IND-CCA secure Cryptography based on a variant of the LPN Problem

Nico Döttling[1], Jörn Müller-Quade[1], and Anderson Nascimento[2]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
{doettling,mueller-quade}@kit.edu
[2] University of Brasilia, Brasilia, Brazil
andclay@ene.unb.br

Abstract. In 2003 Michael Alekhnovich (FOCS 2003) introduced a novel variant of the learning parity with noise problem and showed that it implies IND-CPA secure public-key cryptography. In this paper we introduce the first public-key encryption-scheme based on this assumption which is IND-CCA secure in the standard model. Our main technical tool to achieve this is a novel all-but-one simulation technique based on the correlated products approach of Rosen and Segev (TCC 2009). Our IND-CCA1 secure scheme is asymptotically optimal with respect to ciphertext-expansion. To achieve IND-CCA2 security we use a technique of Dolev, Dwork and Naor (STOC 1991) based on one-time-signatures. For practical purposes, the efficiency of the IND-CCA2 scheme can be substantially improved by the use of additional assumptions to allow for more efficient signature schemes. Our results make Alekhnovich's variant of the learning parity with noise problem a promising candidate to achieve post quantum cryptography.

Keywords: IND-CCA2 Security, Learning Parity with Noise, All-But-One Decryption

1 Introduction

This paper presents the first IND-CCA2 secure cryptosystem based on a computational assumption first introduced by Michael Alekhnovich in the year 2003 [Ale03]. This assumption essentially states that for a given random linear code C with a constant rate, a random code word with an inverse square root fraction of noise is indistinguishable from a random string. Alekhnovich [Ale03] was able to construct a semantically secure cryptosystem which was based solely on this assumption. It can be seen as an special case of the decisional learning parity with noise (LPN) problem. The decisional LPN problem (henceforth LPN problem), asks to distinguish noisy binary linear equations $Ax + e$ from uniformly random. The problem is parametrized by the number of samples provided (i.e the number of rows of A) and the amount of noise (i.e. the distribution of e). While most

cryptographic constructions based on LPN (e.g. [HB01,JW05,KSS10]) use the standard parameter-choice of a polynomial number of samples and a constant fraction of noise, Alekhnovich's LPN problem uses a *linear* number of samples and an inverse square root fraction of noise. These two parameter-choices are apparently incomparable. On one side, providing a larger amount of samples makes the problem apparently easier. On the other side, a larger amount of noise seems to make the problem harder. Nevertheless, Alekhnovich's parameter choice seems to yield the stronger assumption, as constructing a public key cryptosystem from LPN with a constant fraction of noise remains an important open problem.

LPN assumptions are of a more combinatorial nature and seem incomparable to the algebraic assumptions needed for the McEliece cryptosystem. For the security of the McEliece cryptosystem one has to additionally assume that scrambled Goppa-codes are computationally indistinguishable from random linear codes [McE78,BS08,NIKM08,DMQN09]. Moreover, though there is a syntactic similarity to the learning with errors (LWE) problem, LPN and LWE also seem rather incomparable. LWE asks to distinguish a polynomial number noisy linear equations over $\mathbb{Z}_q$ (for a polynomial sized q), where the error-distribution is euclidean, from uniformly random. IND-CCA2 encryption schemes based on LWE [PW08,Pei09,MP12] use properties that are very specific to LWE (e.g. short dual-lattice bases) and not available in the binary domain. It has been open for nine years if an IND-CCA2 secure scheme could be built from Alekhnovich's LPN problem. In this paper we present such a IND-CCA2 secure scheme which is based on the all-but-one approach [DDN00,PW08,RS09]. The new construction is asymptotically optimal for IND-CCA1 security. It has only a constant factor ciphertext-expansion and the ciphertexts are of size $O(k^{2/(1-2\epsilon)})$, where k is the security parameter and ϵ a small constant. To achieve IND-CCA2 security we use a generic transformation based on one-time-signatures [DDN00]. A more efficient construction is possible using additional assumptions yielding more efficient signature schemes. The trapdoor of our scheme is substantially different from Alekhnovich's original construction, but bears some similarities with the above-mentioned lattice-based constructions. It allows witness recovery and decryption with incomplete keys, which is necessary for applying the all-but-one approach. Different from [PW08,RS09,Pei09,DMQN09] we do not achieve the all-but-one property by repeatedly encrypting the same ciphertext or a correlated product. We employ a bitwise decryption and use error correction to cope with incomplete decryptions. The novel all-but-one simulation technique employed in this construction allows for a significant improvement in efficiency compared with previous constructions. While this new technique might be of interest in lattice-based cryptography, we see no obvious way to make use of our technique in McEliece-based constructions. Crucial to our technique is the ability to recover individual bits of a plaintext from the ciphertext using a partial secret key. This, however, seems out of reach for constructions based on the McEliece assumption.

Related Work Ciphertext indistinguishability under chosen ciphertext attacks (IND-CCA2) security [RS91] is one of the strongest known notions of security for public key encryption schemes (PKE). Many computational assumptions have been used in the literature for obtaining cryptosystems meeting this security notion. Given one-way trapdoor permutations, CCA2 security can be obtained from any semantically secure public key cryptosystem [NY90,Sah99,Lin03]. Efficient constructions are also known based on number-theoretic assumptions [CS98,CS03,HK09], lattice-based assumptions [PW08,Pei09,MP12], the McEliece assumption [DMQN09] or identity based encryption schemes [CHK04].

2 Preliminaries

2.1 Coding-Theory

We need a few coding-theoretic facts and constructions for our schemes and proofs. We denote the finite field with q elements by $\mathbb{F}_q$. The hamming-weight $|x|$ of a vector $x \in \mathbb{F}_q^n$ is the number of its non-zero locations. The q-ary entropy function is defined as $H_q(\alpha) = \alpha \log_q(q-1) - \alpha \log_q \alpha - (1-\alpha)\log_q(1-\alpha)$. It assumes its maximum at $\alpha = 1 - 1/q$ with $H_q(1-1/q) = 1$. The volume $\mathsf{Vol}_q(\alpha n, n)$ of the hamming-ball of radius αn in $\mathbb{F}_q^n$ can be bounded by $q^{H_q(\alpha)\cdot n - o(n)} \leq \mathsf{Vol}_q(\alpha n, n) \leq q^{H_q(\alpha)\cdot n}$.

Random Codes and the Gilbert-Varshamov bound The Gilbert-Varshamov bound guarantees the existence of q-ary codes with almost maximal relative minimum-distance $1 - 1/q$. Moreover, with high probability, randomly chosen codes enjoy this property. Let $n, d, k \in \mathbb{N}$ and $\lambda > 0$. If it holds that $k \leq n - \log_q \mathsf{Vol}_q(d, n) - \lambda n$, then the code $\mathcal{C}(G)$ generated by a uniformly chosen matrix $G \in \mathbb{F}_q^{n\times k}$ has minimum-distance at least d, except with probability $q^{-\lambda n}$. Therefore, if $\delta < 1 - 1/q$ it holds that $n - \log_q \mathsf{Vol}_q(\delta n, n) \geq (1 - H_q(\delta))n =: \zeta n$. Thus, if $k \leq \zeta n/2$, a uniformly random chosen matrix $G \in \mathbb{F}_q^{n\times k}$ generates a code $\mathcal{C}(G)$ with minimum-distance at least δn, except with probability $q^{-\zeta n/2}$.

Asymptotically good codes with efficient error-correction The decryption algorithm of our scheme will introduce errors in the plaintext when decrypting. We will therefore use asymptotically good error-correcting codes $\mathcal{C}$ with efficient error-correction algorithm $\mathsf{Decode}_{\mathcal{C}}$ to encode plaintexts. Prominent examples of such codes are binary expander-codes [SS96,Zém01]: There exists an explicit family of binary linear codes $\{\mathcal{C}_n\}$ of constant rate R arbitrarily close to 1 that can efficiently correct an α-fraction of errors, for a constant $\alpha > 0$.

