Code-based public-key encryption resistant to key leakage [⋆]

Edoardo Persichetti

University of Warsaw

Abstract. Side-channel attacks are a major issue for implementation of secure cryptographic schemes. Among these, key-leakage attacks describe a scenario in which an adversary is allowed to learn arbitrary information about the private key, the only constraint being the number of bits learned. In this work, we study key-leakage resilience according to the model presented by Akavia, Goldwasser and Vaikuntanathan at TCC '09. As our main contribution, we present a code-based hash proof system; we obtain our construction by relaxing some of the requirements from the original definition of Cramer and Shoup. We then propose a leakage-resilient public-key encryption scheme that makes use of this hash proof system. To do so, we adapt a framework featured in a previous work by Alwen et al. regarding identity-based encryption (EUROCRYPT '10). Our construction features error-correcting codes as a technical tool, and, as opposed to previous work, does not require the use of a randomness extractor.

1 Introduction

Traditionally, the security of cryptographic schemes is analyzed with respect to an idealized, abstract adversarial model [14]. Unfortunately, in the real world, implementations of cryptographic schemes are often vulnerable to an additional kind of threat, of a more physical nature. These are the so-called *side-channel* attacks, which are based on the observation of phenomena directly connected with the implementation, such as power or timing measurements, detection of internal faults and leakage of some private information. It is therefore important to build schemes whose security can be argued even in presence of such attacks.

In this work, we focus on one particular type of attacks, known as "cold boot" or memory attacks, first introduced by Halderman et al. [16] in 2008. The authors show how it is possible to recover private information stored in the device's memory after the device is turned off; this is because typical DRAM memories only lose their content during a gradual period

[⋆] European Research Council has provided financial support under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no CNTM-207908

of time. In particular, a significant fraction of the cryptographic key stored in the memory can be easily recovered, leading to potentially devastating attacks. We therefore speak about *key-leakage* attacks. A general framework that models key-leakage attacks was introduced by Akavia, Goldwasser and Vaikuntanathan [1]. In this model, the adversary is allowed the knowledge of arbitrarily chosen functions of the private key, with the restriction that the total output of these functions doesn't exceed a certain bound λ. Attacks can be performed, as usual, both in an adaptive or non-adaptive fashion. The authors then show that the lattice-based public-key encryption scheme of Regev [23] and the identity-based encryption scheme of Gentry, Peikert and Vaikuntanathan [13] are resistant to such leakage. The framework was subsequently revisited by Naor and Segev [21] and further generalized by Alwen, Dodis and Wichs [3], who provide the first public-key primitives in the *Bounded Retrieval Model (BRM)*. This model had been previously introduced by Dziembowski and Di Crescenzo in independent works [10,7] for symmetric encryption schemes.

To date, many cryptographic primitives have been shown to be resilient to key-leakage attacks, based on a variety of assumptions such as DDH or d-Linear, quadratic residuosity, composite residuosity and LWE; however, there is no known construction based on coding theory assumptions. Code-based cryptography is one of the candidates for "post-quantum" cryptography; compared to LWE-based schemes, code-based schemes have the advantage of a simpler structure (for example, they usually work over the binary field) thus allowing for more practical implementations.

Our contribution In this paper, we propose a protocol based solely on coding theory assumptions, that achieves semantic security against key-leakage attacks. The first step is to build a *Hash Proof System (HPS)*. This primitive, essentially a special kind of non-interactive zero-knowledge proof system for a language, was first introduced by Cramer and Shoup in [6] as a theoretical tool to construct efficient public-key encryption schemes. It was later shown by Kiltz et al. [20] that it is possible to view a HPS as a key encapsulation mechanism (KEM) with special properties, and that it is possible to obtain secure hybrid encryption schemes by using randomness extractors. The work of Naor and Segev [21] builds on this method, and the authors present a general framework to design leakage-resilient encryption schemes using any HPS together with a randomness extractor. This is also the basis for a recent paper by Alwen et al. [2] in which the framework is extended to the identity-based setting. Our construction works as follows. The private key has a high min-entropy that is guaranteed by analyzing the volume of spheres centered on codewords of a randomly generated code. This induces an error in the decapsulation procedure; however, we manage to bound the size of the error term by carefully choosing the scheme's parameters. This allows us to deal with the possible decryption error by using error-correcting codes in our encryption scheme. Finally, we achieve ciphertext indistinguishability thanks to the pseudorandomness of the syndrome construction (Fischer and Stern [11]) for low-weight error vectors. To the best of our knowledge, this is the first construction of a hash proof system from coding theory assumptions.

