsf{Server} \mid C' \mid s'}$ $k' \leftarrow K;\ s' := \alpha(k', \mathsf{label}, C)$

$\mathsf{label}' := \varepsilon$

$k \leftarrow K;\ s := \alpha(k, \mathsf{label}', C')$

$\mathsf{tk} := H_k(\mathsf{label}', C', w)$

$\oplus H_{k'}(\mathsf{label}, C, w)$

$\mathsf{sk} \leftarrow \{0,1\}^\ell;\ c := \mathsf{ECC}(\mathsf{sk})$

$\Delta := \mathsf{tk} \oplus c$

$\sigma \leftarrow \mathsf{Sign}_{\mathsf{SK}}(C|C'|s'|s|\Delta)$ $\xrightarrow{s \mid \Delta \mid \sigma}$

if $\mathsf{Vrfy}_{\mathsf{VK}}(C|C'|s'|s|\Delta,\ \sigma) = 1$:

$\mathsf{tk}' := H_k(\mathsf{label}', C', w)$

$\oplus H_{k'}(\mathsf{label}, C, w)$

$\mathsf{sk} := \mathsf{ECC}^{-1}(\mathsf{tk}' \oplus \Delta)$

Fig. 1. A 3-round PAKE protocol. The common session key is sk.

defined in Section 2. A high-level overview of the protocol is given in Figure 1; a more detailed discussion follows.

Setup. We assume a common reference string is established before any executions of the protocol take place. The common reference string consists of a public key pk for a CCA-secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that has an associated ε-approximate smooth projective hash system $(K, G, \mathbb{H} = \{H_k : X \to \{0,1\}^n\}_{k \in K}, S, \alpha : K \times (\{0,1\}^* \times \mathcal{C}) \to S)$. We stress that no parties in the system need to hold the secret key corresponding to pk.

Protocol execution. We now describe an execution of the protocol between an honest client Client and server Server, holding common password w. To begin, the client runs a key-generation algorithm $\mathcal{K}$ for a one-time signature scheme to generate verification key VK and corresponding secret (signing) key SK. The client sets $\mathsf{label} := \mathsf{VK}|\mathsf{Client}|\mathsf{Server}$ and then encrypts the password w using this label to obtain ciphertext C. It then sends the message $\mathsf{Client}|\mathsf{VK}|C$ to the server.

Upon receiving the initial message $\mathsf{Client}|\mathsf{VK}|C$ from the client, the server computes its own encryption of the password using label $\mathsf{label}' = \varepsilon$, resulting in a ciphertext C'. The server also chooses a random hash key $k' \leftarrow K$ and then computes the projection $s' := \alpha(k', \mathsf{label}, C)$. It sends C' and s' to the client.

After receiving the second protocol message from the server, the client chooses a random hash key $k \leftarrow K$ and computes the projection $s := \alpha(k, \mathsf{label}', C')$. At this point it computes a *temporary* session key $\mathsf{tk} = H_k(\mathsf{label}', C', w) \oplus H_{k'}(\mathsf{label}, C, w)$, where $H_k(\mathsf{label}', C', w)$ is computed using the known hash key k, and $H_{k'}(\mathsf{label}, C, w)$ is computed using the randomness r that was used to generate C. (Recall that C is an honestly generated encryption of w.) Up to this point, the protocol follows the Gennaro-Lindell framework exactly. As will become clear, however, the server will not be able to recover tk but will instead only recover some value tk' that is *close* to tk; the rest of the client's computation is aimed at repairing this defect.

The client chooses a random session key $\mathsf{sk} \in \{0,1\}^\ell$ for some ℓ to be specified. Let $\mathsf{ECC} : \{0,1\}^\ell \to \{0,1\}^n$ be an error-correcting code correcting a 2ε-fraction of errors. The client computes $c := \mathsf{ECC}(\mathsf{sk})$ and sets $\Delta := \mathsf{tk} \oplus c$. Finally, it signs $C|C'|s'|s|\Delta$ and sends s, Δ, and the resulting signature σ to the server.

The server verifies σ in the obvious way and rejects if the signature is invalid. Otherwise, the server computes a temporary session key tk' analogously to the way the client did: that is, the server sets $\mathsf{tk}' = H_k(\mathsf{label}', C', w) \oplus H_{k'}(\mathsf{label}, C, w)$, where $H_{k'}(\mathsf{label}, C, w)$ is computed using the hash key k' known to the server, and $H_k(\mathsf{label}', C', w)$ is computed using the randomness r' that was used to generate C'. (Recall that C' is an honestly generated encryption of w.) Finally, the server computes $\mathsf{sk} := \mathsf{ECC}^{-1}(\mathsf{tk}' \oplus \Delta)$.

Correctness. We now argue that, in an honest execution of the protocol, the client and server compute matching session keys with all but negligible probability. Approximate correctness of the smooth projective hash function implies that $H_k(\mathsf{label}, C, w)$ as computed by the client is within Hamming distance εn from $H_k(\mathsf{label}, C, w)$ as computed by the server, except with negligible probability. The same holds for $H_{k'}(\mathsf{label}', C', w)$. Thus, with all but negligible probability we have $\mathsf{Ham}(\mathsf{tk}, \mathsf{tk}') \leq 2\varepsilon \cdot n$. Assuming this is the case we have

$$\mathsf{Ham}(\mathsf{tk}' \oplus \Delta,\ c) = \mathsf{Ham}(\mathsf{tk}' \oplus \Delta,\ \mathsf{tk} \oplus \Delta) \leq 2\varepsilon \cdot n,$$

and so $\mathsf{ECC}^{-1}(\mathsf{tk}' \oplus \Delta) = \mathsf{ECC}^{-1}(c) = \mathsf{sk}$.

