Low Noise LPN: KDM Secure Public Key Encryption and Sample Amplification*

Nico Döttling†‡

Dept. of Computer Science, Aarhus University

January 12, 2015

Abstract

Cryptographic schemes based on the Learning Parity with Noise (LPN) problem have several very desirable aspects: Low computational overhead, simple implementation and conjectured post-quantum hardness. Choosing the LPN noise parameter sufficiently low allows for public key cryptography. In this work, we construct the first standard model public key encryption scheme with key dependent message security based solely on the low noise LPN problem. Additionally, we establish a new connection between LPN with a bounded number of samples and LPN with an unbounded number of samples. In essence, we show that if LPN with a small error and a small number of samples is hard, then LPN with a slightly larger error and an unbounded number of samples is also hard. The key technical ingredient to establish both results is a variant of the LPN problem called the extended LPN problem.

Keywords: Low Noise LPN, Key Dependent Message Security, LPN Hardness Reduction

1 Introduction

The LPN Problem The learning parity with noise (LPN) problem asks to find a secret binary vector $\mathbf{s} \in \mathbb{F}_2^n$ given noisy linear samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{F}_2^n \times \mathbb{F}_2$ where $\mathbf{a}$ is chosen uniformly at random and e is an additive noise term that occurs with probability ρ. Due to its simplicity and binary arithmetic, the LPN problem has become a central hub in secret key cryptography [13, 30, 32, 29, 35]. These applications use the *high noise* LPN problem where the noise rate $\rho < 1/2$ is a constant. In the *low noise* LPN problem, the noise rate ρ tends asymptotically to 0. Alekhnovich [6] provided a construction of a public key encryption scheme based on LPN for noise rates $\rho = O(1/\sqrt{n})$. Recently, more complex cryptographic primitives have been constructed from low noise LPN such as chosen ciphertext secure public key encryption [21, 33] and composable oblivious transfer [18]. In the original formulation of the LPN problem, the search algorithm/adversary may demand an unbounded number of samples whereas the bounded

†Supported by European Research Commission Starting Grant no. 279447.

‡The authors acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed; and also from the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed.

samples version (e.g. used in [6, 21, 33]) only provides an a priori bounded number of samples to the search algorithm. So far, it was unknown whether the hardness of LPN with a bounded number of samples implies the hardness of LPN with an unbounded number of samples, even if a modest increase in the noise rate is tolerated.

Key Dependent Message Security A public key encryption scheme is called key dependent message (KDM) secure, if encryptions of the secret key, or more generally encryptions of functions of several secret keys are indistinguishable of encryptions of (say) the all-zero message. We will exclusively consider KDM-CPA security in this work, i.e. KDM adversaries do not have access to a decryption oracle. While for most natural cryptographic tasks standard notions of security are sufficient, the notion of KDM security is relevant for contexts such as computational soundness [12, 2] or when hard-disks are encrypted that store the corresponding secret key (as mentioned in [15]). It has been shown that standard IND-CPA (or even IND-CCA) security does not imply KDM security [1, 16], i.e. there exist public key encryption schemes with IND-CPA security relative to some standard assumption which are provably not KDM secure. Standard model KDM secure public key public key cryptosystems were constructed from a variety of assumptions, starting with the construction of Boneh et al [15]. Applebaum et al. [10] provided both a circular secure public key encryption scheme from the LWE assumption and a circular secure *private key* encryption scheme from the (high noise) LPN problem. The latter scheme was later shown to fulfill the stronger notion of related-key KDM security by Applebaum [9]. In [8], Applebaum provided a construction of a KDM secure PKE for arbitrary (bounded size) circuits from any KDM secure PKE for affine functions. Constructing a KDM secure public key encryption scheme from low noise LPN has remained an open problem so far.

1.1 Extended LPN

The central tool we use in our constructions is a version of the LPN problem called *extended decisional LPN problem*, or eDLPN in short. The eDLPN problem can be seen as a special case for $q = 2$ of the extended LWE problem introduced O'Neill, Peikert and Waters [37] and proven hard under standard LWE by Alperin-Sheriff and Peikert [7]. The binary version we use in this work was first discussed by Kiltz, Masny and Pietrzak [33].

In the eDLPN problem, the adversary's goal is to distinguish $(\mathbf{A}, \mathbf{RA}, \mathbf{e}, \mathbf{Re})$ from $(\mathbf{A}, \mathbf{U}, \mathbf{e}, \mathbf{Re})$, where $\mathbf{A}$ is a randomly chosen matrix, $\mathbf{R}$ is a randomly chosen low weight matrix, $\mathbf{U}$ is a randomly chosen matrix and $\mathbf{e}$ follows some distribution $\boldsymbol{\chi}$. This is similar to the dual formulation of the decisional LPN problem, where the adversary has to distinguish $(\mathbf{A}, \mathbf{RA})$ from $(\mathbf{A}, \mathbf{U})$. However, in the extended decisional LPN problem, the adversary obtains an extra advice $\mathbf{Re}$ about a secret matrix $\mathbf{R}$, where the vector $\mathbf{e}$ can have any distribution. Kiltz, Masny and Pietrzak [33] observed that in the LPN case, this advice can be extremely useful to enable reductions to simulate faithfully. In particular, the eDLPN problem can effectively be used as a computational substitute for the (generalized) leftover hash lemma [31, 19] or gaussian regularity lemmata for lattices [26].

In Section 3, we provide a generalization of the extended LPN problem we call leaky LPN (ℓ-LPN), which may be of independent interest. In the ℓ-LPN problem, the advice given to the adversary can be described by an arbitrary adversarially chosen leakage function γ from a family $\mathcal{L}$ and is not limited to linear functions as in the extended LPN problem. Clearly, the hardness of the extended LPN problem follows immediately from the hardness of the leaky LPN problem when instantiating the leakage functions with linear functions. If the functions in $\mathcal{L}$ output short strings, say strings of at most logarithmic length, then the hardness of the ℓ-LPN *search* problem follows immediately from the standard LPN problem, since all possible leakage values can be efficiently

enumerated (or guessed). The situation is slightly different for decisional problems. In general, decisional problems become easy if even a single bit of arbitrary leakage is allowed. However, we only allow the leakage to depend on $\mathbf{R}$ and in particular not on $\mathbf{A}$. We show that a sample preserving search to decision reduction of Applebaum et al. [11] is in fact *leakage preserving*. We can thus base the hardness of the decisional problem ℓ-DLPN on ℓ-LPN, and therefore on LPN given that the functions in $\mathcal{L}$ only provide short advice.

1.2 KDM Secure Public Key Encryption

We will now provide an overview of our construction of a KDM secure public key encryption scheme from LPN. The construction is inspired by the public key encryption scheme of Applebaum et al. [10], which however lives in the LWE realm. The basic idea, as in [10], is to make encryptions of the secret key syntactically similar to the public key. More specifically, public keys in our scheme will be of the form $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ where $\mathbf{s}$ is the secret key. It follows immediately from the decisional LPN problem that the public key is pseudorandom. Encryption takes a message $\mathfrak{m}$ and computes

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{RA} \\ \mathbf{c}_2 &= \mathbf{Ry} + \mathbf{G}\mathfrak{m}, \end{aligned}$$

where the matrix $\mathbf{R}$ is chosen from a low weight distribution and $\mathbf{G}$ is the generator matrix of a good, efficiently decodable binary linear code. We remark that while this scheme bears strong resemblances with (and is inspired by) the LWE based scheme of [10], it is rather incomparable to the (high noise) LPN based private key encryption schemes of [10, 9] or previous low-noise LPN public key encryption schemes [6, 21, 33]. Notice that standard IND-CPA security of this scheme follows directly from the fact that $\mathbf{y}$ is pseudorandom and thus also $(\mathbf{RA}, \mathbf{Ry})$ is pseudorandom given the public key $(\mathbf{A}, \mathbf{y})$, by using the dual formulation of the decisional LPN problem (i.e. $(\mathbf{A}', \mathbf{RA}') \approx_c (\mathbf{A}', \mathbf{U})$). To decrypt a ciphertext $\mathfrak{c} = (\mathbf{C}_1, \mathbf{c}_2)$, we basically compute

$$\mathbf{z} = \mathbf{c}_2 - \mathbf{C}_1\mathbf{s}$$

and recover $\mathfrak{m}$ from $\mathbf{z}$ by using the efficient decoding algorithm for the code generated by $\mathbf{G}$. Correctness of the scheme follows from the fact that

$$\begin{aligned} \mathbf{z} &= \mathbf{c}_2 - \mathbf{C}_1\mathbf{s} \\ &= \mathbf{Ry} + \mathbf{G}\mathfrak{m} - \mathbf{RAs} \\ &= \mathbf{R}(\mathbf{As} + \mathbf{e}) + \mathbf{G}\mathfrak{m} - \mathbf{RAs} \\ &= \mathbf{G}\mathfrak{m} + \mathbf{Re}. \end{aligned}$$

Since we have chosen $\mathbf{R}$ and $\mathbf{e}$ from low noise distributions, the term $\mathbf{Re}$ has low weight with high probability. Thus it follows that a decoder of the code generated by $\mathbf{G}$ will be able to recover $\mathfrak{m}$ from $\mathbf{z}$. We will briefly sketch how to establish 1-circular security of this scheme, where the adversary gets a single encryption of the secret key (or an encryption of 0). For the full proof of KDM security for affine functions, refer to Section 4. An encryption of the secret key has the form $(\mathbf{RA}, \mathbf{Ry} + \mathbf{Gs})$. Figure 1 provides the game transform for this security reduction.

The first three steps shown in Figure 1 do not change the real experiment but basically rewrite the challenge ciphertext. From step 3 to step 4 we replace the matrix $\mathbf{RA}$ by a uniformly random matrix $\mathbf{U}$. Since we also need the additional term $\mathbf{Re}$ to provide the correct distribution to the adversary, we will use the extended decisional LPN problem to show that these two experiments are

	Game	public key	challenge ciphertext	remark
1.	Real	$(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$	$(\mathbf{RA}, \mathbf{Ry} + \mathbf{Gs})$	
2.	Real	$(\mathbf{A}, \mathbf{As} + \mathbf{e})$	$(\mathbf{RA}, \mathbf{R}(\mathbf{As} + \mathbf{e}) + \mathbf{Gs})$	identical
3.	Real	$(\mathbf{A}, \mathbf{As} + \mathbf{e})$	$(\mathbf{RA}, (\mathbf{RA} + \mathbf{G})\mathbf{s} + \mathbf{Re})$	identical
4.	$\mathbf{H}_1$	$(\mathbf{A}, \mathbf{As} + \mathbf{e})$	$(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{Re})$	eDLPN
5.	$\mathbf{H}_1$	$(\mathbf{A}, \mathbf{As} + \mathbf{e})$	$(\mathbf{U}' - \mathbf{G}, \mathbf{U}'\mathbf{s} + \mathbf{Re})$	identical
6.	$\mathbf{H}_2$	$(\mathbf{A}, \mathbf{As} + \mathbf{e})$	$(\mathbf{RA} - \mathbf{G}, \mathbf{RAs} + \mathbf{Re})$	eDLPN
7.	$\mathbf{H}_2$	$(\mathbf{A}, \mathbf{As} + \mathbf{e})$	$(\mathbf{RA} - \mathbf{G}, \mathbf{R}(\mathbf{As} + \mathbf{e}))$	identical
8.	$\mathbf{H}_3$	$(\mathbf{A}, \mathbf{u})$	$(\mathbf{RA} - \mathbf{G}, \mathbf{Ru})$	DLPN
9.	$\mathbf{H}_3$	$(\mathbf{A}, \mathbf{u})$	$(\mathbf{U} - \mathbf{G}, \mathbf{u}')$	DDLPN
10.	$\mathbf{H}_3$	$(\mathbf{A}, \mathbf{u})$	$(\mathbf{U}, \mathbf{u}')$	identical

Figure 1: The Game Transform for KDM-CPA security

computationally indistinguishable. In particular, we use $\mathbf{A}$ and $\mathbf{e}$ provided by the eDLPN problem to construct a public key, while we use $\mathbf{RA}$ and the advice $\mathbf{Re}$ to construct the encryption of the secret key. Then, we replace $\mathbf{RA}$ by a random matrix $\mathbf{U}$, which yields an indistinguishable experiment by the hardness of eDLPN. Step 4 to 5 is another bridging step which does not change the experiment. Since $\mathbf{U}$ is distributed uniformly random, so is the matrix $\mathbf{U}' = \mathbf{U} + \mathbf{G}$. Thus, instead of choosing $\mathbf{U}$ uniformly at random we can choose $\mathbf{U}'$ uniformly at random and set $\mathbf{U} = \mathbf{U}' - \mathbf{G}$. From step 5 to step 6 we replace the matrix $\mathbf{U}'$ by $\mathbf{RA}$. Again, we have to use the extended decisional LPN problem as we need the extra advice $\mathbf{Re}$. It now becomes clear that we have used steps 3 to 6 to *pull* the matrix $\mathbf{G}$ from the second component of the challenge ciphertext to its first component, i.e. we have transformed $(\mathbf{RA}, (\mathbf{RA}+\mathbf{G})\mathbf{s}+\mathbf{Re})$ into $(\mathbf{RA}-\mathbf{G}, \mathbf{RAs}+\mathbf{Re})$. Step 6 to step 7 is another basic bridging step. From step 7 to step 8 we replace the second component $\mathbf{As}+\mathbf{e}$ of the public key by a randomly chosen $\mathbf{u}$, indistinguishability follows from the standard decisional LPN problem. From step 8 to step 9 we replace $(\mathbf{A}, \mathbf{RA}, \mathbf{u}, \mathbf{Ru})$ by $(\mathbf{A}, \mathbf{U}, \mathbf{u}, \mathbf{u}')$ for uniformly random $\mathbf{U}$ and $\mathbf{u}'$, indistinguishability follows from the dual formulation of the decisional LPN problem. Finally, from step 9 to step 10, we replace $\mathbf{U} - \mathbf{G}$ by $\mathbf{U}$. We can do this since the uniform distribution $\mathbf{U}$ is invariant under an additive shift by a constant matrix $\mathbf{G}$. Thus, in the last experiment the challenge ciphertext is just uniformly random, which concludes this outline.