The paper is structured as follows: in Section 2 we give some preliminary definitions necessary for the remainder of the paper. Next, we describe hash proof systems (Section 3) and leakage-resilient public-key encryption (Section 4). Our construction of a code-based hash proof system is presented in Section 5, and the encryption scheme based on it in Section 6. We conclude in Section 7.

2 Preliminaries

We start by providing some probability notions that are relevant for the paper. The notation $x \leftarrow \chi_\rho$ means sampling an element x from the Bernoulli distribution with parameter ρ: this extends naturally to vectors and matrices with notation, respectively, χ_ρ^n and $\chi_\rho^{m \times n}$. The *statistical distance* between two random variables X and Y with values in Ω is defined as $\Delta(X,Y) = \frac{1}{2}\sum_{\omega \in \Omega} \Big| \mathsf{Pr}[X = \omega] - \mathsf{Pr}[Y = \omega] \Big|$. The *min-entropy* of a random variable X is $H_\infty(X) = -\log(\max_{\omega \in \Omega} \mathsf{Pr}[X = \omega])$. We also present the notion of *average min-entropy*, which is useful to describe the unpredictability of a random variable X conditioned on the value of another random variable Y, as $\tilde{H}_\infty(X|Y) = -\log \left(\mathbb{E}_{y \xleftarrow{\$} Y} \left[2^{-H_\infty(X|Y=y)} \right] \right)$.

Next, we present a few basic coding theory notions. Throughout the paper, we will treat all vectors as column vectors, and denote them by a boldface letter. An $[n,k]$ linear code over the finite field $\mathbb{F}_q$ is a vector subspace $\mathcal{C}$ of $\mathbb{F}_q^n$ of dimension k. A *generator matrix* for $\mathcal{C}$ is a matrix whose rows form a basis for the subspace. A *parity-check matrix* for $\mathcal{C}$ is an $(n-k) \times n$ matrix H such that $H\boldsymbol{x} = \mathbf{0}$ for all codewords $\boldsymbol{x}$. For every vector $\boldsymbol{x} \in \mathbb{F}_q^n$ the *Hamming weight* $\mathsf{wt}(\boldsymbol{x})$ is the number of its non-zero positions; $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathsf{wt}(\boldsymbol{x} - \boldsymbol{y})$ is the *Hamming distance* between the two words $\boldsymbol{x}$ and $\boldsymbol{y}$. The *minimum distance* of a code $\mathcal{C}$ is simply the minimum between the distance of all codewords of $\mathcal{C}$. The following is a very important bound on the minimum distance of linear codes.

Definition 1 (GV Bound). *Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. The* Gilbert-Varshamov (GV) Distance *is the largest integer d_0 such that*

$$|\mathcal{B}(\mathbf{0}, d_0 - 1)| \leq q^{n-k} \tag{1}$$

where $\mathcal{B}(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathbb{F}_q^n | \mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) \leq r\}$ is the n-dimensional ball of radius r centered in $\boldsymbol{x}$.

For an integer w below the GV bound, the following problem is hard (Berlekamp, McEliece and van Tilborg [4]).

Definition 2 (Syndrome Decoding Problem). *Given an $(n-k) \times n$ parity-check matrix H for an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$, a vector $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ and an integer $w \in \mathbb{N}^+$, find $\boldsymbol{e} \in \mathbb{F}_q^n$ such that $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}$ and $\mathsf{wt}(\boldsymbol{e}) \leq w$.*

3 Hash Proof Systems

Like in the majority of previous work, we use a HPS for our construction. To describe our HPS, we adapt the "simplified" definition of [2] to public-key encryption schemes.