Security. The proof of security of the protocol follows [17, 9] closely; we sketch the main ideas. First, as in [17, 9], we note that for a passive adversary (i.e., one that simply observes interactions between the server and the client), the shared session-key is pseudorandom. This is simply because the transcript of each interaction consists of semantically-secure encryptions of the password w and the projected keys of the approximate SPH system.

It remains to deal with active (man-in-the-middle) adversaries that modify the messages sent from the client to the server and back. The crux of our proof, as in [17, 9], is a combination of the following two observations (for concreteness, consider an adversary that interacts with a client instance holding password w).

- By the CCA-security of the encryption scheme, the probability that the adversary can construct a new ciphertext that decrypts to the client's password w is at most $q/|\mathcal{D}| + \mathsf{negl}(n)$, where q is the number of on-line attacks and $\mathcal{D}$ is the password dictionary.
- If the adversary sends the client a ciphertext that does not decrypt to the client's password, then the session-key computed by the client is statistically close to uniform conditioned on the adversary's view.

We defer a complete proof to the full version.

Recalling the definitions from Section 2, note that correctness of the protocol relies on (approximate) correctness for honestly generated encryptions of the correct password (i.e., for $x \in \bar{L}$), whereas security requires smoothness for ciphertexts that do not decrypt to the correct password (i.e., for $x \notin L$).

4 The Learning with Errors Problem

The "learning with errors" (LWE) problem was introduced by Regev [23] as a generalization of the "learning parity with noise" problem. For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{Z}_q$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and a noise term $x \leftarrow \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + x) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

For an integer $q = q(n)$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the learning with errors problem $\mathsf{LWE}_{q,\chi}$ is defined as follows: Given access to an oracle that outputs (polynomially many) samples from $A_{\mathbf{s},\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, output $\mathbf{s}$ with noticeable probability. The decisional variant of the LWE problem, denoted $\mathsf{distLWE}_{q,\chi}$, is to distinguish samples chosen according to $A_{\mathbf{s},\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ from samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Regev [23] shows that for $q = \mathsf{poly}(n)$ prime, the LWE and $\mathsf{distLWE}$ problems are polynomially equivalent.

Gaussian error distributions. For any $r > 0$, the density function of a one-dimensional Gaussian distribution over $\mathbb{R}$ is given by $D_r(x) = 1/r \cdot \mathsf{exp}(-\pi(x/r)^2)$. In this work we always use a *truncated* Gaussian distribution, i.e., the Gaussian distribution D_r whose support is restricted to x such that $|x| < r\sqrt{n}$. The truncated and non-truncated distributions are statistically close, and we drop the word "truncated" from now on. For $\beta > 0$, define $\overline{\Psi}_\beta$ to be the distribution on $\mathbb{Z}_q$ obtained by drawing $y \leftarrow D_\beta$ and outputting $\lfloor q \cdot y \rceil \pmod{q}$. We write $\mathsf{LWE}_{q,\beta}$ as an abbreviation for $\mathsf{LWE}_{q,\overline{\Psi}_\beta}$.

We also define the *discrete Gaussian distribution* $D_{\mathbb{Z}^m,r}$ over the integer lattice $\mathbb{Z}^m$, which assigns probability proportional to $\prod_{i \in [m]} D_r(e_i)$ to each $\mathbf{e} \in \mathbb{Z}^m$. It is possible to efficiently sample from $D_{\mathbb{Z}^m,r}$ for any $r > 0$ [10].

Evidence for the hardness of $\mathsf{LWE}_{q,\beta}$ follows from results of Regev [23], who gave a *quantum* reduction from approximating certain problems on n-dimensional lattices in the worst case to within $\widetilde{O}(n/\beta)$ factors to solving $\mathsf{LWE}_{q,\beta}$

for dimension n, subject to the condition that $\beta \cdot q \geq 2\sqrt{n}$. Recently, Peikert [20] also gave a related *classical* reduction for similar parameters. For our purposes, we note that the $\mathsf{LWE}_{q,\beta}$ problem is believed to be hard (given the state-of-the-art in lattice algorithms) for any polynomial q and inverse-polynomial β (subject to the above condition).

Matrix notation for LWE. In this paper, we view all our vectors as column vectors. At times, we find it convenient to describe the LWE problem $\mathsf{LWE}_{q,\beta}$ using a compact matrix notation: find $\mathbf{s}$ given $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{x})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$ is chosen uniformly and $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$. We also use similar notation for the decision version $\mathsf{distLWE}$.

Connection to lattices. The LWE problem can be thought of as a "bounded-distance decoding problem" on a particular kind of *m-dimensional* lattice defined by the matrix $\mathbf{A}$. Specifically, define the lattice

$$\Lambda(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \ : \ \exists \mathbf{s} \in \mathbb{Z}^n \text{ s.t. } \mathbf{y} \equiv \mathbf{A}^T\mathbf{s} \pmod{q}\}.$$

The LWE problem can then be restated as: given $\mathbf{y}$ which is the sum of a lattice point $\mathbf{A}\mathbf{s}$ and a short "noise vector" $\mathbf{x}$, find the "closest" lattice vector $\mathbf{s}$. One can show that as long as $\mathbf{x}$ is short (say, $||\mathbf{x}|| < q/16$), there is a unique closest vector to $\mathbf{y}$ (see, e.g., [10]).

4.1 Some Supporting Lemmas

We present two technical lemmas regarding the LWE problem that will be used to prove smoothness of our (approximate) SPH systems in Sections 5.2 and 6.2.

If $m \geq n \log q$, the lattice $\Lambda(\mathbf{A})$ is quite sparse. In fact, we expect most vectors $\mathbf{z} \in \mathbb{Z}_q^m$ to be far from $\Lambda(\mathbf{A})$. The first lemma (originally shown in [23]) formalizes this intuition.