1.3 Unbounded Samples LPN from Bounded Samples LPN

In the following we will distinguish between bounded and unbounded samples LPN. We will denote search LPN with a secret of length n, m samples and noise rate ρ by $\mathsf{LPN}(n, m, \rho)$ and decisional LPN with a secret of length n, unbounded samples and noise rate ρ' by $\mathsf{DLPN}(n, \rho')$. Our second contribution is a hardness reduction which bases the hardness of $\mathsf{DLPN}(n, \rho')$ on $\mathsf{LPN}(n, 2n, \rho)$. More specifically, we show that if $\mathsf{LPN}(n, 2n, \rho)$ is hard, then $\mathsf{DLPN}(n, \rho')$ is also hard, where

$$\rho' = \frac{1}{2} - \frac{1}{2}(1 - 2\rho)^{\lfloor \rho 2n \rfloor} \leq 2\rho^2 n.$$

For the Learning With Errors (LWE) problem, there exists a statistical *random self reduction* [26, 10]. The idea of this reduction is to use $m \approx n \log(q)$ *seed samples* to generate arbitrarily many fresh samples. The noise rate in the new samples increases only slightly. Specifically, if $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{z})$ is such a given set of *seed samples*, then one can generate new samples by drawing $\mathbf{e} \in \mathbb{Z}_q^m$ from a discrete gaussian [5, 36] and setting $\mathbf{a}' = \mathbf{A}^\top \mathbf{e}$ and $y' = \mathbf{e}^\top \mathbf{y}$. Now it holds

$$y' = \mathbf{e}^\top \mathbf{y} = \mathbf{e}^\top \mathbf{As} + \mathbf{e}^\top \mathbf{z} = \mathbf{a}'^\top \mathbf{s} + \langle \mathbf{e}, \mathbf{z} \rangle.$$

The pair $(\mathbf{a}', y' + e')$, where e' is a gaussian smoothing term, is a proper LWE sample, as $\mathbf{a}' = \mathbf{e}^\top \mathbf{A}$ can be shown to be statistically close to uniform *and* $\langle \mathbf{e}, \mathbf{z} \rangle$ follows an independent discrete gaussian distribution even conditioned on $\mathbf{a}' = \mathbf{e}^\top \mathbf{A}$ [1].

Such an approach, however, cannot be directly transferred to the LPN setting. For the vector $\mathbf{a}' = \mathbf{A}^\top \mathbf{e}$ to be statistically close to uniform, $\mathbf{e}$ must have min-entropy $\approx n$, and thus high weight. But this in turn means that $\langle \mathbf{e}, \mathbf{z} \rangle$ will only have a negligibly small bias. We remark that such a high noise sample amplification was used Lyubashevsky [34] to cryptanalize LPN in sub-exponential time, but this technique does not seem to be applicable in the context of an *efficient* (i.e. PPT) hardness reduction, especially when the number of samples is at most polynomial.

Therefore, in our reduction we will replace the statistical tools in the above reduction by a computational technique based on the eDLPN problem. Again, we start with a given amount of $m = 2n$ seed samples and generate new samples from these. While we cannot hope that the samples we generate in this way have the proper distribution (in the statistical sense), we will be able to show that the distribution generated in this way is computationally indistinguishable from the real LPN distribution. More specifically, let $(\mathbf{A}, \mathbf{y})$ be the LPN seed samples. We will compute new samples by choosing a random low weight $\mathbf{r}$ and setting $\mathbf{a} = \mathbf{A}^\top \mathbf{r}$ and $y' = \mathbf{r}^\top \mathbf{y} = \langle \mathbf{r}, \mathbf{y} \rangle$. Now, assume first that $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$. Then it holds that

$$y' = \mathbf{r}^\top \mathbf{A}\mathbf{s} + \langle \mathbf{r}, \mathbf{e} \rangle = \langle \mathbf{a}, \mathbf{s} \rangle + \langle \mathbf{r}, \mathbf{e} \rangle.$$

While $(\mathbf{a}, y')$ syntactically looks like an LPN sample, it is statistically far away from a correctly distributed sample. There are two issues. First, $\mathbf{a} = \mathbf{r}^\top \mathbf{A}$ is not distributed uniformly. Second, the noise term $\langle \mathbf{r}, \mathbf{e} \rangle$ is correlated with $\mathbf{a}$. The first issue *alone* could be resolved by assuming the hardness of the DLPN. To deal with both issues simultaneously, we will resort to the eDLPN problem, which allows us to present a noise term $\langle \mathbf{r}, \mathbf{e} \rangle$ with the right distribution. More specifically, the eDLPN problem allows us to replace $\mathbf{a} = \mathbf{r}^\top \mathbf{A}$ by a uniformly random $\mathbf{a}$ but also provides us with an *advice* $(\mathbf{e}, \langle \mathbf{r}, \mathbf{e} \rangle)$ that allows us to simulate the noise term $\langle \mathbf{r}, \mathbf{e} \rangle$ correctly. On the other hand, if $\mathbf{y}$ was chosen uniformly at random, then the pseudorandomness of $(\mathbf{a}, y') = (\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{y})$ follows easily from the dual formulation of the LPN problem DLPN. Since we can base the hardness of all auxiliary problems on $\mathsf{LPN}(n, 2n, \rho)$, it follows that $\mathsf{DLPN}(n, \rho')$ is at least as hard as $\mathsf{LPN}(n, 2n, \rho)$. This concludes this outline.

2 Preliminaries

In the following, let λ always denote the security parameter. We call a machine PPT if it runs in probabilistic (expected) polynomial time. For a search problem P and an adversary/search algorithm $\mathcal{A}$ let $\mathsf{Adv}_{\mathsf{P}}(\mathcal{A})$ denote the probability of $\mathcal{A}$ finding a solution of a random instance of P. For a decisional problem D which consists in distinguishing two distributions X and Y and a distinguishing algorithm $\mathcal{D}$ define $\mathsf{Adv}_{\mathsf{D}}(\mathcal{D}) = |\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(Y) = 1]|$. When we don't write it explicitly, we will implicitly assume that search algorithms and distinguishers get 1^λ as an additional input. We will denote the Hamming weight of a vector $\mathbf{x} \in \mathbb{F}_2^n$ by $\|\mathbf{x}\|_0 = |\{i : x_i \neq 0\}|$. For a matrix $\mathbf{M} \in \mathbb{F}_2^{m \times n}$, we define the Hamming weight of $\mathbf{M}$ by $\|\mathbf{M}\|_0 = \max_i \|\mathbf{m}_i\|_0$ where the $\mathbf{m}_i$ are the column vectors of $\mathbf{M}$. It follows easily for all $\mathbf{M} \in \mathbb{F}_2^{m \times n}$ and $\mathbf{x} \in \mathbb{F}_2^n$ that $\|\mathbf{M}\mathbf{x}\|_0 \leq \|\mathbf{M}\|_0 \cdot \|\mathbf{x}\|_0$. We need asymptotically good, efficiently decodable binary linear codes for the construction of our KDM secure public key encryption scheme. A binary linear $[k, n]$ code C is

[1] This can be established via a Lemma due to Regev [39] or its refinement due to Peikert [38], which show that the distribution of $\mathbf{e}$ remains discrete gaussian even conditioned on $\mathbf{e}^\top \mathbf{A}$, though the variance of the distribution decreases.

a n dimensional subspace of $\mathbb{F}_2^k$. We call $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ a generator matrix of C if every $\mathbf{c} \in \mathsf{C}$ can be written as $\mathbf{c} = \mathbf{G}\mathbf{x}$ for some $\mathbf{x} \in \mathbb{F}_2^n$. We assume codes C come with efficient encoding and decoding procedures $\mathsf{C.Encode}$ and $\mathsf{C.Decode}$, where $\mathsf{C.Encode}(\mathbf{x}) = \mathbf{G} \cdot \mathbf{x}$ for some generator matrix $\mathbf{G}$ of C. An error correcting code can efficiently correct an α fraction of errors, if for every $\mathbf{e} \in \mathbb{F}_2^k$ with $\|\mathbf{e}\|_0 \leq \alpha k$, it holds that $\mathsf{C.Decode}(\mathsf{C.Encode}(\mathbf{x}) + \mathbf{e}) = \mathbf{x}$. There exists a large corpus of literature of linear codes that can efficiently correct a constant fraction of errors, for instance concatenated codes [25] or expander codes [40, 41].

2.1 Learning Parity with Noise

We will denote the Bernoulli distribution with parameter $\rho \in [0, 1/2]$ on $\mathbb{F}_2^m$ by $\mathsf{Ber}(m, \rho)$. For an $\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$, each component e_i of $\mathbf{e}$ independently takes the value 1 with probability ρ and 0 with probability $1 - \rho$. We write $\mathsf{Ber}(\rho) := \mathsf{Ber}(1, \rho)$. We will distinguish between LPN with a bounded and an unbounded number of samples.

Definition 1 (Learning Parity with Noise)**.** Let $\boldsymbol{\chi}$ be an error distribution on $\mathbb{F}_2^m$ and $\rho = \rho(\lambda) \in [0, 1/2]$. Let $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$ be chosen uniformly at random, let $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ be chosen uniformly at random and let $\mathbf{e} \leftarrow_\$ \boldsymbol{\chi}$.

1. In the bounded samples search problem $\mathsf{LPN}(n, m, \boldsymbol{\chi})$, the goal is to find $\mathbf{s}$, given $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$.
2. In the unbounded samples search problem $\mathsf{LPN}(n, \rho)$, the goal is to find $\mathbf{s}$, given an oracle that outputs an arbitrary number of samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$ and $e \leftarrow_\$ \mathsf{Ber}(\rho)$.
3. In the bounded samples decisional problem $\mathsf{DLPN}(n, m, \boldsymbol{\chi})$, the goal is to distinguish the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{u} \leftarrow_\$ \mathbb{F}_2^m$ is chosen uniformly at random.
4. In the unbounded samples decisional problem $\mathsf{DLPN}(n, \rho)$, the goal is to distinguish two oracles, namely one that outputs samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ (where $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$ and $e \leftarrow_\$ \mathsf{Ber}(\rho)$) from one that outputs samples of the form $(\mathbf{a}, u)$ (where $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$ and $u \leftarrow_\$ \mathbb{F}_2$).

For bounded samples LPN with errors $\mathbf{e}$ from the Bernoulli distribution $\mathsf{Ber}(m, \rho)$ we will write $\mathsf{LPN}(n, m, \rho)$ for $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$ and also $\mathsf{DLPN}(n, m, \rho)$ for $\mathsf{DLPN}(n, m, \mathsf{Ber}(m, \rho))$. By a standard argument, one can show that if $\mathbf{e} \in \mathbb{F}_2^m$ is distributed according to $\mathsf{Ber}(m, \rho)$ and $\mathbf{z} \in \mathbb{F}_2^m$ is an arbitrary vector of weight $\lfloor \rho m \rfloor$, then $\langle \mathbf{z}, \mathbf{e} \rangle$ is distributed according to $\mathsf{Ber}(\rho')$, where $\rho' = \frac{1}{2} - \frac{1}{2}(1 - 2\rho)^{\lfloor \rho m \rfloor} \leq \rho^2 m$. Following Alekhnovich [6], we will choose the noise parameter ρ of the form $O(1/\sqrt{n})$ and $n, m = \Omega(\lambda^2)$ to be able to use low weight vectors as trapdoors and have 2^λ (conjectured) security for $\mathsf{LPN}(n, m, \rho)$.

A series of works have established relations between search and decisional LPN problems [13, 32, 11]. The hardness reduction of Applebaum et al. [11] is sample preserving, i.e. it shows that the hardness of $\mathsf{DLPN}(n, m, \boldsymbol{\chi})$ follows directly from the hardness of $\mathsf{LPN}(n, m, \boldsymbol{\chi})$, for any error distribution $\boldsymbol{\chi}$.

Lemma 1 (Applebaum et al. [11])**.** *Let $\boldsymbol{\chi}$ be an error distribution on $\mathbb{F}_2^m$ and assume that $\mathsf{LPN}(n, m, \boldsymbol{\chi})$ is hard. Then $\mathsf{DLPN}(n, m, \boldsymbol{\chi})$ is also hard. More specifically, assume there exists a PPT adversary $\mathcal{A}$ that distinguishes $\mathsf{DLPN}(n, m, \boldsymbol{\chi})$ with advantage ϵ. Then there exists a PPT adversary $\mathcal{A}'$ that breaks $\mathsf{LPN}(n, m, \boldsymbol{\chi})$ with advantage $\epsilon^2/8$.*

Let $\mathsf{S}(m, \rho)$ denote the distribution on $\mathbb{F}_2^m$ which outputs uniformly random vectors in $\mathbb{F}_2^m$ of weight $\lfloor \rho m \rfloor$, i.e. $\mathsf{S}(m, \rho)$ is the uniform distribution on the set $M = \{\mathbf{x} \in \mathbb{F}_2^m \mid \|\mathbf{x}\|_0 = \lfloor \rho m \rfloor\}$. It is easy to see that if $\mathsf{LPN}(n, m, \rho)$ is hard, then $\mathsf{LPN}(n, m, \mathsf{S}(m, \rho))$ is also hard.

Corollary 2. *Let $\mathcal{A}$ be a PPT adversary that breaks $\mathsf{LPN}(n, m, \mathsf{S}(m, \rho))$ with advantage ϵ. Then there exists a PPT adversary $\mathcal{A}'$ that breaks $\mathsf{LPN}(n, m, \rho)$ with advantage $\frac{(1-o(1))\epsilon}{\sqrt{2\pi m\rho(1-\rho)}}$. Moreover, if there exists a PPT distinguisher $\mathcal{D}$ that distinguishes $\mathsf{DLPN}(n, m, \mathsf{S}(m, \rho))$ with advantage ϵ, then there exists a PPT adversary $\mathcal{A}'$ that breaks $\mathsf{LPN}(n, m, \rho)$ with advantage $\frac{(1-o(1))\epsilon^2}{8\cdot\sqrt{2\pi m\rho(1-\rho)}}$.*

Proof. We show that if $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ is an instance of $\mathsf{LPN}(n, m, \rho)$, then with high probability $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ is also an instance of $\mathsf{LPN}(n, m, \mathsf{S}(m, \rho))$. Since both problems are search problems, a solution can easily be verified. First note that we can safely assume that $\rho = \omega(1/n)$, as for $\rho = O(1/n)$ $\mathsf{LPN}(n, m, \rho)$ is trivially easy. As $\rho = \omega(1/n)$, an error vector $\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$ has weight $\lfloor \rho m \rfloor$ with probability $\frac{(1-o(1))}{\sqrt{2\pi m\rho(1-\rho)}}$ by the deMoivre-Laplace binomial point mass limit theorem (see e.g. [24]). Moreover, conditioned to $\|\mathbf{e}\|_0 = \lfloor \rho m \rfloor$ it holds that $\mathbf{e}$ follows exactly the distribution $\mathsf{S}(m, \rho)$. The hardness of the decisional problem follows by Lemma 1. □

As a convenient reformulation of the LPN problem, we define the decisional dual LPN problem.