Table 1: Hash Proof System.

Setup	The setup algorithm takes as input a security parameter θ and returns the public parameters of the scheme. The algorithm also defines the set K of encapsulated keys.
KeyGen	The key generation algorithm takes as input the public parameters and outputs a public key pk and a private key sk.
Encap	The *valid* encapsulation algorithm receives as input the public key pk and returns a ciphertext/key pair (ψ_0, K).
Encap*	The *invalid* encapsulation algorithm receives as input the public key pk and samples an invalid ciphertext ψ_0.
Decap	The decapsulation algorithm takes as input a private key sk and a ciphertext ψ_0 and outputs a key K'.

Note that all of the above algorithms are probabilistic, except Decap, which is deterministic. There are two important requirements on the output of Decap. If ψ_0 is a valid ciphertext (i.e. produced by Encap) we require *correctness.*

Definition 3 (Correctness of decapsulation). *Fix any values of pk and sk, as output by KeyGen, and let $(\psi_0, K) = \mathsf{Encap}(\mathsf{pk})$ and $K' = \mathsf{Decap}(\mathsf{sk}, \psi_0)$. Then:*

$$\mathsf{Pr}[K \neq K'] = negl(\theta). \tag{2}$$

For our scheme, a relaxation of the above requirement will suffice, called *approximate correctness.* This notion was introduced by Katz and Vaikuntanathan in [19], and asks that the output of Decap is "close" (in Hamming sense) to the actual encapsulated key.

Definition 4 (t-Approximate Correctness). *Fix any values of pk and sk, as output by KeyGen, and let $(\psi_0, K) = \mathsf{Encap}(\mathsf{pk})$ and $K' = \mathsf{Decap}(\mathsf{sk}, \psi_0)$. Then:*

$$\mathsf{Pr}[\mathsf{d}(K, K') > t] = negl(\theta). \tag{3}$$

For invalid ciphertexts (i.e. produced by Encap^*), instead, we want Decap to return strings that are almost uniformly distributed. Following [2], we present three distinct notions in this regard.

Definition 5 (Universality). *Let SK and PK be random variables representing, respectively, sk and pk. We say that an HPS is (η, ν)-*universal *if $\tilde{H}_\infty(SK|PK) \geq \eta$ and, for any fixed values pk and $\mathsf{sk} \neq \mathsf{sk'}$, it holds:*

$$\mathsf{Pr}[\mathsf{Decap}(\mathsf{sk}, \psi_0) = \mathsf{Decap}(\mathsf{sk'}, \psi_0)] \leq \nu \tag{4}$$

where $\psi_0 = \mathsf{Encap}^*(\mathsf{pk})$.

Definition 6 (Smoothness). *For any fixed pk and $\mathsf{sk} \neq \mathsf{sk'}$, let $\psi_0 = \mathsf{Encap}^*(\mathsf{pk})$, $K = \mathsf{Decap}(\mathsf{sk}, \psi_0)$. Then we say that an HPS is* smooth *if:*

$$\Delta((\psi_0, K), (\psi_0, K')) = negl(\theta) \tag{5}$$

where $\psi_0 = \mathsf{Encap}^*(\mathsf{pk})$, $K = \mathsf{Decap}(\mathsf{sk}, \psi_0)$ and K' is chosen uniformly at random.

Definition 7 (Leakage Smoothness). *We say that an HPS is* λ-leakage smooth *if, for any (possibly randomized) function f with output size bounded by λ, it holds:*

$$\Delta((\psi_0, f(\mathsf{sk}), K), (\psi_0, f(\mathsf{sk}), K')) = negl(\theta) \tag{6}$$

for ψ_0, K and K' sampled as in Definition 6.

Finally, an HPS requires an indistinguishability property for ciphertexts, that is, a random valid ciphertext should be computationally indistinguishable from an invalid one. The definition is the following.