Let $\mathsf{dist}(\mathbf{z}, \Lambda(\mathbf{A}))$ denote the distance of the vector $\mathbf{z}$ from the lattice $\Lambda(\mathbf{A})$. The lemma shows that for most matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the fraction of vectors $\mathbf{z} \in \mathbb{Z}_q^m$ that are "very close" to $\Lambda(\mathbf{B})$ is "very small". The proof is by probabilistic method, and appears in the full version.

Lemma 1. *Let n, q, m be integers such that $m \geq n \log q$. For all but a negligible fraction of matrices $\mathbf{A}$,*

$$\Pr_{\mathbf{z} \leftarrow \mathbb{Z}_q^m}[\mathsf{dist}(\mathbf{z}, \Lambda(\mathbf{A})) \leq \sqrt{q}/4] \leq q^{-(m+n)/2}.$$

Fix a number $r > 0$, and let $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ be drawn from the discrete Gaussian distribution over the integer lattice $\mathbb{Z}^m$. If the vector $\mathbf{z}$ is (close to) a linear combination of the columns of $\mathbf{A}$, then given $\mathbf{e}^T\mathbf{A}$ one can (approximately) predict $\mathbf{e}^T\mathbf{z}$. The second lemma shows a converse of this statement when r is large enough. Namely, it says that if $\mathbf{z}$ and all its non-zero multiples are far from the lattice $\Lambda(\mathbf{A})$, then $\mathbf{e}^T\mathbf{A}$ does not give any information about $\mathbf{e}^T\mathbf{z}$. In other words, given $\mathbf{e}^T\mathbf{A}$ (where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ for a large enough r) $\mathbf{e}^T\mathbf{z}$ is statistically close to random. This lemma was first shown in [10], and was used in the construction of an oblivious transfer protocol in [21].

More formally, for a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and a vector $\mathbf{z} \in \mathbb{Z}_q^m$, let $\Delta_r(\mathbf{A}, \mathbf{z})$ denote the statistical distance between the uniform distribution on $\mathbb{Z}_q^{n+1}$ and the distribution of $(\mathbf{e}^T\mathbf{A}, \mathbf{e}^T\mathbf{z})$, where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$. Then,

Lemma 2. [10, Lemma 6.3] *Let $r \geq \sqrt{q} \cdot \omega(\sqrt{\log n})$. Then for most matrices $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, the following is true: if $\mathbf{z} \in \mathbb{Z}_q^m$ is such that for all non-zero $a \in \mathbb{Z}_q$, $\mathsf{dist}(a\mathbf{z}, \Lambda(\mathbf{A})) \geq \sqrt{q}/4$, then $\Delta_r(\mathbf{A}, \mathbf{z}) \leq \mathsf{negl}(n)$.*

5 Approximate Smooth Projective Hashing from Lattices

As a warmup to our main result we first construct a CPA-secure encryption scheme with an approximate SPH system. The main ideas in our final construction are already present here.

5.1 A CPA-Secure Encryption Scheme

The encryption scheme we use is a variant of the scheme presented in [10, 20], and is based on the hardness of the LWE problem. We stress that the novelty of this work is in constructing an approximate SPH system for this scheme.

We begin by describing a basic encryption scheme having decryption time exponential in the message length.[2] We then modify the scheme so that decryption can be done in polynomial time.

The message space is $\mathbb{Z}_q^\ell$ for some integers q, ℓ. In the basic encryption scheme, the public key consists of a matrix $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$, along with $\ell + 1$ vectors $\mathbf{u}_0, \ldots, \mathbf{u}_\ell \in \mathbb{Z}_q^m$. To encrypt a message $\mathbf{w} = (w_1, \ldots, w_\ell) \in \mathbb{Z}_q^\ell$ the sender chooses a uniformly random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and an error vector $\mathbf{x} \leftarrow \overline{\Psi}_\beta^m$. The ciphertext is

$$\mathbf{y} = \mathbf{B}\mathbf{s} + \big(\mathbf{u}_0 + \sum_{i=1}^{\ell} w_i \cdot \mathbf{u}_i\big) + \mathbf{x} \qquad \in \mathbb{Z}_q^m$$

The scheme is CPA-secure, since the $\mathsf{distLWE}_{q,\beta}$ assumption implies that the ciphertext is pseudorandom.

The ciphertext produced by the encryption algorithm is a vector $\mathbf{y}$ such that $\mathbf{y} - \big(\mathbf{u}_0 + \sum_{i=1}^{\ell} w_i \cdot \mathbf{u}_i\big)$ is "close" to the lattice $\Lambda(\mathbf{B})$ (the exact definition of "close" depends on the error parameter β). Decrypting a ciphertext is done by finding (via exhaustive search over the message space) a message $\mathbf{w}$ for which $\mathbf{y} - \big(\mathbf{u}_0 + \sum_{i=1}^{\ell} w_i \cdot \mathbf{u}_i\big)$ is "close" to $\Lambda(\mathbf{B})$, using the following trapdoor structure first discovered by Ajtai [1], and later improved by Alwen and Peikert [2].

[2] Interestingly, for our eventual application to PAKE a CCA-secure version of this scheme would suffice since the scheme has the property that it *is* possible to efficiently tell whether a given ciphertext is an encryption of a given message (and this is all that is needed to prove security for the protocol in Section 3).