Definition 2. Let $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m\times n}$, $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$, $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^m$, $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}$ and $\mathbf{u} \leftarrow_\$ \mathbb{F}_2^k$. The goal of the $\mathsf{DDLPN}(n, m, k, \rho)$ problem is to distinguish the distributions $(\mathbf{A}, \mathbf{RA}, \mathbf{a}, \mathbf{Ra})$ and $(\mathbf{A}, \mathbf{U}, \mathbf{a}, \mathbf{u})$.

The hardness of $\mathsf{DDLPN}(n, m, k, \rho)$ follows from $\mathsf{DLPN}(n, m, \rho)$ (see e.g. [10] or [21]) using the fact that for a randomly chosen matrix $\mathbf{A}$ we can also sample a random $\mathbf{H}$ such that it holds $\mathbf{H} \cdot \mathbf{A} = \mathbf{0}$ and $\mathbf{H}$ is uniformly random (not given $\mathbf{A}$).

Lemma 3. *Let $m \geq 2n$. Assume there exists a PPT distinguisher $\mathcal{D}$ that distinguishes the problem $\mathsf{DDLPN}(n, m, k, \rho)$ with advantage ϵ. Then there exists a PPT adversary $\mathcal{A}$ that breaks $\mathsf{LPN}(n, m, \rho)$ with advantage $\frac{\epsilon^2}{8k^2}$.*

Following Kiltz et al. [33] and Alperin-Sheriff and Peikert [7], we provide a definition of the extended LPN problem. We only define the extended LPN problem for Bernoulli error distributions.

Definition 3. Let χ be any distribution on $\mathbb{F}_2^m$. Let $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m\times n}$, $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$, $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}$ and $\mathbf{e} \leftarrow_\$ \boldsymbol{\chi}$. The goal of the $\mathsf{eDLPN}(n, m, k, \rho, \boldsymbol{\chi})$ problem is to distinguish the distributions $(\mathbf{A}, \mathbf{RA}, \mathbf{e}, \mathbf{Re})$ and $(\mathbf{A}, \mathbf{U}, \mathbf{e}, \mathbf{Re})$.

In Section 3 we will establish the hardness of $\mathsf{eDLPN}(n, m, k, \rho, \boldsymbol{\chi})$ from $\mathsf{LPN}(n, m, \rho)$.

2.2 Key Dependent Message Secure Public Key Encryption

Syntactically, a public key encryption scheme PKE consists of three PPT algorithms $\mathsf{PKE.KeyGen}$, $\mathsf{PKE.Enc}$ and $\mathsf{PKE.Dec}$, such that $\mathsf{PKE.KeyGen}$ generates a pair (pk, sk) of public and secret keys, $\mathsf{PKE.Enc}$ takes a public key pk and a plaintext $\mathfrak{m}$ and outputs a ciphertext $\mathfrak{c}$ and $\mathsf{PKE.Dec}$ takes a secret key sk and a ciphertext $\mathfrak{c}$ and outputs a plaintext $\mathfrak{m}$. We say that PKE is correct, if it holds for all plaintexts $\mathfrak{m}$ (of size corresponding to λ) that if $(pk, sk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, then

$$\mathsf{PKE.Dec}(sk, \mathsf{PKE.Enc}(pk, \mathfrak{m})) = \mathfrak{m},$$

except with negligible probability over the randomness used by $\mathsf{PKE.KeyGen}$, $\mathsf{PKE.Enc}$ and $\mathsf{PKE.Dec}$. The security notion we consider in this work is key dependent message security under chosen plaintext attacks. In the security experiment corresponding to this notion, the adversary gets a list of public keys $\{pk_i\}$ and access to an oracle that computes encryptions of functions of the secret keys. We call such dependencies *key cycles*, even though the functional relationships the adversary obtains can be more complex than key cycles.

Definition 4. We say a public key encryption-scheme PKE is ciphertext indistinguishable under key dependent message chosen plaintext attacks (KDM-CPA) for cycles of length l with respect to a class $\mathcal{F}$ of functions mapping l secret keys to a plaintext, if every PPT-adversary $\mathcal{A}$ has success-probability at most negligibly better than $1/2$ in the experiment $\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}$, i.e. $\Pr[\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}(\mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$.

Experiment $\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}$	$\mathcal{O}_{\mathsf{KDM}}(f, j)$
For $i = 1, \ldots, l$	If $b = 0$
$(pk_i, sk_i) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$	$f \leftarrow \mathbf{0}$
$b \leftarrow_\$ \{0, 1\}$	$\mathfrak{c} \leftarrow \mathsf{PKE.Enc}(pk_j, f(\{sk_i\}))$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$	Return $\mathfrak{c}$
Return 1 iff $b = b'$.	

Remark 4. We implicitly assume that sanity checks are performed by the oracle, i.e. it only accepts KDM queries with $f \in \mathcal{F}$ and $j \in \{1, \ldots, l\}$. Moreover, we assume that the KDM oracle may have access to all local variables of the experiment $\mathsf{KDM\text{-}CPA}$, in particular the pk_i and sk_i and the bit b.

Applebaum [8] provides a general transfomation which transforms any public key encryption scheme with KDM security against *affine functions* into a public key encryption scheme with KDM security against arbitrary functions with circuits of *bounded size*. Thus, it is sufficient to construct a public key encryption scheme with KDM security against affine functions to obtain a scheme with security against the more general class of functions.

3 Leaky LPN

Kiltz, Masny and Pietrzak [33], following Alperin-Sheriff and Peikert [7], suggested to use the extended LPN problem for cryptographic constructions. We will provide a generalization of the extended LPN problem which we call *leaky LPN*. While in the extended LPN problem the adversary obtains some linear auxiliary information about the LPN error term $\mathbf{e}$, the leaky LPN problem allows this auxiliary information to be modeled by a general leakage function.

Definition 5 (Leaky LPN)**.** Let $\mathcal{L}$ be a family of (randomized) functions $\mathbb{F}_2^m \to \{0,1\}^\ell$. Let $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$ be chosen uniformly at random, let $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ be chosen uniformly at random and let $\mathbf{e} \leftarrow_\$ \boldsymbol{\chi}$.

1. The problem $\ell\text{-}\mathsf{LPN}(n, m, \boldsymbol{\chi}, \mathcal{L})$ is to find $\mathbf{s}$, given $(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e}))$ for an arbitrary $\gamma \in \mathcal{L}$.
2. The problem $\ell\text{-}\mathsf{DLPN}(n.m, \boldsymbol{\chi}, \mathcal{L})$ is to distinguish the distributions $(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e}))$ and $(\mathbf{A}, \mathbf{u}, \gamma(\mathbf{e}))$ for an arbitrary $\gamma \in \mathcal{L}$.

We can think of $\ell\text{-}\mathsf{LPN}$ and $\ell\text{-}\mathsf{DLPN}$ as games between an adversary/distinguisher and an experiment in the following way. First the adversary specifies an $\gamma \in \mathcal{L}$ of its choice to the experiment and then receives $(\mathbf{A}, \mathbf{y}, z)$. Now the adversary either has to guess a solution or distinguish (depending on whether he is playing against $\ell\text{-}\mathsf{LPN}$ or $\ell\text{-}\mathsf{DLPN}$).

We observe that the search-to-decision reduction of Applebaum et al. [11] can be easily adapted to the case of leaky LPN. This yields that $\ell\text{-}\mathsf{LPN}$ and $\ell\text{-}\mathsf{DLPN}$ are in fact equivalent.

Lemma 5. *Assume there exists a PPT distinguisher $\mathcal{D}$ that distinguishes ℓ-$\mathsf{DLPN}(n, m, \boldsymbol{\chi}, \mathcal{L})$ with advantage ϵ. Then there exists a PPT algorithm $\mathcal{A}$ that solves ℓ-$\mathsf{LPN}(n, m, \boldsymbol{\chi}, \mathcal{L})$ with advantage $\epsilon^2/8$.*

Proof. Assume towards contradiction that there exists a PPT-algorithm $\mathcal{D}$ and an $\gamma \in \mathcal{L}$ such that $\mathcal{D}$ distinguishes $(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e}))$ and $(\mathbf{A}, \mathbf{u}, \gamma(\mathbf{e}))$ with non-negligible advantage ϵ. Thus assume that

$$\mathsf{Adv}_{\ell\text{-}\mathsf{DLPN}}(\mathcal{D}) = |\Pr[\mathcal{D}(\mathbf{A}, \mathbf{u}, \gamma(\mathbf{e})) = 1] - \Pr[\mathcal{D}(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e})) = 1]| \geq \epsilon.$$

We will construct an algorithm $\mathcal{A}'$ that computes the Goldreich-Levin hardcore-bit $\langle \mathbf{r}, \mathbf{s} \rangle$ of $\mathbf{s}$, given $(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e}))$ with advantage $\epsilon/2$. By the Goldreich Levin theorem [28, 27][2] this algorithm $\mathcal{A}'$ can be used to construct a PPT algorithm $\mathcal{A}$ that solves the leaky LPN search problem with advantage $\frac{1}{2}\left(\frac{\epsilon}{2}\right)^2$. $\mathcal{A}'$ is given as follows.

Adversary $\mathcal{A}'$
Input: $(\mathbf{A}, \mathbf{y}, z) \in \mathbb{F}_2^{m \times n} \times \mathbb{F}_2^m$ and $\mathbf{r} \in \mathbb{F}_2^n$
$\mathbf{u} \leftarrow_\$ \mathbb{F}_2^m$
$\mathbf{B} \leftarrow \mathbf{A} - \mathbf{u} \cdot \mathbf{r}^\top$
$b \leftarrow \mathcal{D}(\mathbf{B}, \mathbf{y}, z)$
return b

We will now analyze the success probability of $\mathcal{A}'$. Let $(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e}))$ and $\mathbf{r}$ be $\mathcal{A}'$'s input. First notice that since $\mathbf{A}$ is distributed uniformly at random, so is $\mathbf{B} = \mathbf{A} - \mathbf{ur}^\top$, as the uniform distribution is shift invariant. Moreover, since $\mathbf{A} = \mathbf{B} + \mathbf{u} \cdot \mathbf{r}^\top$, it holds that $\mathbf{y} = \mathbf{As} + \mathbf{e} = \mathbf{Bs} + \mathbf{ur}^\top\mathbf{s} + \mathbf{e} = \mathbf{Bs} + \mathbf{u}\langle \mathbf{r}, \mathbf{s} \rangle + \mathbf{e}$. On one hand, if $\langle \mathbf{r}, \mathbf{s} \rangle = 0$, then $(\mathbf{B}, \mathbf{y}, z)$ has the distribution $(\mathbf{B}, \mathbf{Bs} + \mathbf{e}, \gamma(\mathbf{e}))$. On the other hand, if $\langle \mathbf{r}, \mathbf{s} \rangle = 1$, then $\mathbf{y} = \mathbf{Bs} + \mathbf{e} + \mathbf{u}$. As $\mathbf{u}$ is uniformly distributed (independently of $\mathbf{B}$, $\mathbf{s}$ and $\mathbf{e}$), $\mathbf{y}$ is also uniformly distributed. Thus, $(\mathbf{B}, \mathbf{y}, z)$ has the distribution $(\mathbf{B}, \mathbf{u}', \gamma(\mathbf{e}))$, for uniformly chosen $\mathbf{u}'$. We conclude that

$$\begin{aligned}
\mathsf{Adv}_{\mathsf{GL}}(\mathcal{A}') &= \left|\Pr[\mathcal{A}'((\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e})), \mathbf{r}) = \langle \mathbf{r}, \mathbf{s} \rangle] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\Pr[\mathcal{A}'((\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e})), \mathbf{r}) = 0 | \langle \mathbf{r}, \mathbf{s} \rangle = 0]\right. \\
&\quad \left. + \frac{1}{2}\Pr[\mathcal{A}'((\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e})), \mathbf{r}) = 1 | \langle \mathbf{r}, \mathbf{s} \rangle = 1] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\Pr[\mathcal{D}(\mathbf{B}, \mathbf{Bs} + \mathbf{e}, \gamma(\mathbf{e})) = 0] + \frac{1}{2}\Pr[\mathcal{D}(\mathbf{B}, \mathbf{u}', \gamma(\mathbf{e})) = 1] - \frac{1}{2}\right| \\
&= \frac{1}{2}\left|\Pr[\mathcal{D}(\mathbf{B}, \mathbf{u}', \gamma(\mathbf{e})) = 1] - \Pr[\mathcal{D}(\mathbf{B}, \mathbf{Bs} + \mathbf{e}, \gamma(\mathbf{e})) = 1]\right| \\
&= \mathsf{Adv}_{\ell\text{-}\mathsf{DLPN}}(\mathcal{D}) \geq \epsilon/2,
\end{aligned}$$

i.e. $\mathcal{A}'$ computes the Goldreich-Levin hardcore bit $\langle \mathbf{r}, \mathbf{s} \rangle$ with advantage $\frac{\epsilon}{2}$. Thus, by the Goldreich-Levin theorem there exists a PPT adversary that breaks ℓ-$\mathsf{LPN}(n, m, \boldsymbol{\chi}, \mathcal{L})$ with advantage $\epsilon^2/8$. □

We will now establish hardness of $\mathsf{eDLPN}(n, m, k, \rho, \boldsymbol{\chi})$ from $\mathsf{LPN}(n, m, \rho)$ using leaky LPN.

[2] We use the more efficient reduction for the Goldreich Levin hardcore bit provided by Proposition 2.5.4 in [27], where the runtime of the reduction is independent of the adversaries advantage (up to a logarithmic factor, which can be upper bounded by λ).