2.2 Bernoulli distributions and bounds

In this section we will briefly gather some facts about low-noise Bernoulli distributions. While Alekhnovich's [Ale03] original proposal used a noise distribution

that samples vectors of low-weight t uniformly at random, we will use Bernoulli-distributions where each bit of a vector is 1 with probability t/n and otherwise 0. The advantage of Bernoulli-distributions over the former distribution is that all components are independent of one another. We will take advantage of this fact when bounding the hamming-weight of matrix-vector products when the matrix is chosen from a Bernoulli distribution. The decryption-algorithm of Alekhnovich's and our encryption-scheme computes inner-products of Bernoulli-distributed vectors. To ensure that the inner-product of two Bernoulli-distributed vectors is 0 with high probability, we need to choose the bit-flip probability ρ below a $1/\sqrt{n}$ amount. If ρ is too big (e.g. constant), then the distribution of the inner-product would be statistically close to uniform and our decryption-approach would fail. Finally, we show that matrices X chosen from a component-wise low-noise Bernoulli distribution enjoy (with high probability) the property, that a product Xs has low-hamming-weight, for any vector s with sufficiently small hamming-weight. We will call such matrices *good*, and we will use this property for proving correctness of our schemes and in the proof of IND-CCA1 security.

Bernoulli distributions For a noise-parameter ρ, we write χ_ρ for the Bernoulli-distribution that outputs 1 with probability ρ and 0 with probability $1-\rho$. The distribution of the hamming-weight of a vector of n iid distributed Bernoulli-distributed random variables is the binomial distribution $B_{\rho,n}$. Throughout the paper, we frequently need to bound Binomial distributions. For this we require two different Chernoff bounds. Let x be distributed by χ_ρ^n.

1. It holds for any $R \geq 6\rho n$ that $\Pr[|x| > R] < 2^{-R}$.
2. It holds for any $0 < \delta < 1$ that $\Pr[||x| - \rho n| \geq \delta\rho n] < 2e^{-\delta^2\rho n/3}$.

Distributions of inner products For the decryption-algorithms of our schemes we require that the inner-product of a Bernoulli-distributed vector x and a vector s of small hamming weight is 0 with probability bounded away from $1/2$. We will thus show that the probability of the inner-product being 1 is sub-constant for a proper choice of ρ. Let $s \in \mathbb{F}_2^n$ be a fixed vector and x be distributed by χ_ρ^n. By a simple XOR-Lemma, it holds that

$$\Pr[x^T s = 1] = \frac{1}{2} \cdot (1 - (1-2\rho)^{|s|}),$$

i.e. the random variable $x^T s$ is distributed according to $\chi_{\rho'}$ with $\rho' = \frac{1}{2} \cdot (1 - (1-2\rho)^{|s|})$. If it holds that $\rho = \rho(n) = O(n^{-1/2-\epsilon})$ for some constant $\epsilon > 0$ and $|s| < \gamma\rho n$ for some constant $\gamma > 0$, we get the following estimate for ρ'. By the mean-value-theorem it holds for any p in the interval $(0, e^{-1})$ that $e^{-ep} \leq 1-p$, therefore we get

$$\rho' = \frac{1}{2}\cdot(1-(1-2\rho)^{|s|}) \leq \frac{1}{2}\cdot(1-e^{-2e\rho|s|}) \leq \frac{1}{2}\cdot(1-e^{-2e\gamma\rho^2 n}) = \frac{1}{2}\cdot(1-e^{-O(n^{-2\epsilon})}).$$

The last term is sub-constant in n, i.e. $\rho'(n) = o(1)$. This means that for sufficiently large n ρ' is arbitrarily small.

Multiplication with random matrices We will now give bounds for how much the hamming weight of a vector s increases when multiplied with a matrix $X \in \mathbb{F}_2^{l \times n}$ chosen from $\chi_\rho^{l \times n}$. Let x be distributed by χ_ρ^n and the hamming-weight of s be bounded by $|s| < \gamma\rho n$. Then by the above $\rho' = \Pr[x^T s = 1]$ can be made an arbitrarily small constant if $\rho = O(n^{-1/2-\epsilon})$. If $X \in \mathbb{F}_2^{l \times n}$ is distributed by $\chi_\rho^{l \times n}$, then $|Xs|$ is distributed by the Binomial-distribution $B_{\rho',l}$. The Chernoff-bound thus yields that for any $R \geq 6\rho' l$ it holds that $\Pr[|Xs| > R] < 2^{-R}$. The volume $\mathsf{Vol}_2(\gamma\rho n, n)$ of the hamming-ball of radius $\gamma\rho n$ in $\mathbb{F}_2^n$ is bounded by $2^{H_2(\gamma\rho)n}$. Thus, there are at most $2^{H_2(\gamma\rho)n}$ vectors s satisfying $|s| < \gamma\rho n$. A union-bound yields for any $R \geq 6\rho' l$

$$\Pr[\exists s \in \mathbb{F}_2^n : |s| < \gamma\rho n \text{ and } |Xs| > R] < 2^{H_2(\gamma\rho)n} \cdot 2^{-R}.$$

If $l = \Omega(n)$ and $\beta > 0$ it holds that

$$\Pr[\exists s \in \mathbb{F}_2^n : |s| < \gamma\rho n \text{ and } |Xs| > \beta l] < 2^{-\Omega(n)},$$

as $H_2(\gamma\rho)n$ is sub-linear in n (i.e. $o(n)$) since $\rho = O(n^{-1/2-\epsilon})$.

Definition 1. *Fix a constant β and $\epsilon = \epsilon(n)$. We shall call a matrix $X \in \mathbb{F}_2^{l \times n}$ (β, ϵ)-good, if for all $s \in \mathbb{F}_2^n$ with $|s| < \epsilon n$ it holds that $|Xs| \leq \beta l$.*

The above now implies that for $\rho = O(n^{-1/2-\epsilon})$, any fixed $\beta, \gamma > 0$ and sufficiently large n, a matrix X sampled from $\chi_\rho^{l \times n}$ is $(\beta, \gamma\rho)$-good with overwhelming probability in n.

2.3 Public Key Encryption

This Section is only meant to provide reference for the standard notions of security for encryption schemes and can be safely skipped. Let k be a security parameter.

Definition 2. *A public key encryption scheme* PKE *is a tuple* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, such that*

- $\mathsf{KeyGen}(1^k)$ *is a PPT-algorithm that takes a security-parameter k and outputs a pair of public and private keys (pk, sk).*
- $\mathsf{Enc}_{pk}(m)$ *is a PPT-algorithm that takes a public key pk, a message m and outputs a ciphertext c.*
- $\mathsf{Dec}_{sk}(c)$ *is an efficient deterministic algorithm taking as input a secret key sk and a ciphertext c and outputs a plaintext m.*

A standard-requirement for public key encryption is correctness.

Definition 3. *We say that* $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is correct, if it holds for all plaintexts m that*

$$\Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) \neq m : (pk, sk) = \mathsf{KeyGen}(1^k)] < \mathsf{negl}(k).$$

The three security notions for public key encryption we are concerned with in this paper are IND-CPA, IND-CCA1 and IND-CCA2 security. Let $\mathcal{A}$ be an adversary.

Experiment: IND-CPA

- Generate a pair of keys $(pk, sk) = \mathsf{KeyGen}(1^k)$. Run $\mathcal{A}$ on input pk.
- Once $\mathcal{A}$ outputs a pair (m_0, m_1), flip a coin b and compute $c^* = \mathsf{Enc}_{pk}(m_b)$. Give input c^* to $\mathcal{A}$ and continue its computation.
- Let b' be $\mathcal{A}$'s output. Output 1 if $b' = b$ an 0 otherwise.

Experiment: IND-CCA1

- Generate a pair of keys $(pk, sk) = \mathsf{KeyGen}(1^k)$. Give $\mathcal{A}$ access to a decryption-oracle $\mathsf{Dec}_{sk}(\cdot)$ and run $\mathcal{A}$ on input pk.
- Once $\mathcal{A}$ outputs a pair (m_0, m_1), flip a coin b and compute $c^* = \mathsf{Enc}_{pk}(m_b)$. Give input c^* to $\mathcal{A}$ and continue its computation *without* access to the decryption-oracle.
- Let b' be $\mathcal{A}$'s output. Output 1 if $b' = b$ an 0 otherwise.