Lemma 3 ([1, 2]). *Fix integers $q \geq 2$ and $m \geq 4n \log^2 q$. There is a* PPT *algorithm* $\mathsf{TrapSamp}(1^n, q, m)$ *that outputs matrices* $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$ *and* $\mathbf{T} \in \mathbb{Z}^{m\times m}$ *such that the distribution of* $\mathbf{B}$ *is statistically close to the uniform distribution over* $\mathbb{Z}_q^{m\times n}$*, and there is an algorithm* $\mathsf{BDDSolve}(\mathbf{T}, \cdot)$ *that takes as input a vector* $\mathbf{z} \in \mathbb{Z}^m$ *and does the following:*

– *if there is a vector* $\mathbf{s} \in \mathbb{Z}^m$ *such that* $\mathsf{dist}(\mathbf{z}, \mathbf{Bs} \pmod q) \leq \sqrt{q}/4$*, then the output is* $\mathbf{s}$*.*
– *if for every vector* $\mathbf{s} \in \mathbb{Z}^m$*,* $\mathsf{dist}(\mathbf{z}, \mathbf{Bs}) > \sqrt{q}/4$*, then the output is* $\perp$*.*

Proof. $\mathbf{T}$ is a full-rank matrix such that (a) each row of $\mathbf{t}_i$ has bounded ℓ_2 norm, i.e., $||\mathbf{t}_i|| \leq 4\sqrt{m}$, and (b) $\mathbf{TB} = 0 \pmod q$. [1, 2] showed how to sample a pair $(\mathbf{B}, \mathbf{T})$ such that $\mathbf{B}$ is statistically close to uniform and $\mathbf{T}$ has the above properties.

Given such a matrix $\mathbf{T}$ and a vector $\mathbf{z} \in \mathbb{Z}^m$, $\mathsf{BDDSolve}(\mathbf{T}, \mathbf{z})$ works as follows:

– first, compute $\mathbf{z}' = q \cdot \mathbf{T}^{-1} \cdot \lfloor (\mathbf{T} \cdot \mathbf{z})/q \rceil \pmod q$.
– Compute (using Gaussian elimination) a vector $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\mathbf{z}' = \mathbf{Bs}$ (if such exists; else, output $\perp$).
– If $\mathsf{dist}(\mathbf{z}, \mathbf{Bs}) \leq \sqrt{q}/4$, then output $\mathbf{s}$ else output $\perp$.

First, if $\mathbf{z} = \mathbf{Bs} + \mathbf{x}$ for some $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{x} \in \mathbb{Z}_q^m$ such that $||\mathbf{x}|| \leq \sqrt{q}/4$, then the procedure above computes

$$\mathbf{z}' = q \cdot \mathbf{T}^{-1} \cdot \lfloor (\mathbf{T} \cdot (\mathbf{Bs} + \mathbf{x}))/q \rceil \pmod q = \mathbf{Bs} \pmod q$$

This is because each co-ordinate of $\mathbf{Tx}$ has magnitude at most $||\mathbf{T}|| \cdot ||\mathbf{x}|| \leq 4\sqrt{m} \cdot \sqrt{q}/4 \ll q$, and consequently,

$$\lfloor (\mathbf{T} \cdot (\mathbf{Bs} + \mathbf{x}))/q \rceil = \lfloor (\mathbf{T} \cdot \mathbf{Bs})/q \rceil = \mathbf{T} \cdot (\mathbf{Bs})/q$$

where the final equality is because $\mathbf{TB} = 0 \pmod q$.

Finally, if $\mathsf{dist}(\mathbf{z}, \Lambda(\mathbf{B})) > \sqrt{q}/4$, then the last line of the procedure above causes the output to be $\perp$ always. $\square$

We now modify the decryption algorithm in two ways. The first of these modifications ensures that the decryption algorithm runs in polynomial time, and the second is needed for our approximate SPH system.

First, to avoid the exponential dependence of the decryption time on the message length, we modify the encryption scheme by letting the public key contain the matrix $\mathbf{A} = [\mathbf{B}|\mathbf{U}]$, where the columns of $\mathbf{U} \in \mathbb{Z}_q^{m\times(\ell+1)}$ are the vectors $\mathbf{u}_0, \ldots, \mathbf{u}_\ell$. The secret-key is a trapdoor for the entire matrix $\mathbf{A}$ (as opposed to just $\mathbf{B}$ as in the previous description). The ciphertext from the previous description can then be written as

$$\mathbf{y} = \mathbf{A}^T \begin{pmatrix} \mathbf{s} \\ 1 \\ \mathbf{w} \end{pmatrix} + \mathbf{x} \in \mathbb{Z}_q^m$$

and decryption uses the $\mathsf{BDDSolve}$ procedure from Lemma 3 to recover the vector $(\mathbf{s}, 1, \mathbf{m})$. The crucial point is that, during key generation, the receiver can generate the matrix $\mathbf{A}$ along with an appropriate trapdoor for decryption.

Secondly, we relax the decryption algorithm so that it finds an $a \in \mathbb{Z}_q$ and a message $\mathbf{w}$ for which $a\big(\mathbf{y} - (\mathbf{u}_0 + \sum_{i=1}^{\ell} w_i \cdot \mathbf{u}_i)\big)$ is "close" to $\Lambda(\mathbf{B})$. This modified decryption algorithm correctly decrypts the ciphertexts generated by Enc (which corresponds to the case $a = 1$), but it also decrypts ciphertexts that would never be output by Enc. This modification to the decryption algorithm enables us to prove smoothness for the approximate SPH system.

Parameters. Let n be the security parameter, and $\ell = n$ be the message length. The parameters of the system are a prime $q = q(n, \ell)$, a positive integer $m = m(n, \ell)$, and a Gaussian error parameter $\beta = \beta(n, \ell) \in (0, 1]$ that defines a distribution $\overline{\Psi}_\beta$. For concrete instantiations of these parameters, see Theorem 1.

We now describe the scheme:

Key generation. Choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times(n+\ell+1)}$ together with the trapdoor $\mathbf{T}$ by running $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapSamp}(1^m, 1^{n+\ell+1}, q)$, where $\mathsf{TrapSamp}$ is as described in Lemma 3. Let the public key be $\mathbf{A}$ and the secret-key is $\mathbf{T}$.