Lemma 6. *Let $m \geq 2n$. Then for any distribution $\boldsymbol{\chi}$ on $\mathbb{F}_2^m$ and any $k = \mathsf{poly}(\lambda)$ it holds that if there exists a PPT distinguisher $\mathcal{D}$ that distinguishes $\mathsf{eDLPN}(n, m, k, \rho, \boldsymbol{\chi})$ with advantage ϵ, then there exists a PPT adversary $\mathcal{A}$ that breaks $\mathsf{LPN}(n, m, \rho)$ with advantage $\frac{\epsilon^2}{8k^2}$.*

Proof Sketch. The hardness of $\mathsf{eDLPN}(n, m, k, \rho, \boldsymbol{\chi})$ follows easily from $\mathsf{eDLPN}(n, m, 1, \rho, \boldsymbol{\chi})$ by a k-step hybrid argument. Choosing $\mathcal{F}$ as a family of randomized functions mapping $\mathbf{r}$ to $(\mathbf{z}, \mathbf{z}^\top \mathbf{r})$, where $\mathbf{z} \leftarrow_\$ \boldsymbol{\chi}$, it holds that $\ell\text{-}\mathsf{LPN}(n, m, \rho, \mathcal{F})$ is at least as hard as $\mathsf{LPN}(n, m, \rho)$, as the function output $\mathbf{z}^\top \mathbf{r}$ is just one bit, i.e. we can run the reduction for both possible choices of $\mathbf{z}^\top \mathbf{r}$. By Lemma 5, it follows that $\ell\text{-}\mathsf{DLPN}(n, m, \rho, \mathcal{F})$ is at least as hard as $\ell\text{-}\mathsf{LPN}(n, m, \rho, \mathcal{F})$, i.e. we have that

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{r}, \mathbf{z}, \mathbf{z}^\top \mathbf{r}) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{z}^\top \mathbf{r}),$$

where $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$, $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{r} \leftarrow_\$ \mathsf{Ber}(m, \rho)$, $\mathbf{z} \leftarrow_\$ \boldsymbol{\chi}$ and $\mathbf{u} \leftarrow_\$ \mathbb{F}_2^m$. As $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$ is chosen uniformly at random, we can sample a matrix $\mathbf{A}' \in \mathbb{F}_2^{m \times (m-n)}$ with $\mathbf{A}'^\top \cdot \mathbf{A} = \mathbf{0}$ such that $\mathbf{A}'$ is statistically close to uniform. Thus, it holds that

$$(\mathbf{A}', \mathbf{A}'^\top \mathbf{r}, \mathbf{z}, \mathbf{z}^\top \mathbf{r}) = (\mathbf{A}', \mathbf{A}'^\top \cdot (\mathbf{A}\mathbf{s} + \mathbf{r}), \mathbf{z}, \mathbf{z}^\top \mathbf{r}) \approx_c (\mathbf{A}', \mathbf{u}, \mathbf{z}, \mathbf{z}^\top \mathbf{r}).$$

We can reformulate this to

$$(\mathbf{A}', \mathbf{r}^\top \mathbf{A}', \mathbf{z}, \mathbf{r}^\top \mathbf{z}) \approx_c (\mathbf{A}', \mathbf{u}^\top, \mathbf{z}, \mathbf{r}^\top \mathbf{z}),$$

which is the statement of $\mathsf{eDLPN}(m - n, m, 1, \rho, \boldsymbol{\chi})$. As $m - n \geq n$, any instance of $\mathsf{eDLPN}(m - n, m, 1, \rho, \boldsymbol{\chi})$ can be *truncated* to an instance of $\mathsf{eDLPN}(n, m, 1, \rho, \boldsymbol{\chi})$. We can thus establish the hardness of $\mathsf{eDLPN}(n, m, 1, \rho, \boldsymbol{\chi})$ from $\mathsf{LPN}(n, m, \rho)$. The overall advantage of the LPN adversary is $\frac{\epsilon^2}{8k^2}$, if the advantage of the eDLPN adversary is ϵ. □

4 KDM Secure Public Key Encryption from Low Noise LPN

In this section we will provide a public key encryption scheme with KDM security for affine functions based on the hardness of the low noise LPN problem.

Construction 1. Let $n, m, k = \mathsf{poly}(\lambda)$ be positive integers with $m > k > n$. Let C be binary linear code of length k and dimension n and efficient encoding and decoding procedures $\mathsf{C.Encode}$ and $\mathsf{C.Decode}$. The public key encrypion scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is given by the following algorithms. The message space of PKE is $\mathbb{F}_2^n$.

$\mathsf{PKE.KeyGen}(1^\lambda)$:
- $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$
- $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
- $\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
- $\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$
- $pk \leftarrow (\mathbf{A}, \mathbf{y})$
- $sk \leftarrow \mathbf{s}$
- Return (pk, sk)

$\mathsf{PKE.Enc}(pk, \mathfrak{m})$:
- Parse $pk = (\mathbf{A}, \mathbf{b})$
- $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
- $\mathbf{C}_1 \leftarrow \mathbf{R} \cdot \mathbf{A}$
- $\mathbf{c}_2 \leftarrow \mathbf{R} \cdot \mathbf{y} + \mathsf{C.Encode}(\mathfrak{m})$
- $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
- Return $\mathfrak{c}$

$\mathsf{PKE.Dec}(sk, \mathfrak{c})$:
- Parse $\mathfrak{c} = (\mathbf{C}_1, \mathbf{c}_2)$ and $sk = \mathbf{s}$
- $\mathbf{z} \leftarrow \mathbf{c}_2 - \mathbf{C}_1\mathbf{s}$
- $\mathfrak{m} \leftarrow \mathsf{C.Decode}(\mathbf{z})$
- Return $\mathfrak{m}$

4.1 Correctness

We will first show that the scheme PKE is correct.

Lemma 7. *Assume that* $\mathsf{C.Decode}$ *can efficiently decode from* $\rho' = 4\rho^2 km$ *errors. Then the scheme* PKE *is correct.*

The condition of Lemma 7 can be met by choosing $m, k = \Omega(n)$ and $\rho = O(1/\sqrt{n})$. Thus, to obtain conjectured 2^λ-hardness for LPN, we can take usual parameter choices of $m, n, k = \Theta(\lambda^2)$ and $\rho = \Theta(1/\lambda)$ (as in [6, 21, 33]).

Proof. Assume that $\mathfrak{c} = (\mathbf{C}_1, \mathbf{c}_2)$ is a ciphertext generated by $\mathsf{PKE.Enc}$. Consider the term $\mathbf{z}$ computed during decryption. It holds that

$$\begin{aligned}\mathbf{z} &= \mathbf{c_2} - \mathbf{C}_1\mathbf{s} \\ &= \mathbf{R}\mathbf{y} + \mathsf{C.Encode}(\mathfrak{m}) - \mathbf{R}\mathbf{A}\mathbf{s} \\ &= \mathsf{C.Encode}(\mathfrak{m}) + \mathbf{R}(\mathbf{A}\mathbf{s} + \mathbf{e}) - \mathbf{R}\mathbf{A}\mathbf{s} \\ &= \mathsf{C.Encode}(\mathfrak{m}) + \mathbf{R}\mathbf{e}\end{aligned}$$

By a Chernoff bound, it holds that $\|\mathbf{e}\| \leq 2\rho m$, except with negligible probability $e^{-\frac{1}{3}\rho m}$. Also by a Chernoff bound and a union bound, it holds that $\|\mathbf{R}\|_0 \leq 2\rho k$, except with negligible probability $m \cdot e^{-\frac{1}{3}\rho k}$. Therefore,

$$\|\mathbf{R}\mathbf{e}\|_0 \leq \|\mathbf{R}\|_0 \cdot \|\mathbf{e}\|_0 \leq 4\rho^2 km,$$

except with negligible probability over the choice of $\mathbf{e}$ and $\mathbf{R}$. Consequently, $\mathsf{C.Decode}$ will be able to decode $\mathfrak{m}$ from $\mathbf{z}$. □

4.2 KDM-CPA security

We will now prove KDM-CPA security of PKE.

Theorem 1. *Let* λ *be a security parameter and* $n, m, k, l = \mathsf{poly}(\lambda)$ *with* $m \geq 2n$ *and* $l \geq 1$. *Let* $\rho = \rho(\lambda) \in [0, 1/2]$. *Let* $\mathcal{F} = \{f : (\mathbb{F}_2^n)^l \to \mathbb{F}_2^n\}$ *be a family of affine functions. If* $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$, $\mathsf{DLPN}(n, l \cdot m, \rho)$ *and* $\mathsf{DDLPN}(n, m, k, \rho)$ *are hard, then the scheme* PKE *is* $\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}$ *secure. More precisely, assume that* $\mathcal{A}$ *is a PPT adversary that breaks the* $\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}$ *security of* PKE *with advantage* $\mathsf{Adv}_{\mathsf{KDM\text{-}CPA}}(\mathcal{A})$ *and queries its KDM oracle at most* $q = \mathsf{poly}(\lambda)$ *times. Then there exist PPT distinguishers* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *against the problem* $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$, $\mathcal{D}_3$ *against* $\mathsf{DLPN}(n, l \cdot m, \rho)$ *and* $\mathcal{D}_4$ *against* $\mathsf{DDLPN}(n, m, k, \rho)$ *such that*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{KDM\text{-}CPA}}(\mathcal{A}) \leq\ & lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_2) \\ & + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_3) + lq \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_4).\end{aligned}$$

Corollary 8. *Let* n, m, k, l, ρ *and* $\mathcal{F}$ *be as in Theorem 1. If* $\mathsf{LPN}(n, l \cdot m, \rho)$ *is hard, then* PKE *is* $\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}$ *secure. More precisely, assume that* $\mathcal{A}$ *is a PPT adversary that breaks the* $\mathsf{KDM\text{-}CPA}_{\mathcal{F},l}$ *of* PKE *with advantage* ϵ *and queries its KDM oracle at most* $q = \mathsf{poly}(\lambda)$ *times. Then there exists a PPT adversary* $\mathcal{A}^*$ *that solves* $\mathsf{LPN}(n, l \cdot m, \rho)$ *with advantage* $\frac{\epsilon^2}{128k^2l^2q^2}$.

Proof of Corollary 8. Let $\epsilon = \mathsf{Adv}_{\mathsf{KDM\text{-}CPA}}(\mathcal{A})$. Theorem 1 provides us with PPT distinguishers $\mathcal{D}_1$ and $\mathcal{D}_2$ against $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$, $\mathcal{D}_3$ against $\mathsf{DLPN}(n, l \cdot m, \rho)$ and $\mathcal{D}_4$ against $\mathsf{DDLPN}(n, m, k, \rho)$ such that

$$\epsilon \leq lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_2) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_3) + lq \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_4). \tag{1}$$

By Lemma 6 there exists a PPT adversary $\mathcal{A}_1$ against $\mathsf{LPN}(n, m, \rho)$ (and thus also against $\mathsf{LPN}(n, l \cdot m, \rho)$) such that

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_1) \geq \frac{\mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1)^2}{8k^2}.$$

By the same argument, there exists a PPT adversary $\mathcal{A}_2$ against $\mathsf{LPN}(n, m, \rho)$ (and thus also against $\mathsf{LPN}(n, l \cdot m, \rho)$) such that

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2) \geq \frac{\mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_2)^2}{8k^2}.$$

By Lemma 1 there exists a PPT adversary $\mathcal{A}_3$ against $\mathsf{LPN}(n, l \cdot m, \rho)$ such that

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_3) \geq \frac{\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_3)^2}{8}.$$

Finally, by Lemma 3 there exists an adversary $\mathcal{A}_4$ against $\mathsf{LPN}(n, m, \rho)$ (and thus also against $\mathsf{LPN}(n, l \cdot m, \rho)$) such that

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2) \geq \frac{\mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_4)^2}{8k^2}.$$

Now, define the adversary $\mathcal{A}^*$ as follows. On input $(\mathbf{A}, \mathbf{y})$ $\mathcal{A}^*$ runs the algorithms $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$ on input $(\mathbf{A}, \mathbf{y})$ in parallel, checks the results of all of them (i.e. checks for each candidate solution $\mathbf{s}$ whether $\|\mathbf{y} - \mathbf{A}\mathbf{s}\| \approx \rho lm$) and outputs a solution if one has been found. Clearly, it holds that

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}^*) \geq \max_{i=1,2,3,4} \mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_i),$$

as $\mathcal{A}^*$'s advantage in solving $\mathsf{LPN}(n, l \cdot m, \rho)$ is at least as big as the advantage of the best $\mathcal{A}_i$ (for a given λ). We claim that

$$\max_{i=1,2,3,4} \mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_i) \geq \frac{\epsilon^2}{128k^2l^2q^2}.$$

To see this, note that for every $\lambda \in \mathbb{N}$ at least one of the following must be true

1. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_1) \geq \frac{\epsilon^2}{128k^2l^2q^2}$
2. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2) \geq \frac{\epsilon^2}{128k^2l^2q^2}$
3. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_3) \geq \frac{\epsilon^2}{128}$
4. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_4) \geq \frac{\epsilon^2}{128k^2l^2q^2}$

If this was not the case, then it would hold that

$$\begin{aligned}
&lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_2) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_3) + lq \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_4) \\
&\leq lq\sqrt{8k^2\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_1)} + lq\sqrt{8k^2\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2)} + \sqrt{8\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_3)} + lq\sqrt{8k^2\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_4)} \\
&< \frac{\epsilon}{4} + \frac{\epsilon}{4} + \frac{\epsilon}{4} + \frac{\epsilon}{4} = \epsilon,
\end{aligned}$$

which contradicts inequality (1). Therefore, the advantage of $\mathcal{A}^*$ is at least

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}^*) \geq \max_{i=1,2,3,4} \mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_i) \geq \frac{\epsilon^2}{128k^2l^2q^2},$$

which concludes the proof. □

We will now provide the proof of Theorem 1.

Proof of Theorem 1. Let $\mathcal{A}$ be a KDM-CPA adversary against PKE. Consider the following sequence of hybrid games. For notational convenience we assume that the oracles have access to all local variables of the games (without explicitly specifying so). We will first provide an overview of game 1 - 8 on the next pages.