Definition 8 (Ciphertext Indistinguishability). *We define the following attack game between a challenger and an adversary $\mathcal{A}$:*

1. *Fix system parameters.*
2. *The adversary $\mathcal{A}$ makes a sequence of queries to the challenger, getting back public key/private key pairs $(\mathsf{pk}, \mathsf{sk})$.*
3. *The challenger fixes a target public key pk^*, then chooses a random bit b. If $b = 0$, it computes $(\psi_0, K) = \mathsf{Encap}(\mathsf{pk}^*)$, otherwise computes $\psi_0 = \mathsf{Encap}^*(\mathsf{pk}^*)$. It then gives ψ_0 to $\mathcal{A}$.*
4. *$\mathcal{A}$ keeps performing queries as above. No restrictions apply, hence $\mathcal{A}$ can even get sk^*.*
5. *Finally, $\mathcal{A}$ outputs $b^* \in \{0, 1\}$.*

The adversary succeeds if $b^* = b$. More precisely, we define the advantage *of $\mathcal{A}$ against HPS as*

$$\mathsf{Adv}_{\mathsf{HPS}}(\mathcal{A}, \theta) = \left| \mathsf{Pr}[b^* = b] - \frac{1}{2} \right|. \tag{7}$$

We say that an HPS satisfies the ciphertext indistinguishability *property if the advantage $\mathsf{Adv}_{\mathsf{HPS}}$ of any polynomial-time adversary $\mathcal{A}$ in the above adaptive attack model is negligible.*

Remark 1. In the original definition for an identity-based protocol, the adversary would perform queries adaptively submitting a certain identity and getting back the corresponding private key. Adapting this to the public-key setting just means that the adversary is allowed to see public key/private key pairs, including the "target" one. In both cases, this is a very strong requirement, meaning that ciphertexts have to be computationally indistinguishable even if the whole private key is revealed.

4 Leakage-resilient public-key encryption

We define here the notion of security for public-key encryption schemes under key-leakage attacks. An adversary in this setting is allowed to (adaptively) query a *leakage oracle*, submitting any function f and receiving $f(\mathsf{sk})$, with the only restriction that the total length of the output is not greater than a certain threshold λ. As pointed out by Akavia et al. [1], this is equivalent to querying the leakage oracle on a single function f whose total output doesn't exceed λ bits. The definition is given below.

Definition 9. *An adversary $\mathcal{A}$ for key-leakage security is a polynomial-time algorithm that plays the following attack game:*

1. *Query a key generation oracle to obtain a public key pk.*
2. *Submit a query f to the leakage oracle. The oracle will reply with $f(\mathsf{sk})$ provided that the output is less or equal to λ bits.*
3. *Choose $\phi_0, \phi_1 \in \mathsf{P}$ and submit them to an encryption oracle. The oracle will choose a random $b \in \{0, 1\}$ and reply with the "challenge" ciphertext $\psi^* = \mathsf{Enc}(\mathsf{pk}, \phi_b)$.*
4. *Output $b^* \in \{0, 1\}$.*

We define the advantage *of $\mathcal{A}$ against PKE as*

$$\mathsf{Adv}(\mathcal{A}, \theta) = \left| \mathsf{Pr}[b^* = b] - \frac{1}{2} \right|. \tag{8}$$

We say that a PKE scheme is semantically secure against λ-key-leakage attacks *if the advantage of any adversary $\mathcal{A}$ in the above attack model is negligible.*

As usual, the above notion can be extended to the chosen-ciphertext attack model, allowing for decryption queries before (CCA1) or after (CCA2) the generation of the challenge ciphertext. In this case we speak about resistance to, respectively, *a priori* and *a posteriori* chosen-ciphertext key-leakage attacks.

5 The Construction

Table 2: Code-based HPS.