Experiment: IND-CCA2

- Generate a pair of keys $(pk, sk) = \mathsf{KeyGen}(1^k)$. Give $\mathcal{A}$ access to a decryption-oracle $\mathsf{Dec}_{sk}(\cdot)$ and run $\mathcal{A}$ on input pk.
- Once $\mathcal{A}$ outputs a pair (m_0, m_1), flip a coin b and compute $c^* = \mathsf{Enc}_{pk}(m_b)$. Give input c^* to $\mathcal{A}$ and continue its computation *with* access to the decryption-oracle.
- Let b' be $\mathcal{A}$'s output. Output 1 if $b' = b$ an 0 otherwise.

Definition 4. *For $X \in \{CPA, CCA1, CCA2\}$, we say that the scheme $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-X secure, if it holds for every PPT-adversary $\mathcal{A}$ that $\mathsf{Adv}_{\mathit{IND\text{-}X}}(\mathcal{A}) = |\Pr[\mathit{IND\text{-}X}(\mathcal{A}) = 1] - 1/2| \leq \mathsf{negl}(k)$.*

2.4 One-Time Signatures

We also briefly recall the definition of one-time signatures [Lam79]. Let k be a security parameter.

Definition 5. *A one-time signature scheme SIG is a tuple $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$, such that*

- *$\mathsf{Gen}(1^k)$ is a PPT-algorithm that takes a security-parameter k and outputs a pair of verification and signature keys (vk, sgk).*
- *$\mathsf{Sign}_{sgk}(m)$ is a PPT-algorithm that takes a signature key sgk, a message m and outputs a signature σ.*
- *$\mathsf{Verify}_{vk}(m, \sigma)$ is a PPT-algorithm taking as input a verification key vk, a message m and a signature c and outputs a bit $b \in \{0, 1\}$.*

We require one-time signature schemes to be correct.

Definition 6. *We say that $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ is correct, if it holds for all messages m that*

$$\Pr[\mathsf{Verify}_{vk}(m, \mathsf{Sign}_{sgk}(m)) = 1 : (vk, sgk) = \mathsf{Gen}(1^k)] > 1 - \mathsf{negl}(k).$$

Moreover, we require existential unforgeability under one-time chosen message attacks (EUF-CMA security), specified by the following experiment. Let $\mathcal{A}$ be an adversary.

Experiment: EUF-CMA

- Generate a pair of keys $(vk, sgk) = \mathsf{Gen}(1^k)$. Give $\mathcal{A}$ a access to a signing-oracle $\mathsf{Sign}_{sgk}(\cdot)$ that signs one message m^* of $\mathcal{A}$'s choice and then outputs $\perp$ for any further signing-queries. Run $\mathcal{A}$ on input vk
- Once $\mathcal{A}$ outputs a pair (m, σ) with $m \neq m^*$, compute $b = \mathsf{Verify}_{vk}(m, \sigma)$ and output b. Otherwise output 0.

Definition 7. *We say that* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is EUF-CMA secure, if it holds for every PPT-adversary* $\mathcal{A}$ *that* $\Pr[\textit{EUF-CMA}(\mathcal{A}) = 1] \leq \mathsf{negl}(k)$.

EUF-CMA secure one-time signature schemes can be constructed from any one-way function [Lam79].

3 The Hardness-Assumption

The basic problem we will base the security of our scheme upon is a variant of the decisional learning parity with noise (LPN) problem. Roughly speaking, the LPN problem asks to distinguish a number of noisy samples of a linear function (specified by a secret vector x) from uniform random. The variant considered here differs from the standard LPN problem in two aspects. First, the distinguisher is provided only linear number of samples, rather than an arbitrary polynomial number. Second, the noise-level in this variant is significantly lower than in the standard LPN problem. While the standard LPN problem comes with an error-distribution that flips each output-bit with a small, but constant probability, for this variant the probability is sub-constant. More precisely, we will work with a bit-flip probability of the order $O(n^{-1/2-\epsilon})$ for some small constant ϵ. Here, n is the size of the secret x in bits.

Problem 1. Let $n \in \mathbb{N}$ be a problem parameter, $m = O(n)$ and $\epsilon > 0$ and $\rho = \rho(n) = O(n^{-1/2-\epsilon})$. Let $A \in \mathbb{F}_2^{m \times n}$ be chosen uniformly at random, $x \in \mathbb{F}_2^n$ be chosen uniformly at random and e according to χ_ρ^m. The problem is, given A and y, to decide whether y is distributed according to $Ax+e$ or chosen uniformly at random.

Currently, the best classical algorithms to attack Problem 1 require time of the order $2^{\Omega(n^{1/2-\epsilon})}$ [Ste88,CC98,MMT11,BLP11,BJMM12]. Moreover, there are no quantum algorithms known performing significantly better than the best classical algorithms. In our constructions we will choose n by $n = O(k^{2/(1-2\epsilon)})$, where k is the security parameter. This normalizes the hardness of Problem 1 to $2^{\Theta(k)}$. Thus, we choose ρ by $\rho(k) = O(k^{-(1+2\epsilon)/(1-2\epsilon)})$. In the full version of this paper, we provide a reduction establishing the hardness of problem 1 based on the hardness-assumption used in [Ale03], which uses a different error-distribution. It will be necessary to use a *normal-form* (as in [ACPS09]) of Problem 1 in our cryptographic constructions, which is stated in Problem 2. In this normal-form, the secret x is drawn from the noise-distribution χ_ρ^n.

Problem 2. Let $n \in \mathbb{N}$ be a problem parameter, $m = O(n)$, $\epsilon > 0$ and $\rho = O(n^{-1/2-\epsilon})$. Let $A \in \mathbb{F}_2^{m \times n}$ be chosen uniformly at random, x be distributed according to χ_ρ^n and e be distributed according to χ_ρ^m. The problem is, given A and y, to decide whether y is distributed according to $Ax+e$ or chosen uniformly at random.

The hardness of Problem 2 can be established by a simple reduction from Problem 1, given in the full version of this paper. By a simple hybrid-argument, it follows that that a matrix-version of problem 2 is also hard.

Problem 3. Let $n \in \mathbb{N}$ be a problem parameter, $m, k = \Theta(n)$, $\epsilon > 0$ and $\rho = O(n^{-1/2-\epsilon})$. Let $A \in \mathbb{F}_2^{n \times k}$ be chosen uniformly at random, $T \in \mathbb{F}_2^{m \times n}$ be distributed according to $\chi_\rho^{m \times n}$ and X be distributed according to $\chi_\rho^{m \times k}$. The problem is, given A and B, to decide whether B is distributed according to $TA + X$ or chosen uniformly at random in $\mathbb{F}_2^{m \times k}$.

In the security-proof for our schemes, we will use Problem 3 to establish pseudorandomness of the public keys, while we use Problem 2 to establish pseudorandomness of the ciphertexts.

4 Outline of the Techniques

In this Section, we will outline the techniques used to construct an IND-CCA1 secure scheme based on the hardness of Problem 2 and Problem 3. We will provide the full presentation in the subsequent sections. Let henceforth $\rho = O(n^{-1/2-\epsilon})$ for a small constant $\epsilon > 0$.