Game 1

For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
  $\mathbf{s}_i \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y} \leftarrow \mathbf{A}_i \mathbf{s}_i + \mathbf{e}_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$, $sk \leftarrow \mathbf{s}_i$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
  If $b = 0$
    $f \leftarrow \mathbf{0}$
  $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  $\mathbf{C}_1 \leftarrow \mathbf{R} \cdot \mathbf{A}_j$
  $\mathbf{c}_2 \leftarrow \mathbf{R} \cdot \mathbf{y}_j + \mathsf{C.Encode}(f(\{sk_i\}))$
  $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
  Return $\mathfrak{c}$

Game 2

$\mathbf{s} \leftarrow \mathbb{F}_2^n$
For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$, $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}'_i \leftarrow \mathbf{A}_i \mathbf{s} + \mathbf{e}_i$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{s}_i \leftarrow \mathbf{s} + \mathbf{s}'_i$, $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$, $sk \leftarrow \mathbf{s}_i$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
  If $b = 0$
    $f \leftarrow \mathbf{0}$
  Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
    using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
  $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  $\mathbf{C}_1 \leftarrow \mathbf{R} \cdot \mathbf{A}_j$
  $\mathbf{c}_2 \leftarrow \mathbf{R} \cdot \mathbf{y}_j + \mathbf{G} \cdot (\mathbf{T}_f \mathbf{s} + \mathbf{t}_f)$
  $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
  Return $\mathfrak{c}$

Game 3

$\mathbf{s} \leftarrow \mathbb{F}_2^n$
For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$, $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}'_i \leftarrow \mathbf{A}_i \mathbf{s} + \mathbf{e}_i$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{s}_i \leftarrow \mathbf{s} + \mathbf{s}'_i$, $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$, $sk \leftarrow \mathbf{s}_i$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
  If $b = 0$
    $f \leftarrow \mathbf{0}$
  Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
    using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
  $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  $\mathbf{C}_1 \leftarrow \mathbf{R} \cdot \mathbf{A}_j$
  $\mathbf{c}_2 \leftarrow (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
  $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
  Return $\mathfrak{c}$

Game 4

$\mathbf{s} \leftarrow \mathbb{F}_2^n$
For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$, $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}'_i \leftarrow \mathbf{A}_i \mathbf{s} + \mathbf{e}_i$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{s}_i \leftarrow \mathbf{s} + \mathbf{s}'_i$, $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$, $sk \leftarrow \mathbf{s}_i$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
  If $b = 0$
    $f \leftarrow \mathbf{0}$
  Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
    using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
  $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times n}$
  $\mathbf{C}_1 \leftarrow \mathbf{U}$
  $\mathbf{c}_2 \leftarrow (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
  $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
  Return $\mathfrak{c}$

Game 5

$\mathbf{s} \leftarrow \mathbb{F}_2^n$
For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$, $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}'_i \leftarrow \mathbf{A}_i \mathbf{s} + \mathbf{e}_i$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{s}_i \leftarrow \mathbf{s} + \mathbf{s}'_i$, $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$, $sk \leftarrow \mathbf{s}_i$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
If $b = 0$
  $f \leftarrow \mathbf{0}$
Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
  using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
$\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
$\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times n}$
$\mathbf{C}_1 \leftarrow \mathbf{U} - \mathbf{G}\mathbf{T}_f$
$\mathbf{c}_2 \leftarrow \mathbf{U}\mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
$\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
Return $\mathfrak{c}$

Game 6

$\mathbf{s} \leftarrow \mathbb{F}_2^n$
For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$, $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}'_i \leftarrow \mathbf{A}_i \mathbf{s} + \mathbf{e}_i$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{s}_i \leftarrow \mathbf{s} + \mathbf{s}'_i$, $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$, $sk \leftarrow \mathbf{s}_i$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
If $b = 0$
  $f \leftarrow \mathbf{0}$
Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
  using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
$\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
$\mathbf{C}_1 \leftarrow \mathbf{R}\mathbf{A}_j - \mathbf{G}\mathbf{T}_f$
$\mathbf{c}_2 \leftarrow \mathbf{R}\mathbf{A}_j \mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
$\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
Return $\mathfrak{c}$

Game 7

For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
  $\mathbf{y}'_i \leftarrow_\$ \mathbb{F}_2^m$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
If $b = 0$
  $f \leftarrow \mathbf{0}$
Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
  using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
$\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
$\mathbf{C}_1 \leftarrow \mathbf{R}\mathbf{A}_j - \mathbf{G}\mathbf{T}_f$
$\mathbf{c}_2 \leftarrow \mathbf{R}\mathbf{y}'_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
$\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
Return $\mathfrak{c}$

Game 8

For $i = 1, \ldots, l$
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
  $\mathbf{y}'_i \leftarrow_\$ \mathbb{F}_2^m$
  $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
  $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
$\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times n}$
$\mathbf{u} \leftarrow_\$ \mathbb{F}_2^k$
$\mathbf{C}_1 \leftarrow \mathbf{U}$
$\mathbf{c}_2 \leftarrow \mathbf{u}$
$\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
Return $\mathfrak{c}$

- **Game 1** is identical to the KDM-CPA experiment, we only replace $\mathsf{PKE.KeyGen}$ and $\mathsf{PKE.Enc}$ with their instantiations according to PKE.

- In **game 2**, we change the experiment in three ways. First, the public and secret keys are computed from a *master secret* $\mathbf{s}$. More specifically, we first choose $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$ uniformly at random and then compute $\mathbf{y}'_i \leftarrow \mathbf{A}_i\mathbf{s}+\mathbf{e}_i$ for each index i. We obtain the public and secret keys by rerandomizing $\mathbf{s}$ and $\mathbf{y}'_i$ correlated way. Specifically, we set $\mathbf{s}_i = \mathbf{s} + \mathbf{s}'_i$ and $\mathbf{y}_i = \mathbf{y}'_i + \mathbf{A}_i\mathbf{s}'_i$ for a uniformly random and independent $\mathbf{s}'_i$. Since $\mathbf{s}_i$ is uniformly random and independent of $\mathbf{s}$ and further
$$\mathbf{y}_i = \mathbf{y}'_i + \mathbf{A}_i\mathbf{s}'_i = \mathbf{A}_i\mathbf{s}_i + \mathbf{e}_i,$$
we have that the (pk_i, sk_i) are identically distributed to game 1. Thus, this modification to the experiment did not introduce any statistically difference. Secondly, we write $\mathsf{C.Encode}(\cdot)$ using the generator matrix $\mathbf{G}$ of C, which is also merely a syntactical change.

  The third modification consists in changing the way the functions f the encryption oracle is queried with are evaluated. Since each f is restricted to be an affine function, we can write it as
$$f(\{sk_i\}) = \sum_{i=1}^{l} \mathbf{T}_i\mathbf{s}_i + \mathbf{t}.$$
for $\mathbf{T}_1, , \ldots, \mathbf{T}_l \in \mathbb{F}_2^{n\times n}$ and $\mathbf{t} \in \mathbb{F}_2^n$. Using that $\mathbf{s}_i = \mathbf{s} + \mathbf{s}'_i$ we can write
$$\begin{aligned} f(\{sk_i\}) &= \sum_{i=1}^{l} \mathbf{T}_i(\mathbf{s} + \mathbf{s}'_i) + \mathbf{t} \\ &= \left(\sum_{i=1}^{l} \mathbf{T}_i\right)\mathbf{s} + \sum_{i=1}^{l} \mathbf{T}_i\mathbf{s}'_i + \mathbf{t}. \end{aligned}$$
  Therefore, setting $\mathbf{T}_f = \sum_{i=1}^{l} \mathbf{T}_i$ and $\mathbf{t}_f = \sum_{i=1}^{l} \mathbf{T}_i\mathbf{s}'_i + \mathbf{t}$ we can write $f(\{sk_i\}) = \mathbf{T}_f\mathbf{s} + \mathbf{t}_f$. Thus, also the third modification does not introduce any statistical difference.

- In **game 3** we change the way $\mathbf{c}_2$ is computed. However, plugging in $\mathbf{y}_j = \mathbf{A}_j\mathbf{s}_j + \mathbf{e}_j = \mathbf{A}_j(\mathbf{s} + \mathbf{s}'_j) + \mathbf{e}_j$ and rearranging terms yields
$$\begin{aligned} \mathbf{c}_2 &= \mathbf{R} \cdot \mathbf{y}_j + \mathbf{G} \cdot (\mathbf{T}_f\mathbf{s} + \mathbf{t}_f) \\ &= \mathbf{R} \cdot (\mathbf{A}_j(\mathbf{s} + \mathbf{s}'_j) + \mathbf{e}_j) + \mathbf{G} \cdot (\mathbf{T}_f\mathbf{s} + \mathbf{t}_f) \\ &= (\mathbf{RA} + \mathbf{GT}_f)\mathbf{s} + \mathbf{Re}_j + \mathbf{Gt}_f + \mathbf{RAs}'_j \\ &= (\mathbf{C}_1 + \mathbf{GT}_f)\mathbf{s} + \mathbf{Re}_j + \mathbf{Gt}_f + \mathbf{C}_1\mathbf{s}'_j. \end{aligned}$$

- In **game 4**, at every call to $\mathcal{O}_{\mathsf{KDM}}$ the value $\mathbf{C}_1$ is chosen uniformly at random instead of computed by $\mathbf{C}_1 \leftarrow \mathbf{RA}_j$. We can show that game 3 and game 4 are computationally indistinguishable, given that the problem $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$ is hard. The reduction loses a factor if lq.

- In **game 5** we compute $\mathbf{C}_1$ by $\mathbf{C}_1 \leftarrow \mathbf{U} - \mathbf{GT}_f$. Since $\mathbf{U}$ is chosen independently and uniformly at random, **game 4** and **game 5** are identically distributed from the view of $\mathcal{A}$.

- In **game 6** we replace $\mathbf{U}$ again by $\mathbf{RA}_j$. We can show that game 5 and game 6 are computationally indistinguishable provided that the problem $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$ is hard. The reduction loses a factor if lq.

- In **game 7** we choose the $\mathbf{y}'_i$ uniformly at random instead of by $\mathbf{y}'_i \leftarrow \mathbf{A}_i\mathbf{s} + \mathbf{e}_i$. We can show that game 6 and game 7 are computationally indistinguishable, given that $\mathsf{DLPN}(n, l \cdot m, \rho)$ is hard. This reduction is tight.
- In **game 8** the values $\mathbf{C}_1$ and $\mathbf{c}_2$ are chosen uniformly at random. Therefore the output of $\mathcal{O}_{\mathsf{KDM}}$ is independent of the challenge bit b and consequently $\mathcal{A}$'s advantage in game 8 is 0. We can show that game 7 and game 8 are computationally indistinguishable given that $\mathsf{DDLPN}(n, m, k, \rho)$ is hard. This reduction loses a factor of lq.

We will now show that game 3 and game 4 are computationally indistinguishable, given that $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$ is hard. Assume that

$$|\Pr[\mathsf{Game3}(\mathcal{A}) = 1] - \Pr[\mathsf{Game4}(\mathcal{A}) = 1]| = \epsilon_1$$

for some ϵ_1. Assume further that $\mathcal{A}$ makes at most $q = \mathsf{poly}(\lambda)$ queries to its KDM oracle for each $j \in \{1, \ldots, l\}$. We will construct a PPT distinguisher $\mathcal{D}_1$ which distinguishes $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{S}(m, \rho))$ with advantage $\epsilon_1/(q \cdot l)$. The distinguisher $\mathcal{D}_1$ is given as follows.

Distinguisher $\mathcal{D}_1$
- Input $(\mathbf{A}, \mathbf{C}, \mathbf{e}, \mathbf{z})$
- $j^* \leftarrow_\$ \{1, \ldots, l\}$
- $d^* \leftarrow_\$ \{1, \ldots, q\}$
- $d \leftarrow 0$
- $\mathbf{s} \leftarrow \mathbb{F}_2^n$
- For $i = 1, \ldots, l$
  - If $i = j^*$
    - $\mathbf{A}_i \leftarrow \mathbf{A}$
    - $\mathbf{e}_i \leftarrow \mathbf{e}$
  - Otherwise
    - $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
    - $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  - $\mathbf{y}'_i \leftarrow \mathbf{A}_i\mathbf{s} + \mathbf{e}_i$
  - $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  - $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i\mathbf{s}'_i$
  - $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$
- $b \leftarrow_\$ \{0, 1\}$
- $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
- Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
- If $b = 0$
  - $f \leftarrow \mathbf{0}$
- Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f\mathbf{s} + \mathbf{t}_f$,
  - using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
- If $j = j^*$
  - $d \leftarrow d + 1$
- If $j < j^*$ or $(j = j^*$ and $d < d^*)$
  - $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  - $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times m}$
  - $\mathbf{C}_1 \leftarrow \mathbf{U}$
  - $\mathbf{c}_2 \leftarrow (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j$
- Else if $j = j^*$ and $d = d^*$
  - $\mathbf{C}_1 \leftarrow \mathbf{C}$
  - $\mathbf{c}_2 \leftarrow (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s} + \mathbf{z} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j$
- Else
  - $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  - $\mathbf{C}_1 \leftarrow \mathbf{R}\mathbf{A}_j$
  - $\mathbf{c}_2 \leftarrow (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j$
- $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
- Return $\mathfrak{c}$

$\mathcal{D}_1$ does the following. It first chooses two indices $j^* \leftarrow_\$ \{1, \ldots, l\}$ and $d^* \leftarrow_\$ \{1, \ldots, q\}$ uniformly at random. To compute the public key pk_{j^*}, it uses the $\mathbf{A}$ and $\mathbf{e}$ taken from its own input. For all keys with index $j < j^*$, it simulates queries to the KDM oracle like game 4. For all keys with index $j > j^*$ it simulates queries to the KDM oracle like game 3. For the key with index j, it keeps a counter d. As long as $d < d^*$ it simulates queries to the KDM oracle like game 4, if $d > d^*$ they are simulated as in game 3. However, if $j = j^*$ and $d = d^*$ it embeds $\mathbf{C}$ and $\mathbf{z}$ from its own input into the KDM query.

First observe that $\mathbf{A}_{j^*}$ and $\mathbf{e}_j^*$ have the same distribution as in game 3 and game 4, as $\mathbf{A}$ is chosen uniformly from $\mathbb{F}_2^{m\times n}$ and $\mathbf{e}$ is chosen from $\mathsf{Ber}(m,\rho)$. Assume now that $\mathcal{D}_1$'s input is distributed according to $(\mathbf{A},\mathbf{U},\mathbf{e},\mathbf{Re})$. Then it holds for $j=j^*$ and $d=d^*$ that

$$\mathbf{C}_1 = \mathbf{U}$$

and

$$\begin{aligned}\mathbf{c}_2 &= (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s} + \mathbf{R}\mathbf{e} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j \\ &= (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j\end{aligned}$$

Thus, $\mathcal{D}_1$ answers the d^*-th query for the key with index j^* exactly like game 4. On the other hand, if $\mathcal{D}_1$'s input is distributed according to $(\mathbf{A},\mathbf{RA},\mathbf{e},\mathbf{Re})$, then it holds for $j=j^*$ and $d=d^*$ that

$$\mathbf{C}_1 = \mathbf{R}\mathbf{A} = \mathbf{R}\mathbf{A}_j$$

and

$$\begin{aligned}\mathbf{c}_2 &= (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s} + \mathbf{R}\mathbf{e} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j \\ &= (\mathbf{C}_1 + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j.\end{aligned}$$

Consequently, $\mathcal{D}_1$ answers the d^*-th query for the key with index j^* exactly like game 3. A standard calculation for this type of hybrid argument now shows that

$$\mathsf{Adv}(\mathcal{D}_1) \geq \frac{1}{lq}|\Pr[\mathsf{Game3}(\mathcal{A})=1] - \Pr[\mathsf{Game4}(\mathcal{A})=1]| \geq \frac{\epsilon_1}{lq},$$

i.e. $\mathcal{D}_1$ distinguishes $\mathsf{eDLPN}(n,m,k,\rho,\mathsf{Ber}(m,\rho))$ with advantage $\frac{\epsilon_1}{lq}$.