Encryption. To encrypt the message $\mathbf{w} \in \mathbb{Z}_q^{\ell}$ with respect to a public key as above, the sender chooses $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random, and an error vector $\mathbf{x} \leftarrow \overline{\Psi}_\beta^{mn}$. The ciphertext is

$$\mathbf{y} = \mathbf{A} \cdot \begin{pmatrix} \mathbf{s} \\ 1 \\ \mathbf{w} \end{pmatrix} + \mathbf{x} \pmod q$$

Decryption. The decryption algorithm works as below.

```
**for** $a = 1$ *to* $q - 1$ **do**
    Compute $\begin{pmatrix} \mathbf{s} \\ a' \\ \mathbf{w} \end{pmatrix} \leftarrow \mathsf{BDDSolve}(\mathbf{T}, a\mathbf{y})$
    **if** $a' = a$ **then**
        output $\mathbf{w}/a$ and stop
    **else** try the next value of $a$
**end**
If the above fails for all $a$, output $\perp$
```

Theorem 1. *Let n, ℓ, m, q, β be chosen such that $m \geq 4(n + \ell)\log q$ and $\beta < 1/(2 \cdot m^2 n \cdot \omega(\sqrt{\log n}))$. Then the scheme above is a CCA-secure encryption scheme assuming the hardness of* $\mathsf{distLWE}_{n,m,q,\beta}$.

5.2 An Approximate SPH System

Fix a public key $\mathbf{A} \in \mathbb{Z}_q^{m\times(n+\ell+1)}$ for the system (where we write $\mathbf{A} = [\mathbf{B}|\mathbf{U}]$, as usual), and a dictionary $\mathcal{D} \overset{\text{def}}{=} \mathbb{Z}_q^{\ell}$. Sets X, $\overline{L}_{\mathbf{m}}$ and $L_{\mathbf{m}}$ are defined in Section 2.

(For our purposes, all vectors $\mathbf{y} \in \mathbb{Z}_q^m$ are valid ciphertexts). Let r be such that

$$\sqrt{q} \cdot \omega(\sqrt{\log n}) \leq r \leq \varepsilon/(8 \cdot mn^2 \cdot \beta).$$

(Looking ahead, we remark that the upper bound on r will be used for correctness, and the lower bound will be used for smoothness.)

A key for the SPH system is a k-tuple of vectors $(\mathbf{e}_1, \ldots, \mathbf{e}_k)$ where each $\mathbf{e}_i \leftarrow D_{\mathbb{Z}^m,r}$ is drawn independently from the discrete Gaussian distribution. The reader may want to keep in mind the inverse relationship between the parameters r and β: the larger the error parameter β in the encryption scheme, the smaller the discrete-Gaussian radius r (and vice versa).

1. The projection set $S \overset{\text{def}}{=} (\mathbb{Z}_q^n)^k$. For a key $(\mathbf{e}_1, \ldots, \mathbf{e}_k) \in (\mathbb{Z}_q^m)^k$, the projection is $\alpha(\mathbf{e}_1, \ldots, \mathbf{e}_k) = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$, where $\mathbf{u}_i = \mathbf{B}^T \mathbf{e}_i$.
2. We now define the smooth projective hash function $\mathcal{H} = \{H_k\}_{k \in K}$. On input a key $(\mathbf{e}_1, \ldots, \mathbf{e}_k) \in K$ and a ciphertext $\mathbf{c} = (\mathsf{label}, \mathbf{y}, \mathbf{m})$, the hash function is computed as follows. First compute
$$z_i = \mathbf{e}_i^T \left[\mathbf{y} - \mathbf{U} \cdot \begin{pmatrix} 1 \\ \mathbf{m} \end{pmatrix} \right] \in \mathbb{Z}_q.$$
Treat z_i as a number in $[-(q-1)/2 \ldots (q-1)/2]$ and output $b_1 \ldots b_k \in \{0,1\}^k$ where
$$b_i = \begin{cases} 0 \text{ if } z_i < 0 \\ 1 \text{ if } z_i > 0 \end{cases}.$$
3. On input a projected key $(\mathbf{u}_1, \ldots, \mathbf{u}_k) \in S$, a ciphertext $\mathbf{c} = (\mathsf{label}, \mathbf{y}, \mathbf{m})$ and a witness $\mathbf{s} \in \mathbb{Z}_q^n$ for the ciphertext, the hash function is computed as $H'_{\mathbf{u}}(\mathbf{c}, \mathbf{s}) = b_1 \ldots b_k$ where
$$b_i = \begin{cases} 0 \text{ if } \mathbf{u}_i^T \mathbf{s} < 0 \\ 1 \text{ if } \mathbf{u}_i^T \mathbf{s} > 0 \end{cases}.$$

Theorem 2. *Let the parameters n, ℓ, m, q, β be as in Theorem 1, and r be as above. Then, $\mathcal{H} = \{H_k\}_{k \in K}$ is an ε-approximate smooth projective hash system.*

Proof. Clearly, the following procedures can all be done in polynomial time: (1) sampling a uniform key for the hash function $(\mathbf{e}_1, \ldots, \mathbf{e}_k) \leftarrow (D_{\mathbb{Z}^m,r})^k$, (2) computing the hash function H on input the key $(\mathbf{e}_1, \ldots, \mathbf{e}_k)$ and a ciphertext $\mathbf{c}$, (3) computing the projection-key $\alpha(\mathbf{e}_1, \ldots, \mathbf{e}_k)$, and (4) computing the hash function given the projected key $(\mathbf{u}_1, \ldots, \mathbf{u}_k)$, a ciphertext $\mathbf{c}$, and a witness $\mathbf{s}$ for the ciphertext $\mathbf{c}$.