We will start with a rough outline of a scheme that encrypts single bits and has a substantial decryption-error. On a technical level, this first building block resembles the schemes of Regev [Reg05] and the Dual-Regev Scheme of Gentry et al. [GPV08] (which both live in the LWE realm). Public keys for our scheme are pairs (A, b^T), where $A \in \mathbb{F}_2^{l_1 \times n}$ is chosen uniformly at random and $b^T = t^T A + x^T$ with $t \in \mathbb{F}_2^{l_1}$ is distributed by $\chi_\rho^{l_1}$ and $x \in \mathbb{F}_2^n$ by χ_ρ^n. The secret key is t^T. To encrypt a message $m \in \mathbb{F}_2$, sample s according to χ_ρ^n, e_1 according to χ_ρ^l and e_2 according to χ_ρ. Compute $c = (As + e_1, b^T s + e_2 + m)$ and output c. To decrypt a ciphertext $c = (c_1, c_2)$, compute $y = c_2 - t^T c_1$ and output y. The output y is a *noisy* version of the plaintext m, since it holds that $y = c_2 - t^T c_1 = b^T s + e_2 + m - t^T(As + e_1) = m + t^T As + x^T s + e_2 - t^T As - t^T e_1 = m + x^T s + e_2 - t^T e_1$. By the properties of the distribution χ_ρ, the error-term $v = x^T s + e_2 - t^T e_1$ is 0 with probability bounded away from 1/2, i.e. it holds $y = m$ with substantial probability.

This decryption-error can be dealt with by encoding m (which is now a bit-vector of length n) using an error-correcting code as follows. Let $G \in \mathbb{F}_2^{l_2 \times n}$ be the generator-matrix of a binary linear error-correcting code $\mathcal{C}$. The modified scheme works as follows. Public keys are of the form (A, B) with A as above and $B = TA + X$, where T is chosen from $\chi_\rho^{l_2 \times l_1}$ and X from $\chi_\rho^{l_2 \times n}$. The secret key is T. Messages $m \in \mathbb{F}_2^n$ are encrypted as $c = (As + e_1, Bs + e_2 +$

Gm), with s, e_1, e_2 sampled from the corresponding χ_ρ distributions. Decryption computes $y = c_2 - Tc_1 = Gm + Xs + e_2 - Te_1$. Since the matrices T and X were chosen from a χ_ρ distribution, they are good (as defined in Section 2.2) with overwhelming probability. Thus the error-term $v = Xs + e_2 - Te_1$ has a low hamming-weight and we can use the decoding-procedure of $\mathcal{C}$ to recover m. The IND-CPA security of this scheme follows easily by the hardness of Problem 2 and Problem 3. However, we will require a witness-recovering IND-CPA scheme for the construction of our IND-CCA scheme. A scheme is witness recovering if the decryption recovers the randomness used to encrypt. For the above scheme however, the vector s is "lost" during decryption. We circumvent this problem by using some sort of key-encapsulation. Instead of encrypting a plaintext-vector m using the above scheme, we encrypt the witness s (which has the same size as m). We will then use another instance of Problem 2 to encrypt the plaintext m (using s as symmetric key). Encrypting the witness s instead of m will not harm security. By Problem 3, the matrix B is pseudorandom. Therefore, the matrix $B + G$ is also pseudorandom. Thus, the second part of the ciphertext $c_2 = Bs + e_2 + Gs = (B + G)s + e_2$ is also pseudorandom by Problem 2. Observe that we do not need the entire secret key T to recover s from a ciphertext c. Let $y = c_2 - Tc_1 = Gs + Xs + e_2 - Te_1$. To recover the i-th component y_i of y, we merely need the i-th row t_i^T of the matrix T. If we posses a sufficient amount of the rows of T, yet not all of them, we can still recover s by computing y_i for all the i for which t_i^T is known and setting $y_i = \perp$ (erasure) otherwise. We can now recover s by performing a combined error- and erasure-correction on y using the decoding algorithm of $\mathcal{C}$. If it is guaranteed that the number of erasures is very low, we can simply set all erasures to random values (thereby introducing a few additional random errors) and use the standard decoding-algorithm $\mathsf{Decode}_{\mathcal{C}}$ of $\mathcal{C}$. Micciancio and Peikert [MP12] recently used a very similar witness-recovering mechanism in their construction of an improved LWE-based IND-CCA2 scheme. While our construction uses off-the-shelf binary error-correcting codes to encode the witness s, they needed to construct a special family of lattices for this purpose. These lattices have a short dual basis and an efficient decoding algorithm, thus they can be seen as a euclidean analogue to efficiently decodable error-correcting codes with large minimum distance. We can now give an outline of our IND-CCA1 construction. It is an adoption of the all-but-one simulation-paradigm [PW08,RS09] to the special structure of our CPA scheme. The key-generation samples not just one, but q (for a constant q) matrices $B_1, \ldots, B_q$ and $T_1, \ldots, T_q$. Encryption first samples a tag τ, then derives an instance-public-key B_τ from $B_1, \ldots, B_q$. It further proceeds as the INC-CPA variant using the matrix B_τ instead of B. The ciphertext is (τ, c). Decryption takes the tag τ, derives an instance secret-key T_τ and uses T_τ to decrypt c. After recovering the random coins it checks whether they suffice a certain hamming-weight criterium. If not, it aborts, otherwise it outputs the plaintext m. The instance-key derivation will assemble the matrix B_τ by picking certain rows from the matrices $B_1, \ldots, B_q$ depending on the tag τ. In the security proof, there will be a single tag τ^* for which the simulator is completely oblivious of the instance-

secret key T_{τ^*} (this is the tag where the IND-CPA challenge will be embedded). For all other tags, the simulator needs to be able to simulate a decryption-oracle. This means that no other instance-secret-key T_τ should share too many rows with T_{τ^*}. If this is the case, the simulator will be able to use an incomplete secret key to answer decryption-queries by the above observation. To guarantee that the instance-secret-keys T_τ have small *overlap* with one another, we will use a q-ary error-correcting encoding for the tags τ. This simulation-strategy requires that the hamming-weight of the ciphertext-noise satisfies a certain bound, otherwise the simulator is unable to correct the additional erasure caused by the incomplete secret key. This is the reason why the decryption needs to check the hamming-weight of the witnesses. The IND-CCA2 construction is obtained by replacing the randomly chosen tags τ with the verification keys of a one-time signature scheme and appending an according signature to the ciphertext. This transformation has been used in several contexts to obtain CCA2 secure encryption from different primitives [DDN00,CHK04,PW08,RS09,DMQN09]. The encryption primitives admitting such a transformation can be generalized under the notion of tag-based encryption schemes [Kil06].

5 The IND-CPA Scheme

In this Section we will provide the full construction of an IND-CPA secure encryption scheme. We will use this scheme in the construction of our CCA1 secure scheme.

Let k be a security parameter, $n \in O(k^{2/(1-2\epsilon)})$, $l_1, l_2, l_3 \in O(k^{2/(1-2\epsilon)})$ and $\rho = O(k^{-(1+2\epsilon)/(1-2\epsilon)})$. Let $G \in \mathbb{F}_2^{l_2 \times n}$ be the generator-matrix of a binary linear error-correcting code $\mathcal{C}$ and $\mathsf{Decode}_\mathcal{C}$ an efficient decoding procedure for $\mathcal{C}$ that corrects up to αl_2 errors (for a constant α). Further let $\mathcal{D} \subseteq \mathbb{F}_2^{l_3}$ be a binary error-correcting code with efficient encoding $\mathsf{Encode}_\mathcal{D}$ and error-correction $\mathsf{Decode}_\mathcal{D}$ that corrects up to λl_3 errors.