Next we will show that game 5 and game 6 are computationally indistinguishable, given that $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$ is hard. This proof is basically the same as the proof of indistinguishability of game 3 and game 4. Assume that $\mathcal{A}$ distinguishes game 5 and game 6 with advantage ϵ_2, i.e.

$$|\Pr[\mathsf{Game5}(\mathcal{A}) = 1] - \Pr[\mathsf{Game6}(\mathcal{A}) = 1]| = \epsilon_2.$$

Assume further that for each index $j \in \{1, \ldots, l\}$ $\mathcal{A}$ makes at most q queries to the KDM oracle. We will construct a distinguisher $\mathcal{D}_2$ that distinguishes $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$ with advantage $\epsilon_2/(l \cdot q)$. The distinguisher $\mathcal{D}_2$ is given as follows.

Distinguisher $\mathcal{D}_2$
- Input $(\mathbf{A}, \mathbf{C}, \mathbf{e}, \mathbf{z})$
- $j^* \leftarrow_\$ \{1, \ldots, l\}$
- $d^* \leftarrow_\$ \{1, \ldots, q\}$
- $d \leftarrow 0$
- $\mathbf{s} \leftarrow \mathbb{F}_2^n$
- For $i = 1, \ldots, l$
  - If $i = j^*$
    - $\mathbf{A}_i \leftarrow \mathbf{A}$
    - $\mathbf{e}_i \leftarrow \mathbf{e}$
  - Otherwise
    - $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
    - $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  - $\mathbf{y}'_i \leftarrow \mathbf{A}_i\mathbf{s} + \mathbf{e}_i$
  - $\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
  - $\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i\mathbf{s}'_i$
  - $pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$
- $b \leftarrow_\$ \{0, 1\}$
- $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
- Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
- If $b = 0$
  - $f \leftarrow \mathbf{0}$
- Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f\mathbf{s} + \mathbf{t}_f$,
  - using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
- If $j = j^*$
  - $d \leftarrow d + 1$
- If $j < j^*$ or ($j = j^*$ and $d < d^*$)
  - $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  - $\mathbf{C}_1 \leftarrow \mathbf{R}\mathbf{A}_j - \mathbf{G}\mathbf{T}_f$
  - $\mathbf{c}_2 \leftarrow \mathbf{R}\mathbf{A}_j\mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j$
- Else if $j = j^*$ and $d = d^*$
  - $\mathbf{C}_1 \leftarrow \mathbf{C} - \mathbf{G}\mathbf{T}_f$
  - $\mathbf{c}_2 \leftarrow \mathbf{C}\,\mathbf{s} + \mathbf{z} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j$
- Else
  - $\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
  - $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times m}$
  - $\mathbf{C}_1 \leftarrow \mathbf{U} - \mathbf{G}\mathbf{T}_f$
  - $\mathbf{c}_2 \leftarrow \mathbf{U}\mathbf{s} + \mathbf{R}\mathbf{e}_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1\mathbf{s}'_j$
- $\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
- Return $\mathfrak{c}$

Again, notice that all oracle queries with $j < j^*$ or $j = j^*$ and $d < d^*$ are handled by $\mathcal{D}_2$ as in game 6, while all queries with $j > j^*$ or $j = j^*$ and $d > d^*$ are handled as in game 5. Also, the distinguisher's own challenge is embedded at the query with $j = j^*$ and $d = d^*$. If $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{RA}, \mathbf{e}, \mathbf{Re})$, then this query is answered as in game 6. On the other hand, if $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{U}, \mathbf{e}, \mathbf{Re})$, then the query is answered as in game 5. Again, a standard computation shows that

$$\mathsf{Adv}(\mathcal{D}_2) \geq \frac{1}{lq}|\Pr[\mathsf{Game5}(\mathcal{A}) = 1] - \Pr[\mathsf{Game6}(\mathcal{A}) = 1]| \geq \frac{\epsilon_2}{lq},$$

i.e. $\mathcal{D}_2$ distinguishes $\mathsf{eDLPN}(n, m, k, \rho, \mathsf{Ber}(m, \rho))$ with advantage $\frac{\epsilon_2}{lq}$.

We will now turn to the indistinguishability of game 6 and game 7. Assume again that $\mathcal{A}$ distinguishes between game 6 and game 7 with advantage ϵ_3, i.e.

$$|\Pr[\mathsf{Game6}(\mathcal{A}) = 1] - \Pr[\mathsf{Game7}(\mathcal{A}) = 1]| = \epsilon_3,$$

We will construct a distinguisher $\mathcal{D}_3$ which distinguishes $\mathsf{DLPN}(n, lm, \rho)$ with advantage ϵ_3. For notational convenience, we assume that $\mathcal{A}$ has access to an LPN sample oracle $\mathcal{O}_{\mathsf{LPN}}$ which can be queried l times and outputs chunks of m samples at each query, i.e. at each query $\mathcal{O}_{\mathsf{LPN}}$ outputs a matrix $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ and a vector $\mathbf{y}' \in \mathbb{F}_2^m$. $\mathcal{D}_3$ is given as follows.

Distinguisher $\mathcal{D}_3$
Has access to LPN sample oracle $\mathcal{O}_{\mathsf{LPN}}$
For $i = 1, \ldots, l$
$(\mathbf{A}_i, \mathbf{y}'_i) \leftarrow \mathcal{O}_{\mathsf{LPN}}()$
$\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
$pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
If $b = 0$
$f \leftarrow \mathbf{0}$
Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
$\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
$\mathbf{C}_1 \leftarrow \mathbf{R}\mathbf{A}_j - \mathbf{G}\mathbf{T}_f$
$\mathbf{c}_2 \leftarrow \mathbf{R}\mathbf{y}'_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
$\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
Return $\mathfrak{c}$

The construction of $\mathcal{D}_3$ is actually quite simple. Instead of generating the $\mathbf{A}_i$ and $\mathbf{y}'_i$ by itself, it queries the LPN sample oracle for these values. Now, it is easy to see that if $\mathcal{D}_3$ is connected to an oracle that outputs samples of the form $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, then the view of $\mathcal{A}$ in $\mathcal{D}_3$'s simulation is the same as in game 6. On the other hand, if the LPN sample oracle outputs samples of the form $(\mathbf{A}, \mathbf{u}')$, then $\mathcal{A}$'s view in $\mathcal{D}_3$'s simulation is the same as in game 7. Consequently,

$$\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_3) = |\Pr[\mathsf{Game6}(\mathcal{A}) = 1] - \Pr[\mathsf{Game7}(\mathcal{A}) = 1]| \geq \epsilon_3,$$

which contradicts the hardness of $\mathsf{DLPN}(n, lm, \rho)$.

Finally, we will turn to the indistinguishability of game 7 and game 8. Assume that $\mathcal{A}$ distinguishes between game 7 and game 8 with non-negligible advantage ϵ_4, i.e.

$$|\Pr[\mathsf{Game7}(\mathcal{A}) = 1] - \Pr[\mathsf{Game8}(\mathcal{A}) = 1]| = \epsilon_4.$$

Assume once more that for each $j \in \{1, \ldots, l\}$ the adversary $\mathcal{A}$ makes at most q queries to the KDM oracle with index j. We will now construct a distinguisher $\mathcal{D}_4$ that distinguishes $\mathsf{DDLPN}(n, m, k, \rho)$ with advantage $\epsilon_4/(q \cdot l)$.

$\mathcal{D}_4$ does the following. Let $(\mathbf{A}, \mathbf{C}, \mathbf{a}, \mathbf{c})$ be $\mathcal{D}_4$'s input. It chooses challenge indices j^* and d^* uniformly at random. For $i = j^*$, it sets $\mathbf{A}_i = \mathbf{A}$ and $\mathbf{y}'_i = \mathbf{a}$. Observe that both $\mathbf{A}_i$ and $\mathbf{y}'_i$ have the same distribution as in both game 7 and game 8. For $j < j^*$ or $j = j^*$ and $d < d^*$ $\mathcal{D}_4$ simulates queries to the KDM oracle as in game 8 and for $j > j^*$ or $j = j^*$ and $d > d^*$ as in game 7. Its own challenge is again embedded at $j = j^*$ and $d = d^*$. Assume that $\mathcal{D}_4$'s input $(\mathbf{A}, \mathbf{C}, \mathbf{a}, \mathbf{c})$ is of the form $(\mathbf{A}, \mathbf{R}\mathbf{A}, \mathbf{a}, \mathbf{R}\mathbf{a})$. Then it holds that

$$\mathbf{C}_1 = \mathbf{C} - \mathbf{G}\mathbf{T}_f = \mathbf{R}\mathbf{A} - \mathbf{G}\mathbf{T}_f = \mathbf{R}\mathbf{A}_j - \mathbf{G}\mathbf{T}_f$$

and

$$\begin{aligned}\mathbf{c}_2 &= \mathbf{c} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j \\ &= \mathbf{R}\mathbf{a} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j \\ &= \mathbf{R}\mathbf{y}'_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j.\end{aligned}$$

Distinguisher $\mathcal{D}_4$
Input $(\mathbf{A}, \mathbf{C}, \mathbf{a}, \mathbf{c})$
$j^* \leftarrow_\$ \{1, \ldots, l\}$
$d^* \leftarrow_\$ \{1, \ldots, q\}$
$d \leftarrow 0$
$\mathbf{s} \leftarrow \mathbb{F}_2^n$
For $i = 1, \ldots, l$
If $i = j^*$
$\mathbf{A}_i \leftarrow \mathbf{A}$
$\mathbf{y}'_i \leftarrow \mathbf{a}$
Otherwise
$\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$\mathbf{y}'_i \leftarrow_\$ \mathbb{F}_2^m$
$\mathbf{s}'_i \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{y}_i \leftarrow \mathbf{y}'_i + \mathbf{A}_i \mathbf{s}'_i$
$pk_i \leftarrow (\mathbf{A}_i, \mathbf{y}_i)$
$b \leftarrow_\$ \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KDM}}(\cdot,\cdot)}(\{pk_i\}, 1^\lambda)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{KDM}}(f, j)$
If $b = 0$
$f \leftarrow \mathbf{0}$
Compute $\mathbf{T}_f$, $\mathbf{t}_f$ s.t. $f(\{sk_i\}) = \mathbf{T}_f \mathbf{s} + \mathbf{t}_f$,
using that $sk_i = \mathbf{s} + \mathbf{s}'_i$
If $j = j^*$
$d \leftarrow d + 1$
If $j < j^*$ or $(j = j^*$ and $d < d^*)$
$\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times m}$
$\mathbf{u} \leftarrow_\$ \mathbb{F}_2^k$
$\mathbf{C}_1 \leftarrow \mathbf{U}$
$\mathbf{c}_2 \leftarrow \mathbf{u}$
Else if $j = j^*$ and $d = d^*$
$\mathbf{C}_1 \leftarrow \mathbf{C} - \mathbf{G}\mathbf{T}_f$
$\mathbf{c}_2 \leftarrow \mathbf{c} + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
Else
$\mathbf{R} \leftarrow_\$ \mathsf{Ber}(k \times m, \rho)$
$\mathbf{C}_1 \leftarrow \mathbf{R}\mathbf{A}_j - \mathbf{G}\mathbf{T}_f$
$\mathbf{c}_2 \leftarrow \mathbf{R}\mathbf{y}'_j + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j$
$\mathfrak{c} \leftarrow (\mathbf{C}_1, \mathbf{c}_2)$
Return $\mathfrak{c}$

Thus, $\mathbf{C}_1$ and $\mathbf{c}_2$ are as in game 7. On the other hand, if $\mathcal{D}_4$'s input $(\mathbf{A}, \mathbf{C}, \mathbf{a}, \mathbf{c})$ is of the form $(\mathbf{A}, \mathbf{U}, \mathbf{a}, \mathbf{u}')$, then $\mathbf{C} = \mathbf{U}'$ and $\mathbf{c} = \mathbf{u}'$ for uniformly random $\mathbf{U}' \leftarrow_\$ \mathbb{F}_2^{k \times n}$ and $\mathbf{u}' \leftarrow_\$ \mathbb{F}_2^k$. Thus it holds that

$$\mathbf{C}_1 = \mathbf{U}' - \mathbf{G}\mathbf{T}_f = \mathbf{U}$$

and

$$\mathbf{c}_2 = \mathbf{u}' + \mathbf{G}\mathbf{t}_f + \mathbf{C}_1 \mathbf{s}'_j = \mathbf{u}$$

as $\mathbf{U}'$ and $\mathbf{u}'$ are uniformly random. Consequently, $\mathbf{C}_1$ and $\mathbf{c}_2$ are as in game 6. Again a simple hybrid argument shows that

$$\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_4) = \frac{1}{lq} |\Pr[\mathsf{Game7}(\mathcal{A}) = 1] - \Pr[\mathsf{Game8}(\mathcal{A}) = 1]| \geq \frac{\epsilon_4}{lq},$$

which contradicts the hardness of $\mathsf{DDLPN}(n, m, k, \rho)$.

We will now turn to assembling the quantitative statement of the theorem. By the triangle inequality it holds that

$$\begin{aligned} \mathsf{Adv}_{\mathsf{KDM\text{-}CPA}}(\mathcal{A}) &\leq \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 \\ &\leq lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + lq \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_2) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_3) + lq \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_4). \end{aligned}$$

This concludes the proof. □

5 LPN Sample Amplification

In this Section, we will show that the hardness of LPN with a bounded number of samples implies the hardness of LPN with an unbounded number of samples, if one is willing to accept an increase in the amount of noise. Recall that $\mathsf{S}(2n, \rho)$ is the uniform distribution on vectors of weight $\lfloor \rho 2n \rfloor$ in $\mathbb{F}_2^n$.