Setup	Public parameters are a matrix $A \xleftarrow{\$} \mathbb{F}_2^{k \times n}$ and integers k, n, ℓ with $k < n$, $\ell > k$. Let δ be the minimum distance of the code having A as generator matrix and set $\rho = \delta/n$ and $\tau = \gamma\rho$ for a certain $\gamma > 0$. The set of encapsulated keys is defined as $\mathsf{K} = \mathbb{F}_2^\ell$.
KeyGen	selects matrices $M \xleftarrow{\$} \mathbb{F}_2^{\ell \times k}$ and $E \leftarrow \chi_\rho^{\ell \times n}$ and outputs the private key $\mathsf{sk} = M$ and the public key $\mathsf{pk} = MA + E$.
Encap	chooses $\boldsymbol{s} \leftarrow \chi_\tau^n$ and returns the ciphertext/key pair (ψ_0, K) where $\psi_0 = A\boldsymbol{s}$ and $K = \mathsf{pk} \cdot \boldsymbol{s}$.
Encap^*	chooses $\boldsymbol{r} \xleftarrow{\$} \mathbb{F}_2^k$ and returns the invalid ciphertext $\psi_0 = \boldsymbol{r}$.
Decap	takes as input the private key sk and a ciphertext ψ_0 and obtains K' as $\mathsf{sk} \cdot \psi_0$.

The choice of parameters here is important to guarantee that the construction satisfies the requirements presented in the previous section. In particular, we will need the rate $R = k/n$ to be high enough for ρ to be less than $1/\sqrt{n}$. We now analyze the three properties one at a time:

t-Approximate Correctness As in Definition 4, let $(\psi_0, K) = \mathsf{Encap}(\mathsf{pk})$ and $K' = \mathsf{Decap}(\mathsf{sk}, \psi_0)$; then K and K' differ by a factor of $E\boldsymbol{s}$, and $\mathsf{d}(K, K') = \mathsf{wt}(E\boldsymbol{s})$, hence we just need to bound the weight of this product. Remember that E and $\boldsymbol{s}$ are distributed, respectively, according to $\chi_\rho^{\ell \times n}$ and χ_τ^n, where $\rho = O(n^{-1/2-\varepsilon})$ and $\tau = \gamma\rho$ for $\gamma > 0$. We now use a result from Döttling, Müller-Quade and Nascimento [8]. A matrix $X \in \mathbb{F}_2^{\ell \times n}$ is said to be (β, ϵ)-*good* if for a fixed constant β and $\epsilon = \epsilon(n)$ it holds that for all $\boldsymbol{s} \in \mathbb{F}_2^n$ if $\mathsf{wt}(\boldsymbol{s}) \leq \epsilon n$ then $\mathsf{wt}(X\boldsymbol{s}) \leq \beta\ell$. It is proved in [8] that when $\rho = O(n^{-1/2-\varepsilon})$ and n is sufficiently large, for any fixed $\beta, \gamma > 0$ a matrix sampled from $\chi_\rho^{\ell \times n}$ is $(\beta, \gamma\rho)$-good with overwhelming probability. Thus in our case we will have $\mathsf{wt}(E\boldsymbol{s}) \leq t$ for $t = \beta\ell$, and this concludes the proof.

Universality A defines a random linear code $\mathcal{C}$, hence its minimum distance δ is on the GV bound with high probability. Consider the expected number of codewords in $\mathcal{B}(\boldsymbol{x}, r)$ for $\boldsymbol{x}$ chosen uniformly at random, $\mu_{\mathcal{C}}(r) = \mathbb{E}_{\boldsymbol{x} \in \mathbb{F}_q^n}[|\mathcal{C} \cap \mathcal{B}(\boldsymbol{x}, r)|]$. Following Dumer, Micciancio and Sudan [9], we know that $\mu_{\mathcal{C}}(r) = 2^{k-n} \cdot |\mathcal{B}(\mathbf{0}, r)|$ (we are interested in the case $q = 2$), and that for $r = \delta$ this number is equal to a certain constant $\mu > 1$. Since each row of M is chosen independently, it holds $H_\infty(\mathsf{sk}) \geq \mu^\ell$. This completes the first part. For the second part, recall that $\mathsf{Decap}(\mathsf{sk}, \psi_0) = M\psi_0$. Consider two private keys $M \neq M'$: then $\mathsf{Decap}(\mathsf{sk}, \psi_0) = \mathsf{Decap}(\mathsf{sk}', \psi_0) \iff M\psi_0 = M'\psi_0 \iff (M - M')\psi_0 = \mathbf{0}$. Now, $\ell > k$ and both M and M' are generated uniformly at random, so the matrix $N = (M - M')$ is of full rank k with high probability [5], say p. It follows that $(M - M')\psi_0 = \mathbf{0} \iff \psi_0 = \mathbf{0}$. We conclude that the code-based HPS is (η, ν)-universal, for $\eta = \mu^\ell$ and $\nu = 1 - p$.