Construction 1 *The scheme* $\mathsf{PKE}_1 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is specified by*

- $\mathsf{KeyGen}(1^k)$*: Sample matrices* $A \in \mathbb{F}_2^{l_1 \times n}$ *and* $C \in \mathbb{F}_2^{l_3 \times n}$ *uniformly at random, sample the matrix* T *from* $\chi_\rho^{l_2 \times l_1}$ *and the matrix* X *from* $\chi_\rho^{l_2 \times n}$*. Set* $B = G + T \cdot A + X$*. Set* $pk = (A, B, C)$ *and* $sk = T$*. Output* (pk, sk)*.*
- $\mathsf{Enc}_{pk}(m)$*: Takes a public key* $pk = (A, B, C)$ *and a plaintext* $m \in \mathbb{F}_2^n$ *as input, samples* s *from* χ_ρ^n*,* e_1 *from* $\chi_\rho^{l_1}$*,* e_2 *from* $\chi_\rho^{l_2}$ *and* e_3 *from* $\chi_\rho^{l_3}$*. It sets* $c_1 = A \cdot s + e_1$*,* $c_2 = B \cdot s + e_2$ *and* $c_3 = C \cdot s + e_3 + \mathsf{Encode}_\mathcal{D}(m)$*. Output* $c = (c_1, c_2, c_3)$*.*
- $\mathsf{Dec}_{sk}(c)$*: Takes a secret key* $sk = T$ *and a ciphertext* $c = (c_1, c_2, c_3)$ *as input. Computes* $y = c_2 - T \cdot c_1$ *and* $s = \mathsf{Decode}_\mathcal{C}(y)$*. Outputs* $\perp$ *if decoding fails. Otherwise computes* $m = \mathsf{Decode}_\mathcal{D}(c_3 - C \cdot s)$ *and outputs* m*.*

We will now show that this scheme is correct, i.e. the probability that a decryption-error occurs is negligible in k.

Lemma 1. *The scheme* PKE_1 *is correct.*

Proof. Decryption only fails if one of the two decoding operations fails. We will thus bound the probability of failure for both decoding operations. It holds that

$$y = c_2 - T \cdot c_1 = B \cdot s + e_2 - T(A \cdot s + e_1) = G \cdot s + X \cdot s + e_2 - T \cdot e_1.$$

Thus, it is sufficient to bound the hamming-weight of the error-term $v = X \cdot s + e_2 - T \cdot e_1$. Fix constants $\beta, \gamma > 0$ such that $2\beta + \gamma\rho < \alpha$ and $\gamma\rho < \lambda$. By a Chernoff-bound, it holds that $|s| < \gamma\rho n$, $e_1 < \gamma\rho l_1$, $e_2 < \gamma\rho l_2$ and $e_3 < \gamma\rho l_3$ with overwhelming probability in k. The decoding procedure $\mathsf{Decode}_{\mathcal{C}}$ can correct up to αl_2 errors. With overwhelming probability in k, both matrices X and T are $(\beta, \gamma\rho)$-good (see Section 2.2). Thus it holds that $|Xs| < \beta l_2$ and $|Te_1| < \beta l_2$ (for sufficiently large k). All together, it holds that

$$|v| \leq |Xs| + |e_2| + |Te_1| \leq 2\beta l_2 + \gamma\rho l_2 < \alpha l_2.$$

Therefore, the decoding-procedure $\mathsf{Decode}_{\mathcal{C}}$ will successfully recover s. Moreover, $\mathsf{Decode}_{\mathcal{D}}$ will successfully recover m as $|e_3| < \gamma\rho \cdot l_3 < \lambda l_3$.

We now turn to proof IND-CPA security of the scheme PKE_1.

Theorem 1. *Assume that Problem 2 is hard. Then the scheme* PKE_1 *is IND-CPA secure.*

Proof. Let $\mathcal{A}$ be PPT-bounded IND-CPA adversary against PKE_1. Consider the following sequence of games.

- Game 1: This is the IND-CPA experiment.
- Game 2: This is the same as game 1, except that during key-generation, the matrix B is chosen uniformly at random by the experiment.
- Game 3: The same as game 2, except that during encryption of the challenge-ciphertext, $c^* = (c_1^*, c_2^*, c_3^*)$ is chosen uniformly at random.

Clearly, $\mathcal{A}$'s advantage of winning game 3 is zero, as the challenge-ciphertext c^* is statistically independent of the challenge bit b chosen by the experiment. It remains to show that the views of $\mathcal{A}$ are computationally indistinguishable in game 1, 2 and 3. For contradiction, assume that $\mathcal{A}$ distinguishes game 1 and game 2 with non-negligible advantage $\nu_1(n)$. We will construct a distinguisher $\mathcal{B}_1$ that distinguishes the distributions $(A, T \cdot A + X)$ and (A, U) with advantage $\nu_1(k)$, contradicting the hardness of Problem 3. The input of $\mathcal{B}_1$ is an instance $(A^\dagger, B^\dagger)$. $\mathcal{B}_1$ simulates the interaction with $\mathcal{A}$ in the same way as game 1 does, except for the key generation step. Instead of generating A and B as in game 1, it sets $A = A^\dagger$ and $B = G + B^\dagger$. After the simulation terminates, $\mathcal{B}_1$ outputs whatever $\mathcal{A}$ outputs. Clearly, if $(A^\dagger, B^\dagger)$ is chosen according to $(A, T \cdot A + X)$, then $\mathcal{A}$'s view in $\mathcal{B}_1$'s simulation is identically distributed as in game 1. On the other hand, if $(A^\dagger, B^\dagger)$ is distributed according to (A, U), then $\mathcal{A}$'s view in $\mathcal{B}_1$'s simulation is identical to game 2. Thus it holds that $|\Pr[\mathcal{B}_1(A, TA + X)] - \Pr[\mathcal{B}_1(A, U)]| = |\Pr[\mathsf{view}_{\mathcal{A}}(\mathsf{Game}_1)] - \Pr[\mathsf{view}_{\mathcal{A}}(\mathsf{Game}_2)]| \geq \nu_1(k)$, which contradicts the hardness of Problem 3. Now assume that $\mathcal{A}$ distinguishes between game 2 and game

3 with non-negligible advantage $\nu_2(k)$. We will construct a distinguisher $\mathcal{B}_2$ that distinguishes the distributions $(M, Ms + e)$ and (M, u) with advantage $\nu_2(k)$, contradicting the hardness of Problem 2. Let the input of $\mathcal{B}_2$ be (M, r), where $M \in \mathbb{F}_2^{(l_1+l_2+l_3)\times n}$ and $r \in \mathbb{F}_2^{l_1+l_2+l_3}$. $\mathcal{B}_2$ first partitions M in three matrices $M_1 \in \mathbb{F}_2^{l_1\times n}$, $M_2 \in \mathbb{F}_2^{l_2\times n}$ and $M_3 \in \mathbb{F}_2^{l_3\times n}$. Likewise, it partitions r into $r_1 \in \mathbb{F}_2^{l_1}$, $r_2 \in \mathbb{F}_2^{l_2}$ and $r_3 \in \mathbb{F}_2^{l_3}$. $\mathcal{B}_2$ simulates the interaction with $\mathcal{A}$ exactly like game 2, except for two details. In the key-generation step, it sets $A = M_1$, $B = M_2$ and $C = M_3$. Moreover, the challenge-ciphertext $c^* = (c_1^*, c_2^*, c_3^*)$ by $c_1^* = r_1$, $c_2^* = r_2$ and $c_3^* = r_3 + \mathsf{Encode}_{\mathcal{D}}(m_b)$. After the simulation terminates, $\mathcal{B}_1$ outputs whatever $\mathcal{A}$ outputs. Clearly, if (M, r) is chosen according to $(M, Ms + e)$, then $\mathcal{A}$'s view is identically distributed to game 2. On the other hand, if (M, r) is distributed according to (M, u), then $\mathcal{A}$'s view is identically distributed to game 3. Therefore, it holds that $|\Pr[\mathcal{B}_2(M, Ms+e)] - \Pr[\mathcal{B}_2(M, u)]| = |\Pr[\mathsf{view}_{\mathcal{A}}(\mathsf{Game}_2)] - \Pr[\mathsf{view}_{\mathcal{A}}(\mathsf{Game}_3)]| \geq \nu_2(k)$, which contradicts the hardness of problem 2. This concludes the proof.