Theorem 2. *Let λ be a security parameter, $n = \mathsf{poly}(\lambda)$ be a positive integer and $\rho = \rho(\lambda) \in [0, 1/2]$. Let*

$$\rho' \geq \frac{1}{2}(1 - (1 - 2\rho)^{\lfloor \rho 2n \rfloor}).$$

If $\mathsf{eDLPN}(n, 2n, 1, \rho, \mathsf{S}(2n, \rho))$, $\mathsf{DLPN}(n, 2n, \mathsf{S}(2n, \rho))$ and $\mathsf{DDLPN}(n, 2n, 1, \rho)$ are hard, then it holds that $\mathsf{DLPN}(n, \rho')$ is also hard. Precisely, if $\mathcal{D}$ is a PPT distinguisher against $\mathsf{DLPN}(n, \rho')$ that makes at most q queries to its LPN oracle, then there exist PPT distinguishers $\mathcal{D}_1$ against the problem $\mathsf{eDLPN}(n, 2n, 1, \rho, \mathsf{S}(2n, \rho))$, $\mathcal{D}_2$ against $\mathsf{DLPN}(n, 2n, \mathsf{S}(2n, \rho))$ and $\mathcal{D}_3$ against $\mathsf{DDLPN}(n, 2n, 1, \rho)$ such that

$$\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}) \leq q \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2) + q \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3).$$

Corollary 9. *Let n, ρ and ρ' be as in Theorem 2. If $\mathsf{LPN}(n, 2n, \rho)$ is hard, then $\mathsf{DLPN}(n, \rho')$ is also hard. More precisely, if $\mathcal{D}$ is a PPT distinguisher which distinguishes $\mathsf{DLPN}(n, \rho')$ with advantage ϵ and makes at most q queries to its LPN oracle, then there exists a PPT adversary $\mathcal{A}^*$ which breaks $\mathsf{LPN}(n, 2n, \rho)$ with advantage*

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}^*) \geq \frac{\epsilon^2}{72q^2}.$$

Proof of Corollary 9. Let $\epsilon = \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D})$. By theorem 2 there exist PPT distinguishers $\mathcal{D}_1$ against $\mathsf{eDLPN}(n, 2n, 1, \rho, \mathsf{S}(2n, \rho))$, $\mathcal{D}_2$ against $\mathsf{DLPN}(n, 2n, \mathsf{S}(2n, \rho))$ and $\mathcal{D}_3$ against $\mathsf{DDLPN}(n, 2n, 1, \rho)$ such that

$$\epsilon \leq q \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2) + q \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3). \tag{2}$$

By Lemma 6 there exists a PPT adversary $\mathcal{A}_1$ that solves $\mathsf{LPN}(n, 2n, \rho)$ with advantage

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_1) \geq \frac{\mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1)^2}{8}.$$

By Corollary 2 there exists a PPT adversary $\mathcal{A}_2$ that solves $\mathsf{LPN}(n, 2n, \rho)$ with advantage

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2) \geq \frac{(1-\gamma)\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2)^2}{8 \cdot \sqrt{4\pi n \rho(1-\rho)}}$$

for some $\gamma = o(1)$. Finally, by Lemma 3 there exists a PPT adversary $\mathcal{A}_3$ that solves $\mathsf{LPN}(n, 2n, \rho)$ with advantage

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_3) \geq \frac{\mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3)^2}{8}.$$

We now define the adversary $\mathcal{A}^*$. On input $(\mathbf{A}, \mathbf{y})$ $\mathcal{A}^*$ runs the algorithms $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ on input $(\mathbf{A}, \mathbf{y})$ in parallel, checks the results of all of them (i.e. checks for each candidate solution $\mathbf{s}$ whether $\|\mathbf{y} - \mathbf{A}\mathbf{s}\| \approx \rho 2n$) and outputs a solution if one has been found. Clearly, it holds that

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}^*) \geq \max_{i=1,2,3} \mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_i),$$

as $\mathcal{A}^*$'s advantage in solving $\mathsf{LPN}(n, 2n, \rho)$ is at least as big as the advantage of the best $\mathcal{A}_i$ (for a given λ). We claim that

$$\max_{i=1,2,3} \mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_i) \geq \frac{\epsilon^2}{72 \max\{q^2, \sqrt{4\pi n\rho(1-\rho)}/(1-\gamma)\}}.$$

To see this, note that for every $\lambda \in \mathbb{N}$ at least one of the following must be true

1. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_1) \geq \frac{\epsilon^2}{72q^2}$
2. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2) \geq \frac{\epsilon^2}{72\sqrt{4\pi n\rho(1-\rho)}/(1-\gamma)}$
3. $\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_3) \geq \frac{\epsilon^2}{72q^2}$

If this was not the case, then it would hold that

$$\begin{aligned}
&q \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2) + q \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3) \\
&\leq q\sqrt{8\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_1)} + \sqrt{\frac{8\sqrt{4\pi n\rho(1-\rho)}}{1-\gamma}\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_2)} + q\sqrt{8\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_3)} \\
&< \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} = \epsilon,
\end{aligned}$$

which contradicts inequality (2). Therefore, the advantage of $\mathcal{A}^*$ is at least

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}^*) \geq \max_{i=1,2,3} \mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}_i) \geq \frac{\epsilon^2}{72 \max\{q^2, \sqrt{4\pi n\rho(1-\rho)}/(1-\gamma)\}},$$

As $\sqrt{4\pi n\rho(1-\rho)}/(1-\gamma)$ is upper bounded by $2\sqrt{\pi n}(1+o(1))$ and the solution of $\mathsf{LPN}(n, \rho')$ is not even information-theoretically defined for less then n queries, we can safely assume that $q^2 \geq \sqrt{4\pi n\rho(1-\rho)}/(1-\gamma)$ and therefore

$$\mathsf{Adv}_{\mathsf{LPN}}(\mathcal{A}^*) \geq \frac{\epsilon^2}{72q^2},$$

which concludes the proof. □

Corollary 9 can be seen as a trade-off between noise and extra samples. We tolerate that the amount of noise required gets squared, while in turn we get an arbitrary polynomial amount of samples.

Proof of Theorem 2. We will prove the theorem for the minimal ρ', i.e. $\rho' = \frac{1}{2}(1-(1-2\rho)^{\lfloor \rho 2n \rfloor})$. Let $\mathcal{D}$ be a PPT distinguisher against $\mathsf{DLPN}(n, \rho')$. We will provide a series of hybrid experiments $\mathsf{Exp}_1, \mathsf{Exp}_2, \mathsf{Exp}_3, \mathsf{Exp}_4$ and show that from the view of $\mathcal{D}$ any two of experiments the are indistinguishable. We will provide the experiments by defining the sample oracles $\mathcal{O}$ the distinguisher $\mathcal{D}$ gets access to.

Clearly, experiment Exp_1 provides samples from the LPN distribution while experiment Exp_4 provides uniformly random samples. Thus, we need to establish that from the view of $\mathcal{D}$ the experiments Exp_1 and Exp_4 are indistinguishable. We will start with the indistinguishability of Exp_1 and Exp_2. Assume that $\mathcal{D}$ distinguishes with advantage ϵ_1 between Exp_1 and Exp_2, i.e.

$$|\Pr[\mathsf{Exp}_1(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_2(\mathcal{D}) = 1]| = \epsilon_1.$$

Experiment Exp_1	**Experiment** Exp_2	**Experiment** Exp_3	**Experiment** Exp_4
Initialization:	Initialization:	Initialization:	Initialization:
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$	$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$	$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$	-
Oracle $\mathcal{O}_{\mathsf{Exp}_1}()$	$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$	$\mathbf{r} \leftarrow_\$ \mathbb{F}_2^{2n}$	Oracle $\mathcal{O}_{\mathsf{Exp}_4}()$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$	$\mathbf{z} \leftarrow_\$ \mathsf{S}(2n, \rho)$	Oracle $\mathcal{O}_{\mathsf{Exp}_3}()$	$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$e \leftarrow_\$ \mathsf{Ber}(\rho')$	$\mathbf{r} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$	$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$	$y \leftarrow_\$ \mathbb{F}_2$
$y \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e$	Oracle $\mathcal{O}_{\mathsf{Exp}_2}()$	$\mathbf{a} \leftarrow \mathbf{e}^\top \mathbf{A}$	Return $(\mathbf{a}, y)$
Return $(\mathbf{a}, y)$	$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$	$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$	
	$\mathbf{a} \leftarrow \mathbf{e}^\top \mathbf{A}$	Return $(\mathbf{a}, y)$	
	$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$		
	Return $(\mathbf{a}, y)$		

Assume further that $q = \mathsf{poly}(\lambda)$ is an upper bound on the number of samples $\mathcal{D}$ queries. We will construct a PPT distinguisher $\mathcal{D}_1$ that distinguishes the problem $\mathsf{eDLPN}(n, 2n, 1, \rho, \mathsf{S}(2n, \rho))$ with advantage $\geq \epsilon_1/q$. The distinguisher $\mathcal{D}_1$ is given on the left side of Figure 2.

Notice that $\mathcal{D}_1$ answers the first $i^* - 1$ oracle queries of $\mathcal{D}$ exactly like Exp_1, while it answers the last $q - i^*$ queries like Exp_2. In the i^*-th query however, $\mathcal{D}_1$ embeds its own challenge. Moreover, notice that $\mathcal{D}_1$ is efficient as $\mathcal{D}$ is efficient. To analyze the distinguishing advantage of $\mathcal{D}_1$, we will define a sequence of hybrid experiments $\mathsf{H}_0, \ldots, \mathsf{H}_q$. H_i is crafted to answer the first i queries like Exp_1, while it answers the last $q - i$ queries like Exp_2. Experiment H_i is given on the right side of Figure 2.

Distinguisher $\mathcal{D}_1$	Oracle $\mathcal{O}_{\mathcal{D}_1}()$	**Experiment** H_i	Oracle $\mathcal{O}_{\mathsf{H}_i}()$
Input: $(\mathbf{A}, \mathbf{c}, \mathbf{z}, t)$	If $cnt < i^*$	Initialization:	If $cnt \leq i$
$i^* \leftarrow_\$ \{1, \ldots, q\}$	$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$	$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$	$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$	$e \leftarrow_\$ \mathsf{Ber}(\rho')$	$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$	$e \leftarrow_\$ \mathsf{Ber}(\rho')$
$\mathbf{r} = \mathbf{A}\mathbf{s} + \mathbf{z}$	$y \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e$	$\mathbf{z} \leftarrow_\$ \mathsf{S}_{2n}(\lfloor \rho 2n \rfloor)$	$y \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e$
$cnt = 1$	If $cnt = i^*$	$\mathbf{r} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$	If $cnt > i$
$b \leftarrow \mathcal{D}^{\mathcal{O}_{\mathcal{D}_1}()}$	$\mathbf{a} \leftarrow \mathbf{c}^\top$	$cnt \leftarrow 1$	$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$
return b	$y \leftarrow \langle \mathbf{c}, \mathbf{s} \rangle + t$		$\mathbf{a} \leftarrow (\mathbf{e}^\top \mathbf{A})^\top$
	If $cnt > i^*$		$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
	$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$		$cnt \leftarrow cnt + 1$
	$\mathbf{a} \leftarrow (\mathbf{e}^\top \mathbf{A})^\top$		Return $(\mathbf{a}, y)$
	$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$		
	$cnt \leftarrow cnt + 1$		
	Return $(\mathbf{a}, y)$		

Figure 2: The distinguisher $\mathcal{D}_1$ and the hybrid experiments H_i

We are now ready to analyze the distinguishing advantage of $\mathcal{D}_1$. First assume that $\mathcal{D}_1$'s input is of the form $(\mathbf{A}, \mathbf{e}^\top \mathbf{A}, \mathbf{z}, \mathbf{e}^\top \mathbf{z} = \langle \mathbf{z}, \mathbf{e} \rangle)$. Observe that since $\mathbf{z}$ has weight $\lfloor \rho m \rfloor$ and $\mathbf{e}$ is distributed according to $\mathsf{Ber}(m, \rho)$ it holds that $\langle \mathbf{z}, \mathbf{e} \rangle$ is distributed according to $\mathsf{Ber}(\rho')$. Fix the random choice $i^* = i$. Then the sample oracle $\mathcal{O}_{\mathcal{D}_1}()$ implemented by $\mathcal{D}_1$ behaves identical to the sample oracle of H_{i-1}. Consequently, it holds that

$$\Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{e}^\top \mathbf{A}, \mathbf{z}, \mathbf{e}^\top \mathbf{z}) = 1 | i^* = i] = \Pr[\mathsf{H}_{i-1}(\mathcal{D}) = 1]$$

and thus, as i^* is uniformly chosen from $\{1, \ldots, q\}$

$$\begin{aligned}\Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{e}^\top\mathbf{A}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1] &= \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{e}^\top\mathbf{A}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1 | i^* = i] \\ &= \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathsf{H}_{i-1}(\mathcal{D}) = 1].\end{aligned}$$

Next assume that $\mathcal{D}_1$'s input is of the form $(\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{e}^\top\mathbf{z})$. Again, fix the random choice of i^* to $i^* = i$. Then the sample oracle $\mathcal{O}_{\mathcal{D}_1}()$ implemented by $\mathcal{D}_1$ behaves identical to the sample oracle of H_i, as $\langle \mathbf{z}, \mathbf{e} \rangle$ is distributed according to $\mathsf{Ber}(\rho')$. Consequently,

$$\Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1 | i^* = i] = \Pr[\mathsf{H}_i(\mathcal{D}) = 1]$$

and thus

$$\begin{aligned}\Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1] &= \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1 | i^* = i] \\ &= \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathsf{H}_i(\mathcal{D}) = 1].\end{aligned}$$

Together, this yields

$$\begin{aligned}\mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) &= |\Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{e}^\top\mathbf{A}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1] - \Pr[\mathcal{D}_1(\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{e}^\top\mathbf{z}) = 1]| \\ &= \left| \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathsf{H}_{i-1}(\mathcal{D}) = 1] - \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathsf{H}_i(\mathcal{D}) = 1] \right| \\ &= \frac{1}{q} |\Pr[\mathsf{H}_0(\mathcal{D}) = 1] - \Pr[\mathsf{H}_k(\mathcal{D}) = 1]| \\ &= \frac{1}{q} |\Pr[\mathsf{Exp}_2(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_1(\mathcal{D}) = 1]| \\ &\geq \epsilon_1/q.\end{aligned}$$

Thus, $\mathcal{D}_1$ distinguishes $\mathsf{eDLPN}(n, 2n, 1, \rho, \mathsf{S}(2n, \rho))$ with advantage ϵ_1/q.