Approximate correctness. We now show ε-approximate correctness. Consider any $(\mathsf{label}, \mathbf{y}, \mathbf{m}) \in \overline{L}$, i.e., where $\mathbf{y}$ is a ciphertext produced by the encryption algorithm on input the message $\mathbf{m}$. This means that $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{U} \cdot \begin{pmatrix} 1 \\ \mathbf{m} \end{pmatrix} + \mathbf{x} \pmod{q} \tag{1}$$

where $||\mathbf{x}|| \leq \beta q \cdot \sqrt{mn}$ (recall we work with truncated Gaussians).

We first show that for each $i \in [k]$, the values z_i (computed using the key) and $\mathbf{s}^T \mathbf{u}_i$ (computed using the projected key) are "close". More precisely, we show that $|z_i - \mathbf{u}_i^T \mathbf{s}| \leq \varepsilon/2 \cdot (q/4)$. This follows because

$$|z_i - \mathbf{u}_i^T \mathbf{s}| = |(\mathbf{e}_i^T(\mathbf{B}\mathbf{s} + \mathbf{x}) - \mathbf{u}_i^T \mathbf{s}| = |\mathbf{e}_i^T \mathbf{x}|, \tag{2}$$

where the first equality uses the fact that $\mathbf{y}$ can be written as in Equation (1), and the second uses the fact that $\mathbf{u}_i = \mathbf{e}_i^T \mathbf{B}$. Now, $|\mathbf{e}_i^T \mathbf{x}| \leq ||\mathbf{e}_i|| \cdot ||\mathbf{x}|| \leq (r\sqrt{mn}) \cdot (\beta q \sqrt{mn}) < \varepsilon/2 \cdot q/4$.

Each $\mathbf{u}_i$ is statistically close to uniform, by an application of the leftover hash lemma; in particular, this means that $\mathbf{s}^T \mathbf{u}_i \in \mathbb{Z}_q$ is uniformly random.[3] Let b_i be the i^{th} bit of $H_{(\mathbf{e}_1,\ldots,\mathbf{e}_k)}(\mathbf{c})$ and b'_i be the i^{th} bit of $H'_{(\mathbf{u}_1,\ldots,\mathbf{u}_k)}(\mathbf{c}, \mathbf{s})$. Using Equation (2), we see that the probability that $b_i \neq b'_i$ (over the randomness of $\mathbf{e}_i$) is at most $\varepsilon/2$. Thus, by a Chernoff bound, the Hamming distance between $H_{(\mathbf{e}_1,\ldots,\mathbf{e}_k)}(\mathbf{c})$ and $H'_{(\mathbf{u}_1,\ldots,\mathbf{u}_k)}(\mathbf{c}, \mathbf{s})$ is at most εk with overwhelming probability. This shows approximate correctness.

Smoothness. Consider any $(\mathsf{label}, \mathbf{y}, \mathbf{m}) \in X \setminus L$. By definition of L, this means that the decryption algorithm, on input $(\mathsf{label}, \mathbf{y}, \mathbf{m})$ and *any* possible secret key sk, does not output $\mathbf{m}$. In other words, the decryption algorithm outputs either $\perp$, or a message $\mathbf{m}' \neq \mathbf{m}$. Define

$$\mathbf{z} := \mathbf{y} - \mathbf{U} \cdot \begin{pmatrix} 1 \\ \mathbf{m} \end{pmatrix} \text{ and } \mathbf{z}' := \mathbf{y} - \mathbf{U} \cdot \begin{pmatrix} 1 \\ \mathbf{m}' \end{pmatrix}.$$

We will show that for every non-zero $a \in \mathbb{Z}_q$, $a\mathbf{z}$ is far from the $\Lambda(\mathbf{B})$. More precisely, we will show that for every non-zero $a \in \mathbb{Z}_q$,

$$\mathsf{dist}(a\mathbf{z}, \Lambda(\mathbf{B})) \geq \sqrt{q}/4.$$

An application of Lemma 2 then shows that for every $i \in [k]$, the pair $(\mathbf{e}_i^T \mathbf{B}, \mathbf{e}_i^T \mathbf{z})$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n+1}$.

Let us analyze the two cases:

- The output of the decryption algorithm is $\perp$. In particular, this means that for every $a \in [1 \ldots q-1]$, the vector $a\mathbf{z}$ is far from $\Lambda(\mathbf{B})$.
- The output of the decryption algorithm is a message $\mathbf{m}' \neq \mathbf{m}$. This could happen only if there is an $a' \in \mathbb{Z}_q$ such that $a'\mathbf{z}'$ is close to the lattice $\Lambda(\mathbf{B})$. Suppose, for contradiction, that $a\mathbf{z}$ is close to $\Lambda(\mathbf{B})$ as well. The claim below shows that this cannot happen with high probability over the random choice of $\mathbf{U}$. Thus, with high probability, $a\mathbf{z}$ is far from $\Lambda(\mathbf{B})$.

  Claim. The following event happens with negligible probability over the uniformly random choice of $\mathbf{U} \in \mathbb{Z}_q^{m \times \ell}$: there exist numbers $a, a' \in \mathbb{Z}_q$, vectors $\mathbf{m} \neq \mathbf{m}' \in \mathbb{Z}_q^{\ell}$ and a vector $\mathbf{y} \in \mathbb{Z}_q^m$ s.t.

$$\mathsf{dist}(a\mathbf{z}, \Lambda(\mathbf{B})) \leq \sqrt{q}/4 \text{ and } \mathsf{dist}(a'\mathbf{z}', \Lambda(\mathbf{B})) \leq \sqrt{q}/4.$$

[3] This holds only for $\mathbf{s} \neq \mathbf{0}$. We omit consideration of this technical issue for the purposes of this paper.