6 The IND-CCA1 Scheme

In this Section, we will construct an IND-CCA1 scheme based on the scheme PKE_1 constructed in the last section. We will extend the encryption and decryption algorithms with an instance-key derivation step, that assigns a tag to each ciphertext and derives an instance public or secret key for each tag. These instance-keys will be used as keys for PKE_1. Moreover, we need to ensure that decryption only outputs a plaintext if an incomplete key would have already been sufficient to decrypt. Decryption therefore checks if the hamming-weight of the randomness used to encrypt is small enough. When the scheme is used honestly, this is the case with overwhelming probability. As in the last section, let k be a security parameter, $n \in O(k^{2/(1-2\epsilon)})$, $l_1, l_2, l_3 \in O(k^{2/(1-2\epsilon)})$ and $\rho = O(k^{-(1+2\epsilon)/(1-2\epsilon)})$. Let $G \in \mathbb{F}_2^{l_2\times n}$ be the generator-matrix of a binary linear error-correcting code $\mathcal{C}$ and $\mathsf{Decode}_{\mathcal{C}}$ an efficient decoding procedure that corrects up to αl_2 errors (for a constant α). Let $\mathcal{D} \subseteq \mathbb{F}_2^{l_3}$ be a binary error-correcting code with efficient encoding $\mathsf{Encode}_{\mathcal{D}}$ and error-correction $\mathsf{Decode}_{\mathcal{D}}$ as before. Let $\mathcal{E} \subseteq \Sigma^{l_2}$ be a q-ary code over the alphabet Σ (with $q = |\Sigma|$) with relative minimum-distance δ and dimension n. Such a code can be generated randomly (see Section 2.1). We will now explain how the parameters δ and q must be chosen. Recall that $\mathsf{Decode}_{\mathcal{C}}$ corrects up to αl_2 errors. As explained earlier, α must be big enough to correct the decryption-error, which has hamming-weight less than $(2\beta + \gamma\rho)l_2$ (for any constant $\beta > 0$). As the additional error induced by erasures will have hamming weight $\leq (1 - \delta)l_2$, it is sufficient to choose δ (which must be smaller than $1 - 1/q$) such that $2\beta + \gamma\rho + 1 - \delta < \alpha$. As we can choose β and γ arbitrarily small, we can always find q and δ such that the above is met. Therefore, fix β, γ, q and δ such that for sufficiently large n it holds that $2\beta + \gamma\rho + 1 - \delta < \alpha$. We can choose the constant β arbitrarily small and it holds that $\gamma\rho \in o(1)$. There exist constructions of efficiently decodable linear codes $\mathcal{C}$ such that α is slightly larger than $1/400$ [Zém01]. Thus we can choose q as

small as $q > 1/(\alpha - 2\beta - \gamma\rho) > 400$. We remark that this might be drastically improved if a more sophisticated joint error-and-erasure correction mechanism than ours was used. Our naive mechanism simply treats erasures as errors, but there might be much more efficient mechanism, maybe allowing to choose q as small as 2.

Construction 2 *The scheme* $\mathsf{PKE}_2 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is specified by*

- $\mathsf{KeyGen}(1^k)$*: Sample matrices* $A \in \mathbb{F}_2^{l_1 \times n}$ *and* $C \in \mathbb{F}_2^{l_3 \times n}$ *uniformly at random. For every* $j \in \Sigma$ *sample a matrix* T_j *from* $\chi_\rho^{l_2 \times l_1}$ *and a matrix* X_j *from* $\chi_\rho^{l_2 \times n}$*. Set* $B_j = G + T_j \cdot A + X_j$*. Set* $pk = (A, (B_j)_{j\in\Sigma}, C)$ *and* $sk = (T_j)_{j\in\Sigma}$*. Output* (pk, sk)*.*
- $\mathsf{Enc}_{pk}(m)$*: Takes a public key* $pk = (A, (B_j)_{j\in\Sigma}, C)$ *and a plaintext* $m \in \mathbb{F}_2^n$ *as input. Write each* B_j *as* $B_j = (b_{j,1}, \ldots, b_{j,l_2})^T$ *(The* $b_{j,i}^T$ *are the rows of* B_j*). Sample a tag* $\tau \in \Sigma^n$ *uniformly at random and set* $\hat{\tau} = \mathsf{Encode}_{\mathcal{E}}(\tau)$*. It then sets* $B_{\hat{\tau}} = (b_{\hat{\tau}_1,1}, \ldots, b_{\hat{\tau}_{l_2},l_2})^T$*, i.e. the i-th row of* $B_{\hat{\tau}}$ *is* $b_{\hat{\tau}_i,i}$*. Encryption now samples* s *from* χ_ρ^n*,* e_1 *from* $\chi_\rho^{l_1}$*,* e_2 *from* $\chi_\rho^{l_2}$ *and* e_3 *from* $\chi_\rho^{l_3}$*. It sets* $c_1 = A \cdot s + e_1$*,* $c_2 = B_{\hat{\tau}} \cdot s + e_2$ *and* $c_3 = C \cdot s + e_3 + \mathsf{Encode}_{\mathcal{D}}(m)$*. Output* $c = (\tau, c_1, c_2, c_3)$*.*
- $\mathsf{Dec}_{sk}(c)$*: Takes a secret key* $sk = (T_j)_{j\in\Sigma}$ *and a ciphertext* $c = (\tau, c_1, c_2, c_3)$ *as input. Write each* T_j *as* $T_j = (t_{j,1}, \ldots, t_{j,l_2})^T$ *(The* $t_{j,i}^T$ *are the rows of* T_j*). Then it computes* $\hat{\tau} = \mathsf{Encode}_{\mathcal{E}}(\tau)$ *and* $T_{\hat{\tau}} = (t_{\hat{\tau}_1,1}, \ldots, t_{\hat{\tau}_{l_2},l_2})^T$*. Next it computes* $y = c_2 - T_{\hat{\tau}} \cdot c_1$ *and* $s = \mathsf{Decode}_{\mathcal{C}}(y)$*. Outputs* $\perp$ *if decoding fails. Otherwise compute* $m = \mathsf{Decode}_{\mathcal{D}}(c_3 - C \cdot s)$*. Now it computes* $e_1 = c_1 - A \cdot s$*,* $e_2 = c_2 - B_{\hat{\tau}} \cdot s$*,* $e_3 = c_3 - C \cdot s - \mathsf{Encode}_{\mathcal{D}}(m)$ *and checks whether* $|s| < \gamma\rho n$*,* $|e_1| < \gamma\rho l_1$*,* $|e_2| < \gamma\rho l_2$ *and* $|e_3| < \gamma\rho l_3$*. If yes it outputs* m*, otherwise* $\perp$*.*

Correctness of PKE_2 follows immediately from the correctness of PKE_1. The only additional step is the check of the hamming weights $|s|$, $|e_1|$, $|e_2|$ and $|e_3|$. However, this has been dealt with implicitly in Lemma 1. We will now prove IND-CCA1 security for the scheme PKE_2.

Theorem 2. *The scheme* PKE_2 *is IND-CCA1 secure, provided that the scheme* PKE_1 *is IND-CPA secure and the parameters* α*,* β*,* γ*,* q *and* δ *suffice* $\delta < 1 - 1/q$ *and* $2\beta + \gamma\rho + 1 - \delta < \alpha$*.*

Proof. Let $\mathcal{A}$ be PPT-bounded IND-CPA adversary against PKE_2. Consider the following sequence of games.

- Game 1: This is the IND-CCA1 experiment.
- Game 2: This is the same as game 1, except that the tag τ^* of the challenge-ciphertext $c^* = (\tau^*, c_1^*, c_2^*, c_3^*)$ is chosen before the experiment starts, and game 2 aborts if $\mathcal{A}$ sends a decryption-query with tag τ^*.
- Game 3 This is the same as game 2, except that the decryption-oracle is implemented differently. For a decryption-query $c = (\tau, c_1, c_2, c_3)$ the decryption-oracle proceeds as follows. Let $\hat{\tau} = \mathsf{Encode}_{\mathcal{E}}(\tau)$. For all $i \in \{1, \ldots, l_2\}$ with $\hat{\tau}_i \neq \hat{\tau}_i^*$, it computes $y_i = c_{2,i} - t_{\hat{\tau}_i,i}^T c_1$. For all remaining

i it chooses y_i uniformly at random. The decryption-oracle then continues like in game 2, computing $s = \mathsf{Decode}_{\mathcal{C}}(y)$ (and aborts if decoding fails) and $m = \mathsf{Decode}_{\mathcal{D}}(c_3 - C \cdot s)$, setting $e_1 = c_1 - A \cdot s$, $e_2 = c_2 - B \cdot s$, $e_3 = c_3 - C \cdot s - \mathsf{Encode}_{\mathcal{D}}(m)$ and checking whether $|s| < \gamma\rho n$, $|e_1| < \gamma\rho l_1$, $|e_2| < \gamma\rho l_2$ and $|e_3| < \gamma\rho l_3$. If yes it outputs m, otherwise $\perp$.