Ciphertext Indistinguishability. We know that $\rho = O(n^{-1/2-\varepsilon})$, thus choosing $\boldsymbol{s}$ according to χ_ρ produces a vector with weight below the GV bound. As proved by Fischer and Stern in [11], the vector $\psi_0 = A\boldsymbol{s}$ is pseudorandom. Ciphertext indistinguishability follows directly since clearly the private key M doesn't carry information about the ciphertext.

Remark 2. We remark that the t-approximate correctness of the scheme relies heavily on a careful choice of the values ρ and τ, which are selected such that $\boldsymbol{s}$ and the rows of E have very low weight. It is easy to see that, if this condition is not respected, the weight of the corresponding product $E\boldsymbol{s}$ grows very quickly. One could in fact imagine an attack scenario aimed at obtaining an alternative decapsulation key K'', in which the attacker produces a vector $\boldsymbol{s}' \neq \boldsymbol{s}$ such that $A\boldsymbol{s}' = A\boldsymbol{s}$, and subsequently uses $\boldsymbol{s}'$ to get K'' as $\mathsf{pk} \cdot \boldsymbol{s}'$. Because of the hardness of SDP, though, such an attacker would only be able to recover a vector $\boldsymbol{s}'$ having high weight; for the above argument, the difference factor $E(\boldsymbol{s} + \boldsymbol{s}')$ would also have high weight, hence this attack would not work in practice.

6 The Scheme

In this section, we show how to use the HPS that we presented above to achieve leakage-resilient public-key encryption. In addition to the "standard" protocol presented in [2], we have to include an error-correcting code to deal with the error coming from the approximate correctness. High error-correction capacity can be achieved, for example, by using list-decodable codes; Guruswami and Rudra in [15] show how it is possible to obtain codes that are list-decodable up to a radius $1 - R - \varepsilon$, for any $\varepsilon > 0$.

Table 3: Leakage-Resilient Public-Key Encryption Scheme.

Setup	Set public parameters as in Table 2, and let m be the length of the plaintexts. Fix an integer t' and set $t'' = t + t'$, then select an $[\ell, m]$ linear code $\mathcal{C}$ which is decodable up to the radius t''.
KeyGen	Run $\mathsf{KeyGen}^{\mathsf{HPS}}$ and return the private key $\mathsf{sk} = M$ and the public key $\mathsf{pk} = MA + E$.
Enc	On input a plaintext $\phi \in \{0,1\}^m$, run $\mathsf{Encap}(\mathsf{pk})$ to obtain the pair (ψ_0, K), sample a random vector $\boldsymbol{z} \overset{\$}{\leftarrow} \mathbb{F}_2^\ell$ having $\mathsf{wt}(\boldsymbol{z}) \leq t'$, then set $\psi_1 = K + \boldsymbol{z} + \mathsf{Encode}_{\mathcal{C}}(\phi)$. Finally, output the final ciphertext $\psi = (\psi_0, \psi_1)$.
Dec	On input a private key $\mathsf{sk} = M$ and a ciphertext $\psi = (\psi_0, \psi_1)$, calculate K' as $\mathsf{Decap}(\mathsf{sk}, \psi_0)$ and return the plaintext $\phi = \mathsf{Decode}_{\mathcal{C}}(K' + \psi_1)$.