Next, we turn to the indistinguishability of Exp_2 and Exp_3. Assume towards contradiction that $\mathcal{D}$ distinguishes between Exp_2 and Exp_3 with advantage ϵ_2, i.e.

$$|\Pr[\mathsf{Exp}_2(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_3(\mathcal{D}) = 1]| = \epsilon_2.$$

We will construct a PPT distinguisher $\mathcal{D}_2$ against $\mathsf{DLPN}(n, 2n, \mathsf{S}(2n, \rho)$. $\mathcal{D}_2$ is given as follows.

Distinguisher $\mathcal{D}_2$	Sample Oracle $\mathcal{O}_{\mathcal{D}_2}()$
Input: $(\mathbf{A}, \mathbf{r})$	$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$
$b \leftarrow \mathcal{D}^{\mathcal{O}_{\mathcal{D}_2}()}$	$\mathbf{a} \leftarrow \mathbf{e}^\top\mathbf{A}$
return b	$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
	Return $(\mathbf{a}, y)$

The distinguisher $\mathcal{D}_2$ is efficient, as $\mathcal{D}$ is efficient. First, assume that $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{As} + \mathbf{z})$, where $\mathbf{s}$ is chosen uniformly from $\mathbb{F}_2^n$ and $\mathbf{z}$ is chosen from $\mathsf{S}(2n, \rho)$. Then clearly the

sample oracle $\mathcal{O}_{\mathcal{D}_2}$ behaves just as in Exp_2. On the other hand, if $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{u})$ with $\mathbf{u}$ chosen uniformly random from $\mathbb{F}_2^{2n}$, then the sample $\mathcal{O}_{\mathcal{D}_2}$ simulated by $\mathcal{D}_2$ behaves like the sample oracle in Exp_3. Consequently, it holds that

$$\begin{aligned}\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2) &= |\Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{z}) = 1] - \Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{u}) = 1]| \\ &= |\Pr[\mathsf{Exp}_2(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_3(\mathcal{D}) = 1]| \\ &= \epsilon_2.\end{aligned}$$

Thus, the distinguishing advantage of $\mathcal{D}_2$ against $\mathsf{DLPN}(n, 2n, \mathsf{S}(2n, \rho))$ is ϵ_2.

We will finally turn to showing that from the view of $\mathcal{D}$, Exp_3 and Exp_4 are indistinguishable. Assume towards contradiction that $\mathcal{D}$ distinguishes between Exp_3 and Exp_4 with advantage ϵ_3, i.e.

$$|\Pr[\mathsf{Exp}_3(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_4(\mathcal{D}) = 1]| = \epsilon_3.$$

Assume further $\mathcal{D}$ makes at most $q = \mathsf{poly}(\lambda)$ queries to its sample oracle. We will construct a PPT distinguisher $\mathcal{D}_3$ that distinguishes $\mathsf{DDLPN}(n, 2n, 1, \rho)$ with advantage ϵ_3/q. The distinguisher $\mathcal{D}_3$ is given on the left side of Figure 3.

Distinguisher $\mathcal{D}_3$
Input: $(\mathbf{A}, \mathbf{a}^*, \mathbf{r}, y^*)$
$i^* \leftarrow_\$ \{1, \ldots, q\}$
$cnt = 1$
$b \leftarrow \mathcal{D}^{\mathcal{O}_{\mathcal{D}_3}()}$
return b

Oracle $\mathcal{O}_{\mathcal{D}_3}()$
If $cnt < i^*$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$
$\mathbf{a} \leftarrow (\mathbf{e}^\top \mathbf{A})^\top$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
If $cnt = i^*$
$\mathbf{a} \leftarrow \mathbf{a}^{*\top}$
$y \leftarrow y^*$
If $cnt > i^*$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$y \leftarrow_\$ \mathbb{F}_2$
$cnt \leftarrow cnt + 1$
Return $(\mathbf{a}, y)$

Experiment H'_i
Initialization:
$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$
$\mathbf{r} \leftarrow_\$ \mathbb{F}_2^{2n}$
$cnt \leftarrow 1$

Oracle $\mathcal{O}_{\mathsf{H}'_i}()$
If $cnt \leq i$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(2n, \rho)$
$\mathbf{a} \leftarrow (\mathbf{e}^\top \mathbf{A})^\top$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
If $cnt > i$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$y \leftarrow_\$ \mathbb{F}_2$
$cnt \leftarrow cnt + 1$
Return $(\mathbf{a}, y)$

Figure 3: The distinguisher $\mathcal{D}_3$ and the hybrid experiments H'_i

It is clear that $\mathcal{D}_3$ is efficient, once $\mathcal{D}$ is efficient. Again, to analyze the distinguishing advantage of $\mathcal{D}_3$, we will define a sequence of hybrid experiments $\mathsf{H}'_0, \ldots, \mathsf{H}'_q$. H'_i is crafted to answer the first i queries like Exp_3, while it answers the last $q - i$ queries like Exp_4. Hybrid H'_i is given on the right side of Figure 3. First assume that $\mathcal{D}_3$'s input is of the form $(\mathbf{A}, \mathbf{e}^\top \mathbf{A}, \mathbf{r}, \mathbf{e}^\top \mathbf{r})$. Then it holds that

$$\begin{aligned}\mathbf{a}^* &= (\mathbf{e}^\top \mathbf{A})^\top \\ y^* &= \mathbf{e}^\top \mathbf{r} = \langle \mathbf{e}, \mathbf{r} \rangle\end{aligned}$$

Now fix a random choice $i^* = i$. Then $\mathcal{O}_{\mathcal{D}_3}()$ in $\mathcal{D}_3$'s simulation behaves identically to the sample oracle in H'_i. Thus it holds that

$$\Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{e}^\top \mathbf{A}, \mathbf{r}, \mathbf{e}^\top \mathbf{r}) = 1 | i^* = i] = \Pr[\mathsf{H}'_i(\mathcal{D}) = 1],$$

and consequently

$$\begin{aligned}\Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{e}^\top\mathbf{A}, \mathbf{r}, \mathbf{e}^\top\mathbf{r}) = 1] &= \sum_{i=1}^{q} \frac{1}{q} \Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{e}^\top\mathbf{A}, \mathbf{r}, \mathbf{e}^\top\mathbf{r}) = 1 | i^* = i] \\ &= \sum_{i=1}^{q} \frac{1}{q} \Pr[\mathsf{H}'_i(\mathcal{D}) = 1].\end{aligned}$$

Now suppose that $\mathcal{D}_3$'s input is of the form $(\mathbf{A}, \mathbf{u}, \mathbf{r}, u')$, where $\mathbf{u}^\top \leftarrow_\$ \mathbb{F}_2^n$ and $u' \leftarrow_\$ \mathbb{F}_2$ are chosen uniformly at random. Then it holds that $\mathbf{a}^* = \mathbf{u}$ and $y^* = u'$. Again, fix a random choice $i^* = i$. Then $\mathcal{O}_{\mathcal{D}_3}()$ in $\mathcal{D}_3$'s simulation behaves identically to the sample oracle in H'_{i-1}. Thus it holds that

$$\Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{u}, \mathbf{r}, u') = 1 | i^* = i] = \Pr[\mathsf{H}'_{i-1}(\mathcal{D}) = 1],$$

and consequently

$$\begin{aligned}\Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{u}, \mathbf{r}, u') = 1] &= \sum_{i=1}^{q} \frac{1}{q} \Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{u}, \mathbf{r}, u') = 1 | i^* = i] \\ &= \sum_{i=1}^{q} \frac{1}{q} \Pr[\mathsf{H}'_{i-1}(\mathcal{D}) = 1].\end{aligned}$$

Putting all together, we get

$$\begin{aligned}\mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3) &= |\Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{e}^\top\mathbf{A}, \mathbf{r}, \mathbf{e}^\top\mathbf{r}) = 1] - \Pr[\mathcal{D}_3(\mathbf{A}, \mathbf{u}, \mathbf{r}, u') = 1]| \\ &= \left| \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathsf{H}'_i(\mathcal{D}) = 1] - \sum_{i=1}^{q} \frac{1}{q} \cdot \Pr[\mathsf{H}'_{i-1}(\mathcal{D}) = 1] \right| \\ &= \frac{1}{q} |\Pr[\mathsf{H}'_k(\mathcal{D}) = 1] - \Pr[\mathsf{H}'_0(\mathcal{D}) = 1]| \\ &= \frac{1}{q} |\Pr[\mathsf{Exp}_3(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_4(\mathcal{D}) = 1]| \\ &\geq \epsilon_3/q.\end{aligned}$$

Thus, $\mathcal{D}_3$ distinguishes $\mathsf{DDLPN}(n, 2n, 1, \rho)$ with advantage ϵ_3/q.

We will now turn to the quantitative statement of the theorem. By the triangle inequality it holds that

$$\begin{aligned}\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}) &\leq \epsilon_1 + \epsilon_2 + \epsilon_3 \\ &\leq q \cdot \mathsf{Adv}_{\mathsf{eDLPN}}(\mathcal{D}_1) + \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2) + q \cdot \mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3).\end{aligned}$$

This concludes the proof. □

6 Conclusion

In this work we have constructed the first public key encryption scheme with KDM-CPA security for affine function from the low-noise LPN assumption. Moreover, we have provided a novel connection between LPN with a bounded number of samples and LPN with an unbounded number of samples. Both results have analogues inn the LWE realm (The KDM-CPA secure scheme of Applebaum et

al. [10] and the LWE random self-reduction of Gentry et al. [26]). Both our results follow the same blueprint as their LWE counterparts, however, while in the LWE realm powerful statistical tools such as gaussian regularity [26] and the leftover-hash lemma [31, 19] are available, no comparable statistical techniques are available in the LPN realm. Instead, our approach, following Kiltz et al. [33] was to substitute these techniques with computational counterparts based on LPN. Specifically, the leaky LPN problem, and as a special case the extended LPN problem turned out to be very useful in filling this gap. A natural future direction of work would be to try to lift further results from the LWE/SIS realm into the LPN realm, such as identity based encryption [17, 3, 4, 7] or efficient and compact signature schemes [14, 22, 23].

7 Acknowledgement

I would like to thank the anonymous reviewers of PKC 2015 for their useful feedback. I would further like to thank Daniel Masny for explaining to me the subtleties in the construction of [33] involving the eDLPN problem. Finally, I would like to thank Chris Peikert for pointing me to LWE random self-reductions some time ago. The LPN sample amplification part of this work appeared in my PhD thesis [20], but has not been published in any other peer reviewed publication.

References

[1] T. Acar, M. Belenkiy, M. Bellare, and D. Cash. Cryptographic agility and its relation to circular encryption. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 403–422, 2010.

[2] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005, Proceedings*, pages 374–396, 2005.

[3] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 553–572, 2010.

[4] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 98–115, 2010.

[5] D. Aharonov and O. Regev. A lattice problem in quantum NP. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 210–219, 2003.

[6] M. Alekhnovich. More on average case vs approximation complexity. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 298–307, 2003.

[7] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, pages 334–352, 2012.

[8] B. Applebaum. Key-dependent message security: Generic amplification and completeness. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 527–546, 2011.

[9] B. Applebaum. Garbling XOR gates "for free" in the standard model. In *TCC*, pages 162–181, 2013.

[10] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[11] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography with constant input locality. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 92–110, 2007.

[12] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, pages 62–75, 2002.

[13] A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 278–291, 1993.

[14] F. Böhl, D. Hofheinz, T. Jager, J. Koch, J. H. Seo, and C. Striecks. Practical signatures from standard assumptions. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 461–485, 2013.

[15] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 108–125, 2008.

[16] D. Cash, M. Green, and S. Hohenberger. New definitions and separations for circular security. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, pages 540–557, 2012.

[17] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 523–552, 2010.

[18] B. David, R. Dowsley, and A. C. A. Nascimento. Universally composable oblivious transfer based on a variant of LPN. In *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, pages 143–158, 2014.

[19] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 523–540, 2004.

[20] N. Döttling. *Cryptography based on the Hardness of Decoding.* PhD thesis, Karlsruhe Institute of Technology, May 2014. Available online at http://nbn-resolving.org/urn:nbn:de:swb:90-411105.

[21] N. Döttling, J. Müller-Quade, and A. C. A. Nascimento. IND-CCA secure cryptography based on a variant of the LPN problem. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 485–503, 2012.

[22] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 40–56, 2013.

[23] L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 335–352, 2014.

[24] S. R. Dunbar. Topics in probability theory and stochastic processes. http://www.math.unl.edu/ sdunbar1/ProbabilityTheory/Lessons/BernoulliTrials /DeMoivreLaplaceCLT/demoivrelaplaceclt.pdf, 2011. [Online; accessed 7-January-2015].

[25] G. D. Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12(2):125–131, 1966.

[26] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206, 2008.

[27] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, New York, NY, USA, 2000.

[28] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32, 1989.

[29] S. Heyse, E. Kiltz, V. Lyubashevsky, C. Paar, and K. Pietrzak. Lapin: An efficient authentication protocol based on ring-lpn. In *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, pages 346–365, 2012.

[30] N. J. Hopper and M. Blum. Secure human identification protocols. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 52–66, 2001.

[31] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstracts). In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 12–24, 1989.

[32] J. Katz, J. S. Shin, and A. Smith. Parallel and concurrent security of the HB and hb^+ protocols. *J. Cryptology*, 23(3):402–421, 2010.

[33] E. Kiltz, D. Masny, and K. Pietrzak. Simple chosen-ciphertext security from low-noise lpn. In *Public Key Cryptography*, pages 1–18, 2014.

[34] V. Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 378–389, 2005.

[35] V. Lyubashevsky and D. Masny. Man-in-the-middle secure authentication schemes from LPN and weak prfs. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 308–325, 2013.

[36] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 372–381, 2004.

[37] A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 525–542, 2011.

[38] C. Peikert. An efficient and parallel gaussian sampler for lattices. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 80–97, 2010.

[39] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

[40] M. Sipser and D. A. Spielman. Expander codes. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 566–576, 1994.

[41] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 388–397, 1995.