In game 2, the event that $\mathcal{A}$ sends a decryption-query with tag τ^* has probability at most $f(k)/q^n = \mathsf{negl}(k)$, where $f(k)$ is a polynomial upper bound for the number of decryption-queries $\mathcal{A}$ makes. If this event does not occur, game 1 and game 2 are identically distributed from $\mathcal{A}$'s view. Thus, from $\mathcal{A}$'s view game 1 and game 2 are statistically indistinguishable. We will now show that game 2 and game 3 are statistically indistinguishable from $\mathcal{A}$'s view. First, assume that for every tag τ the matrices $T_{\hat{\tau}}$ and $X_{\hat{\tau}}$ are $(\beta, \gamma\rho)$-good. If this is the case, we claim that the decryption oracles of game 2 and game 3 behave identical. We split the claim in two cases. The first case is simple: If either $|s| \geq \gamma\rho n$, $|e_1| \geq \gamma\rho l_1$, $|e_2| \geq \gamma\rho l_2$ or $|e_3| \geq \gamma\rho l_3$, then the decryption oracle will return $\perp$ in both games, regardless whether decoding fails or not. In the other case it holds that $|s| < \gamma\rho n$, $|e_1| < \gamma\rho l_1$, $|e_2| < \gamma\rho l_2$ and $|e_3| < \gamma\rho l_3$. Now it holds that the hamming-weight of the error-term $v = X_{\hat{\tau}} \cdot s + e_2 - T_{\hat{\tau}} \cdot e_1$ will be bounded by $2\beta l_2 + \gamma\rho l_2$. Thus, in game 2 the decoding-algorithm $\mathsf{Decode}_{\mathcal{C}}$ has to correct at most $(2\beta + \gamma\rho)l_2 < \alpha l_2$ and will thus be successful and output the *unique* s. In game 3, there might be up to $(1 - \delta)l_2$ additional errors $\mathsf{Decode}_{\mathcal{C}}$ has to deal with, as the decryption oracle chooses up to $(1 - \delta)l_2$ components of the codeword y at random. However, since $(2\beta + \gamma\rho + 1 - \delta)l_2 < \alpha l_2$ the decoding-algorithm $\mathsf{Decode}_{\mathcal{C}}$ will also succeed in game 3 and output the unique s. This concludes the claim. What remains to show for this part of the proof is that, with overwhelming probability in k, it holds that for every tag τ the matrices $T_{\hat{\tau}}$ and $X_{\hat{\tau}}$ are $(\beta, \gamma\rho)$-good. We can think of each matrix $T_{\hat{\tau}}$ as a row-sub-matrix of a large matrix $T_{full} \in \mathbb{F}_2^{ql_2 \times n}$ that consists of all the rows of all T_i for $i \in \Sigma$ (i.e. T_{full} is just the vertical concatenation of all T_i). With overwhelming probability in k, T_{full} is $(\beta/q, \gamma\rho)$-good (since q is constant). This means that for each e_1 with $|e_1| < \gamma\rho l_1$ it holds that $|T_{full}e_1| < \beta/q \cdot (ql_2) = \beta l_2$. However, as each $T_{\hat{\tau}}$ is a row-sub-matrix of T_{full}, it also holds that $|T_{\hat{\tau}}e_1| < \beta l_2$. Showing that $|X_{\hat{\tau}}s| < \beta l_2$ works analogously, which concludes this part of the proof. Finally, $\mathcal{A}$'s advantage of winning game 3 is negligible in k, given that PKE_1 is IND-CPA secure. Assume for contradiction that $\mathcal{A}$ wins game 3 with non-negligible advantage $\nu(k)$. We will construct an IND-CPA adversary $\mathcal{B}$ against PKE_1 that wins the IND-CPA experiment with advantage ν. $\mathcal{B}$'s input from the IND-CPA experiment is a public key $pk' = (A', B', C')$ for the scheme PKE_1. $\mathcal{B}$ now runs the key-generation of game 3 with the following modifications. Instead of sampling the matrices A and C uniformly at random, it sets $A = A'$ and $C = C'$. Now it generates the B_j and T_j exactly like the key-generation in game 3. Then however, it replaces the public-key at the locations that constitute $B_{\hat{\tau}}$ with B', i.e. it sets $b^T_{\hat{\tau}_i,i} = b'^T_i$ for $i = 1, \ldots, l_2$. $\mathcal{B}$. Then it simulates the interaction between $\mathcal{A}$ and game 3, answering decryption-queries like game 3. This is possible, as game 3 never uses secret keys $t_{\hat{\tau}_i,i}$ (that correspond to public keys $b_{\hat{\tau}_i,i}$) to

answer decryption queries. Once $\mathcal{A}$ sends challenge messages (m_0, m_1), $\mathcal{B}$ forwards (m_0, m_1) to the IND-CPA experiment and receives a challenge-ciphertext $c^\dagger = (c_1^\dagger, c_2^\dagger, c_2^\dagger)$. $\mathcal{B}$ sends $c^* = (\tau^*, c_1^\dagger, c_2^\dagger, c_2^\dagger)$ to $\mathcal{A}$ and continues the simulation. Once $\mathcal{A}$ terminates, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. From $\mathcal{A}$'s view, $\mathcal{B}$'s simulation and game 3 are perfectly indistinguishable, as the distributions of A and C are the same, as well as the distribution of the partial public keys $b_{j,i}^T$, which are independent of one another (only depending on the same A). Moreover, the decryption-oracle behaves identically in both experiments. Therefore, it holds that $\mathsf{Adv}_{\text{IND-CPA}}(\mathcal{B}) = \mathsf{Adv}_{\text{IND-CCA1}}(\mathcal{A}) = \nu(k)$ which contradicts the IND-CPA security of scheme PKE_1.

7 The IND-CCA2 Scheme

We will now provide details how the scheme PKE_2 can be transformed into an IND-CCA2 secure scheme PKE_3 using additional one-time signatures. We follow an approach by Dolev, Dwork and Naor [DDN00], which has been used in several other constructions [PW08,Pei09,RS09,DMQN09,MP12], especially in the world of lattice and coding assumptions, to achieve full CCA2 security. First observe that it is not necessary to choose the tag $\tau \in \Sigma^n$ uniformly at random in the encryption procedure of PKE_2. We only need to guarantee that a PPT-adversary $\mathcal{A}$ will have negligible probability guessing the secret tag τ^* correctly if it is granted a polynomial number of trials (this immediately yields the statistical indistinguishability of game 1 and game 2 in Theorem 2). Thus it is sufficient to sample the tags τ from a distribution with high min-entropy. Moreover, observe that the proof of Theorem 2 still holds if we allow $\mathcal{A}$ to make decryption-queries even after it has received the challenge-ciphertext c^*. This can be seen by noting that the decryption-oracle in game 3 can answer decryption-queries with $\tau \neq \tau^*$ regardless of whether the challenge-ciphertext has been given to $\mathcal{A}$ or not (decryption-queries with $\tau = \tau^*$ are rejected unconditionally). In fact, the decryption-oracle in game 3 is oblivious of whether the challenge-ciphertext has been given to $\mathcal{A}$ or not. Thus, the scheme PKE_2 can be recast as a tag-based encryption scheme [Kil06]. We will now outline PKE_3. Let $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be an EUF-CMA secure one-time signature scheme. For simplicity, assume that the verification-keys vk of SIG are elements of Σ^n (this can always be accomplished by encoding vk in the q-ary alphabet Σ and choosing n large enough). The key-generation of PKE_3 is identical to the key-generation of PKE_2. The encryption procedure $\mathsf{PKE}_3.\mathsf{Enc}$ first computes a pair of verification and signature-keys $(vk, sgk) = \mathsf{SIG.Gen}(1^k)$. Then it runs the encryption procedure $\mathsf{PKE}_2.\mathsf{Enc}$, with the difference that it sets $\tau = vk$ instead of choosing τ uniformly at random. Let c' be the output of $\mathsf{PKE}_2.\mathsf{Enc}$. $\mathsf{PKE}_3.\mathsf{Enc}$ then computes $\sigma = \mathsf{SIG.Sign}_{sgk}(c')$ and outputs the ciphertext $c = (c', \sigma)$. The decryption procedure $\mathsf{PKE}_3.\mathsf{Dec}$ first checks if σ is a valid signature on c' using the verification-key $vk = \tau$ (where τ is the tag given in c'). If the check succeeds, it runs the decryption procedure $\mathsf{PKE}_2.\mathsf{Dec}$ on the ciphertext c' and outputs whatever $\mathsf{PKE}_2.\mathsf{Dec}$ outputs. We summarize this in the following construction.