As we mentioned above, the correctness of the scheme depends on the t-approximate correctness of the HPS, and the use of error-correcting codes. In fact $K' + \psi_1 = \mathsf{Encode}_{\mathcal{C}}(\phi) + E\boldsymbol{s} + \boldsymbol{z}$ and we expect $E\boldsymbol{s}$ to have weight less or equal to t and consequently $\mathsf{wt}(E\boldsymbol{s}+\boldsymbol{z}) \leq t+t' = t''$. Hence by applying the decoding algorithm is possible to recover the plaintext ϕ.

We now proceed to prove the security of the scheme. We start with a result from Alwen et al. [2, Theorem 3.1].

Theorem 1. *Let $\mathcal{H}$ be an (η, ν)-universal HPS with key space $\mathsf{K} = \{0,1\}^{\ell}$. Then $\mathcal{H}$ is also λ-leakage smooth as long as $\lambda \leq \eta - \ell - \omega(\log \theta)$ and $\nu \leq 2^{-\ell}(1 + negl(\theta))$.*

It is easy to see that the code-based HPS that we described in the previous section satisfies the conditions of Theorem 1. In addition, we will need the following computational assumption.

Assumption 1 *Let $E, \boldsymbol{s}$ and $\boldsymbol{z}$ be distributed as in Table 3. Then given E and $\boldsymbol{y} = E\boldsymbol{s} + \boldsymbol{z}$ it is hard to recover $\boldsymbol{s}$.*

One could think to use a generic decoding algorithm (e.g. Information Set Decoding [22]) or a dedicated decoding algorithm (e.g. bit flipping as for LDPC codes [12]); all these approaches, however, require at some point to check the syndrome equations. Because of the presence of $\boldsymbol{z}$, these equations can't be trusted. The attacker would then need to guess the positions of $\boldsymbol{z}$, either beforehand, or during the execution of the algorithm. In both cases, this implies a huge computational effort: there are in fact $N = \sum_{i=0}^{t'} \binom{n}{i}$ possibilities for $\boldsymbol{z}$. Therefore, it is plausible to assume that there is no efficient way to recover $\boldsymbol{s}$.

We are now ready to prove the following theorem.

Theorem 2. *Given that Assumption 1 holds, the scheme in Table 3 is semantically secure against λ-key-leakage attacks.*

Proof. For our security analysis we use a sequence of games. This is inspired by the proof of [2, Theorem 4.1], although a few *ad hoc* modifications are needed, due to the particular nature of our scheme.

Game 0: This is the semantic security game with leakage λ as presented in Definition 9.

Game 1: This game proceeds exactly as Game 0, except that we modify the encryption oracle as follows. We calculate $(\psi_0, K) = \mathsf{Encap}(\mathsf{pk})$ as usual, but instead of using K for encrypting the message, we use $K' = \mathsf{Decap}(\mathsf{sk}, \psi_0)$. Because of the approximate correctness, we have to artificially add some noise to preserve the structure of the ciphertext. We thus generate a random vector $\boldsymbol{y}$ of weight less or equal to t'', according to the same distribution of $E\boldsymbol{s}+\boldsymbol{z}$. The challenge ciphertext will then be (ψ_0^*, ψ_1^*), where $\psi_0^* = \psi_0$ and $\psi_1^* = K' + \boldsymbol{y} + \mathsf{Encode}_{\mathcal{C}}(\phi_b)$. Now, suppose the adversary is in possession of the private key. In order to distinguish between the two games, it could easily recover E from pk and $\boldsymbol{y}$ from ψ_1^* and try to solve for $\boldsymbol{s}$. However, because of Assumption 1, we claim that there is no efficient way to do this. Hence, the two games are computationally indistinguishable.

Game 2: This is the same as Game 1, but we modify again the encryption oracle, now replacing a valid ciphertext with an invalid one. More precisely, we calculate $\psi_0 = \mathsf{Encap}^*(\mathsf{pk})$ and $K' = \mathsf{Decap}(\mathsf{sk}, \psi_0)$, then return the challenge ciphertext (ψ_0^*, ψ_1^*) where $\psi_0^* = \psi_0$ and $\psi_1^* = K' + \boldsymbol{y} + \mathsf{Encode}_{\mathcal{C}}(\phi_b)$. By the ciphertext indistinguishability property of the scheme, the two games are computationally indistinguishable. Note that, by definition, this indistinguishability holds even if the whole private key is revealed, hence in particular it holds for any bounded leakage $f(\mathsf{sk})$.