Proof. Fix some $a, a' \in \mathbb{Z}_q$, $\mathbf{m} \neq \mathbf{m}' \in \mathbb{Z}_q^\ell$ and $\mathbf{y} \in \mathbb{Z}_q^m$. We first observe that since the vectors $\begin{pmatrix} 1 \\ \mathbf{m} \end{pmatrix}$ and $\begin{pmatrix} 1 \\ \mathbf{m}' \end{pmatrix}$ are linearly independent and $\mathbf{U}$ is uniformly random, the vectors $a\mathbf{z}$ and $a'\mathbf{z}'$ are uniformly random and (statistically) independent. Applying Lemma 1, we get that

$$\begin{aligned}\Pr_{\mathbf{U} \in \mathbb{Z}_q^{m\times\ell}}[\mathsf{dist}(a\mathbf{z}, \Lambda(\mathbf{B}))\sqrt{q}/4 \text{ and } \mathsf{dist}(a'\mathbf{z}', \Lambda(\mathbf{B})) \leq \sqrt{q}/4] \\ \leq (q^{-m/2} \cdot \mathsf{negl}(n))^2 = q^{-m} \cdot \mathsf{negl}(n).\end{aligned}$$

Now, an application of union bound shows that the required probability is at most $q^2 \cdot q^{2\ell} \cdot q^m \cdot (q^{-m} \cdot \mathsf{negl}(n))$, which is negligible in n. □

This completes the proof of Theorem 2. □

6 A CCA-Secure Encryption Scheme based on Lattices

In this section we describe a CCA-secure encryption scheme, along with an approximate SPH system, based on the hardness of the LWE problem. The CCA-secure encryption scheme builds on the CPA-secure encryption scheme described in Section 5.1, and the SPH system is the same as the one from Section 5.2 with a few modifications.

6.1 A CCA-Secure Encryption Scheme

The encryption scheme is similar to the schemes in [20, 12] (which, themselves, are instantiations of the general construction of Rosen and Segev [24]). The main difference between [20, 12] and our scheme is the relaxed notion of decryption, which we already use in the CPA-secure construction in Section 5.1. A formal description of the scheme follows.

Parameters. Let n be the security parameter, and $\ell = \mathsf{poly}(n)$ be the message length. The parameters of the system are a prime $q = q(n, \ell)$, an integer $m = m(n, \ell) \in \mathbb{Z}^+$, and a Gaussian error parameter $\beta = \beta(n, \ell) \in (0, 1]$ that defines a distribution $\overline{\Psi}_\beta$. For concrete instantiations of these parameters, see Theorem 3.

Key generation. For $i \in [n]$ and $b \in \{0, 1\}$, choose $2n$ matrices $\mathbf{A}_{i,b} \leftarrow \mathbb{Z}_q^{m\times(n+\ell+1)}$ together with short bases $\mathbf{S}_{i,b} \in \mathbb{Z}^{m\times m}$ for $\Lambda^\perp(\mathbf{A}_{i,b})$. More precisely, let

$$(\mathbf{A}_{i,b}, \mathbf{S}_{i,b}) \leftarrow \mathsf{TrapSamp}(1^m, 1^{n+\ell+1}, q),$$

where $\mathsf{TrapSamp}$ is as described in Lemma 3. Output the public and secret keys

$$pk = \{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i\in[n]} \text{ and } sk = \{\mathbf{S}_{1,0}, \mathbf{S}_{1,1}\}.$$

(Note that the receiver does not use the trapdoors for $i > 1$ and so the $\{\mathbf{A}_{i,b}\}_{i>1}$ could, in fact, simply be chosen at random.)

Encryption. To encrypt the message $\mathbf{w} \in \mathbb{Z}_q^\ell$ with respect to a public key as above, the sender first generates a key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathsf{SigKeyGen}(1^n)$ for a one-time signature scheme; let $\mathsf{VK} = \mathsf{VK}_1, \ldots, \mathsf{VK}_n$ denote the bits of the verification key. Define the matrix $\mathbf{A}_{\mathsf{VK}}$ as

$$\mathbf{A}_{\mathsf{VK}} = \begin{bmatrix} \mathbf{A}_{1,\mathsf{VK}_1} \\ \vdots \\ \mathbf{A}_{n,\mathsf{VK}_n} \end{bmatrix}.$$

Choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random, and choose an error vector $\mathbf{x} \leftarrow \overline{\Psi}_\beta^{mn}$. The ciphertext is $(VK, \mathbf{y}, \sigma)$ where

$$\mathbf{y} = \mathbf{A}_{\mathsf{VK}} \cdot \begin{pmatrix} \mathbf{s} \\ 1 \\ \mathbf{w} \end{pmatrix} + \mathbf{x} \pmod{q}$$

and $\sigma = \mathsf{Sign}_{\mathsf{SK}}(\mathbf{y})$.

Decryption. To decrypt a ciphertext $(\mathsf{VK}, \mathbf{y}, \sigma)$, first verify that σ is a correct signature on $\mathbf{y}$ and output $\perp$ if not. Otherwise, parse $\mathbf{y}$ into n consecutive blocks $\mathbf{y}_1, \ldots, \mathbf{y}_n$, where $\mathbf{y}_i \in \mathbb{Z}_q^m$. Then,

```
for $a = 1$ to $q - 1$ do
    Compute $\mathbf{t} := \begin{pmatrix} \mathbf{s} \\ a' \\ \mathbf{w} \end{pmatrix} \leftarrow \mathsf{BDDSolve}(\mathbf{T}_{1,\mathsf{VK}_1}, a\mathbf{y})$
    if $a' = a$ then
        if $||\mathbf{A}_{i,\mathsf{VK}_i} \cdot \mathbf{t} - a\mathbf{y}_i|| \leq \sqrt{q}/4$ for all $i \in [n]$ then
            output $\mathbf{w}/a$ and stop
    else try the next value of $a$
end
If the above fails for all $a$, output $\perp$
```

Theorem 3. *Let n, ℓ, m, q, β be such that $m \geq 4(n + \ell)\log^2 q$ and $\beta < 1/(2 \cdot m^2 n \cdot \omega(\sqrt{\log n}))$. Then, the scheme above is a CCA-secure encryption scheme assuming the hardness of $\mathsf{distLWE}_{n,m,q,\beta}$.*

The proof of correctness is similar to that of the CPA-secure encryption scheme. CCA-security follows from the ideas of [20, 12]. As we observed, the main change between our encryption scheme and the one in [20, 12] is that the decryption algorithm tries to decrypt "all multiples of the ciphertext". We defer the details of the proof to the full version.