Let $\mathsf{Enc}'_{pk}(m, vk)$ be a procedure that does exactly the same as $\mathsf{PKE}_2.\mathsf{Enc}_{pk}(m)$, but sets $\tau = vk$ instead of choosing τ uniformly at random.

Construction 3 *The scheme* $\mathsf{PKE}_3 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is specified by*

- $\mathsf{KeyGen}(1^k)$*: Compute* $(pk, sk) = \mathsf{PKE}_2.\mathsf{KeyGen}(1^k)$ *and output* (pk, sk).
- $\mathsf{Enc}_{pk}(m)$*: Generate* $(vk, sgk) = \mathsf{SIG.Gen}(1^k)$*, encrypt* $c' = \mathsf{Enc}'_{pk}(m, vk)$*, sign* $\sigma = \mathsf{SIG.Sign}_{sgk}(c')$ *and output* $c = (c', \sigma)$.
- $\mathsf{Dec}_{sk}(c)$*: Let* $c = (c', \sigma)$ *and* $c' = (\tau, c_1, c_2, c_3)$*. Set* $vk = \tau$*. Check if* $\mathsf{SIG.Verify}_{vk}(c', \sigma) = 1$*, if not abort. Otherwise compute* $m = \mathsf{PKE}_2.\mathsf{Dec}_{sk}(c')$ *and output* m.

Theorem 3. *The scheme* PKE_3 *is IND-CCA2 secure, provided that* SIG *is an EUF-CMA secure one-time signature scheme and the same requirements as in Theorem 2 are given.*

Proof. (Sketch) Let $\mathcal{A}$ be PPT-bounded IND-CCA2 adversary against PKE_3. It suffices to show that with overwhelming probability, every decryption-query by $\mathcal{A}$ tagged with τ^* (the tag of the challenge-ciphertext) is rejected. Thus, we can recycle the proof of Theorem 5 almost entirely, we only need to replace the indistinguishability of game 1 and game 2 in the proof of Theorem 2. The rest of the proof is identical. Consider the following two games.

- Game 1: This is the IND-CCA2 experiment.
- Game 2: This is the same as game 1, except that the tag τ^* of the challenge-ciphertext $c^* = (\tau^*, c_1^*, c_2^*, c_3^*, \sigma^*)$ is generated before the experiment starts, and game 2 aborts if $\mathcal{A}$ sends a decryption-query with tag τ^*.

Assume that $\mathcal{A}$ distinguishes between game 1 and game 2 with non-negligible advantage $\nu(k)$. Clearly, given that the decryption-oracle rejects every decryption-query tagged with τ^*, both games are identically distributed from $\mathcal{A}$'s view. Thus, to distinguish game 1 and game 2 $\mathcal{A}$ must generate a decryption-query tagged with τ^* that is accepted by the decryption-oracle. This implies that such a decryption-query $c = (c', \sigma)$ with $c' = (\tau^*, c_1, c_2, c_3)$ suffices the condition $\mathsf{SIG.Verify}_{vk}(c', \sigma) = 1$, where $vk = \tau^*$. Thus we can assume that $\mathcal{A}$ generates such a decryption-query with probability $\nu(k)$. We construct an EUF-CMA adversary $\mathcal{B}$ that breaks the EUF-CMA security of SIG with probability $\nu(k)$. Let vk be the verification key provided to $\mathcal{B}$ by the EUF-CMA experiment. $\mathcal{B}$ simulates game 2 with $\mathcal{A}$, but makes the following changes. Instead of generating the tag τ^* itself, it sets $\tau^* = vk$. Moreover, $\mathcal{B}$ obtains the signature σ^* of the challenge-ciphertext c^* by querying its signature-oracle with c'^*, where $c'^* = \mathsf{Enc}'_{pk}(m_b, vk)$. Finally, once $\mathcal{A}$ sends a decryption-query $c = (c', \sigma)$ with $c' = (\tau^*, c_1, c_2, c_3)$ and $\mathsf{SIG.Verify}_{vk}(c', \sigma) = 1$, $\mathcal{B}$ outputs (c', σ) an terminates. Clearly, game 2 and the simulation of $\mathcal{B}$ are identically distributed from the view of $\mathcal{A}$. Thus, the event that $\mathcal{A}$ sends a decryption-query $c = (c', \sigma)$ with $c' = (\tau^*, c_1, c_2, c_3)$ and $\mathsf{SIG.Verify}_{vk}(c', \sigma) = 1$ happens with probability $\nu(k)$ in $\mathcal{B}$'s simulation. This means that $\mathcal{B}$ outputs a valid forged signature with probability $\nu(k)$, contradicting the EUF-CMA security of SIG.

8 Conclusion

In this work we constructed the first IND-CCA2 secure public key encryption scheme based solely on the hardness of a low-noise variant of the learning parity with noise problem. To achieve this, we introduced a novel all-but-one simulation technique. This new technique enabled the construction of a CCA1 secure scheme, which is more efficient than any previous such construction based on the correlated-products approach. The scheme enjoys a constant-factor ciphertext expansion as well as asymptotically efficient key-generation, encryption and decryption.

9 Acknowledgement

The authors would like to thank the anonymous reviewers of ASIACRYPT 2012 for providing helpful comments and pointing out insightful connections to related work. Nico Döttling was supported by IBM Research & Development Germany within the HomER-project.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307, 2003.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1 + 1 = 0 improves information set decoding. In *EUROCRYPT*, pages 520–536, 2012.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In *CRYPTO*, pages 743–760, 2011.

[BS08] Bhaskar Biswas and Nicolas Sendrier. Mceliece cryptosystem implementation: Theory and practice. In *PQCrypto*, pages 47–62, 2008.

[CC98] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO*, pages 126–144, 2003.

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

[DMQN09] Rafael Dowsley, Jörn Müller-Quade, and Anderson C. A. Nascimento. A cca2 secure public key encryption scheme based on the mceliece assumptions in the standard model. In *CT-RSA*, pages 240–251, 2009.

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

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *ASIACRYPT*, pages 52–66, 2001.

[HK09] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT*, pages 313–332, 2009.

[JW05] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In *CRYPTO*, pages 293–308, 2005.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and concurrent security of the hb and hb$^+$ protocols. *J. Cryptology*, 23(3):402–421, 2010.

[Lam79] Leslie Lamport. Constructing digital signatures from one-way functions. *SRI intl. CSL-98*, 1979.

[Lin03] Yehuda Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. In *EUROCRYPT*, pages 241–254, 2003.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. In *DSN Progress Report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA*, page , 1978.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in o(2 0.054n). In *ASIACRYPT*, pages 107–124, 2011.

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

[NIKM08] Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Des. Codes Cryptography*, 49(1-3):289–305, 2008.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

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

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications*, pages 106–113, 1988.

[Zém01] Gilles Zémor. On expander codes. *IEEE Transactions on Information Theory*, 47(2):835–837, 2001.