Game 3: In Game 3 we proceed as in Game 2, but now we generate ψ_1^* as a uniformly random string. That is, we calculate $\psi_0^* = \mathsf{Encap}^*(\mathsf{pk})$ and $\psi_1^* \overset{\$}{\leftarrow} \mathbb{F}_2^m$, then return the challenge ciphertext (ψ_0^*, ψ_1^*). Game 2 and Game 3 are statistically indistinguishable because of the leakage smoothness property.

Finally, in Game 3 the advantage of any adversary $\mathcal{A}$ is equal to 0, since this is independent from the chosen bit b. This completes the proof.

□

7 Conclusions

We have shown how to construct an HPS based on coding theory assumptions. The public-key encryption scheme that is based on it is inspired by a framework from Alwen et al. [2]; however we do not need to use randomness extractors to achieve semantic security against key-leakage attacks. This is because of the universality of our HPS construction. We remark that our scheme is but a first step towards achieving efficient leakage-resilient code-based encryption schemes. We thus hope to stimulate discussion about some open questions that stem from this work. For example, it would be important to improve the scheme in order to resist chosen-ciphertext attacks, without having to use impractical variants such as the one based on the Naor-Yung "double encryption" paradigm presented in [21].

References

1. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.
2. J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs. Public-Key Encryption in the Bounded-Retrieval Model. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 113–134. Springer, 2010.
3. J. Alwen, Y. Dodis, and D. Wichs. Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In Halevi [17], pages 36–54.
4. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384 – 386, may 1978.
5. I. F. Blake and C. Studholme. Properties of random matrices and applications. *Unpublished report available at http://www. cs. toronto. edu/˜ cvs/coding*, 2006.
6. R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In L. R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.
7. G. Di Crescenzo, R. J. Lipton, and S. Walfish. Perfectly Secure Password Protocols in the Bounded Retrieval Model. In Halevi and Rabin [18], pages 225–244.
8. N. Döttling, J. Müller-Quade, and A. C. A. Nascimento. IND-CCA Secure Cryptography Based on a Variant of the LPN Problem. In X. Wang and K. Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 485–503. Springer, 2012.
9. I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003.
10. S. Dziembowski. Intrusion-Resilience Via the Bounded-Storage Model. In Halevi and Rabin [18], pages 207–224.
11. J.-B. Fischer and J. Stern. An Efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding. In U. M. Maurer, editor, *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 245–255. Springer, 1996.
12. R. Gallager. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1):21–28, 1962.
13. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In C. Dwork, editor, *STOC*, pages 197–206. ACM, 2008.
14. S. Goldwasser and S. Micali. Probabilistic Encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
15. V. Guruswami and A. Rudra. Explicit capacity-achieving list-decodable codes. In J. M. Kleinberg, editor, *STOC*, pages 1–10. ACM, 2006.

16. J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest We Remember: Cold Boot Attacks on Encryption Keys. In P. C. van Oorschot, editor, *USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.

17. S. Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.

18. S. Halevi and T. Rabin, editors. *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*. Springer, 2006.

19. J. Katz and V. Vaikuntanathan. Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 636–652. Springer, 2009.

20. E. Kiltz, K. Pietrzak, M. Stam, and M. Yung. A New Randomness Extraction Paradigm for Hybrid Encryption. In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 590–609. Springer, 2009.

21. M. Naor and G. Segev. Public-Key Cryptosystems Resilient to Key Leakage. In Halevi [17], pages 18–35.

22. C. Peters. Information-Set Decoding for Linear Codes over F_q. In N. Sendrier, editor, *PQCrypto*, volume 6061 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2010.

23. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *STOC*, pages 84–93. ACM, 2005.