6.2 An Approximate SPH System

Fix a public key $\{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i \in [n]}$, and a password dictionary $\mathcal{D} \stackrel{\text{def}}{=} \mathbb{Z}_q^\ell$. The main difference from the presentation in Section 5.2 is in the definition of ciphertext validity: now, a labeled ciphertext $(\mathsf{label}, \mathsf{VK}, \mathbf{y}, \sigma)$ is defined to be valid if $\mathsf{Verify}_{\mathsf{VK}}(\mathsf{label}||\mathbf{y},\sigma) = \mathsf{accept}$. Clearly, all honestly generated ciphertexts are valid and this condition can be checked in polynomial time. We define the sets X, $\overline{L}_{\mathbf{m}}$, and $L_{\mathbf{m}}$ for $\mathbf{m} \in \mathcal{D}$ exactly as in Section 2.

As in Section 5.2, a hash key is a k-tuple of vectors $(\mathbf{e}_1,\ldots,\mathbf{e}_k)$ where each $\mathbf{e}_i \leftarrow D_{\mathbb{Z}^m,r}$ is drawn independently from the discrete Gaussian distribution. The projection function and the hash computation are the same, except that here they use the matrices $\mathbf{B}_{\mathsf{VK}}$ and $\mathbf{U}_{\mathsf{VK}}$ respectively (instead of $\mathbf{B}$ and $\mathbf{U}$ in Section 5.2). In particular, this means that the projection function depends on the ciphertext (as allowed by the definition of an approximate SPH). The proof of the theorem below follows analogously to that of Theorem 2; we defer the proof to the full version of this paper.

Theorem 4. *Let $m \geq 4(n+\ell)\log q$, $\beta < 1/(2\cdot m^2 n \cdot \omega(\sqrt{\log n}))$ and r be such that*

$$\sqrt{q}\cdot\omega(\sqrt{\log n}) \leq r \leq \varepsilon/(8\cdot mn^2\cdot\beta).$$

Then $\mathcal{H} = \{H_k\}_{k\in K}$ is an ε-approximate smooth projective hash system.

References

1. M. Ajtai. Generating hard instances of the short basis problem. In *26th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
2. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 75–86. Schloss Dagstuhl, 2009. Available at `http://drops.dagstuhl.de/portals/STACS09/`.
3. M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In *Advances in Cryptology — Eurocrypt 2000*, volume 1807 of *LNCS*, pages 139–155. Springer, 2000.
4. S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *IEEE Symposium on Security & Privacy*, pages 72–84. IEEE, 1992.
5. V. Boyko, P. D. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Advances in Cryptology — Eurocrypt 2000*, volume 1807 of *LNCS*, pages 156–171. Springer, 2000.
6. R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. D. MacKenzie. Universally composable password-based key exchange. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 404–421. Springer, 2005.
7. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology — Eurocrypt 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.
8. R. Gennaro. Faster and shorter password-authenticated key exchange. In *5th Theory of Cryptography Conference — TCC 2008*, volume 4948 of *LNCS*, pages 589–606. Springer, 2008.
9. R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. *ACM Trans. Information and System Security*, 9(2):181–234, 2006.
10. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 197–206. ACM Press, 2008.

11. O. Goldreich and Y. Lindell. Session-key generation using human passwords only. *Journal of Cryptology*, 19(3):241–340, 2006.
12. S. Goldwasser and V. Vaikuntanathan. Correlation-secure trapdoor functions and CCA-secure encryption from lattices, 2009. Manuscript.
13. L. Gong, T. M. A. Lomas, R. M. Needham, and J. H. Saltzer. Protecting poorly chosen secrets from guessing attacks. *IEEE J. Selected Areas in Communications*, 11(5):648–656, 1993.
14. S. Halevi and H. Krawczyk. Public-key cryptography and password protocols. *ACM Trans. Information and System Security*, 2(3):230–268, 1999.
15. S. Jiang and G. Gong. Password based key exchange with mutual authentication. In *11th Annual International Workshop on Selected Areas in Cryptography (SAC)*, volume 3357 of *LNCS*, pages 267–279. Springer, 2004.
16. J. Katz, P. D. MacKenzie, G. Taban, and V. D. Gligor. Two-server password-only authenticated key exchange. In *3rd Intl. Conference on Applied Cryptography and Network Security (ACNS)*, volume 3531 of *LNCS*, pages 1–16. Springer, 2005.
17. J. Katz, R. Ostrovsky, and M. Yung. Efficient password-authenticated key exchange using human-memorable passwords. In *Advances in Cryptology — Eurocrypt 2001*, volume 2045 of *LNCS*, pages 475–494. Springer, 2001.
18. P. D. MacKenzie, S. Patel, and R. Swaminathan. Password-authenticated key exchange based on RSA. In *Advances in Cryptology – Asiacrypt 2000*, volume 1976 of *LNCS*, pages 599–613. Springer, 2000.
19. M.-H. Nguyen and S. Vadhan. Simpler session-key generation from short random passwords. *Journal of Cryptology*, 21(1):52–96, 2008.
20. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 333–342. ACM Press, 2009.
21. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008. A full version, containing additional results, is avalable at `http://eprint.iacr.org/2007/348.pdf`.
22. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 187–196. ACM Press, May 2008.
23. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 84–93. ACM Press, 2005.
24. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *6th Theory of Cryptography Conference — TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.