

# Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise<sup>\*</sup>

Abhishek Jain<sup>1</sup>, Stephan Krenn<sup>2</sup>, Krzysztof Pietrzak<sup>2</sup>, and Aris Tentes<sup>3</sup>

<sup>1</sup> Massachusetts Institute of Technology, USA, and  
Boston University, USA  
`abhishek@csail.mit.edu`

<sup>2</sup> Institute of Science and Technology Austria  
`{stephan.krenn,pietrzak}@ist.ac.at`

<sup>3</sup> Department of Computer Science, New York University, USA  
`tentes@cs.nyu.edu`

**Abstract.** We construct a perfectly binding string commitment scheme whose security is based on the learning parity with noise (LPN) assumption, or equivalently, the hardness of decoding random linear codes. Our scheme not only allows for a simple and efficient zero-knowledge proof of knowledge for committed values (essentially a  $\Sigma$ -protocol), but also for such proofs showing any kind of relation amongst committed values, i.e., proving that messages  $\mathbf{m}_0, \dots, \mathbf{m}_u$ , are such that  $\mathbf{m}_0 = C(\mathbf{m}_1, \dots, \mathbf{m}_u)$  for any circuit  $C$ .

To get soundness which is exponentially small in a security parameter  $t$ , and when the zero-knowledge property relies on the LPN problem with secrets of length  $\ell$ , our 3 round protocol has communication complexity  $\mathcal{O}(t|C|\ell \log(\ell))$  and computational complexity of  $\mathcal{O}(t|C|\ell)$  bit operations. The hidden constants are small, and the computation consists mostly of computing inner products of bit-vectors.

## 1 Introduction

Commitment schemes and zero-knowledge proofs are fundamental cryptographic primitives. In this work we propose a simple string commitment scheme and show efficient zero-knowledge proofs for any relation amongst committed values. The security (more precisely, the computational hiding property) of our commitment scheme relies on the learning parity with noise (LPN) assumption, or equivalently, on the hardness of decoding random linear codes.

*Commitment schemes.* A commitment scheme allows a party to commit to a message  $\mathbf{m}$  by publishing a commitment  $\sigma$ , and this commitment can be opened at a later point in time. The security properties required are called the hiding and binding property. Hiding means that one cannot learn anything about the

---

<sup>\*</sup> This work was in part supported by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC Starting Grant (259668-PSPC).

committed message  $\mathbf{m}$  from the commitment  $\sigma$ , binding means that one cannot open a commitment  $\sigma$  to two different messages  $\mathbf{m} \neq \mathbf{m}'$ .

In our scheme, the commitment to a message  $\mathbf{m}$  is simply the encoding of  $\mathbf{m}$  using a random linear code, with some noise added to the codeword. Exploiting the linear structure of this scheme, we get simple and efficient zero-knowledge proofs for linear and multiplicative relations of committed values.

*Zero-knowledge proofs of knowledge.* Zero-knowledge proofs of knowledge are two party protocols, which allow a prover to convince a verifier that it knows some secret piece of information, without the verifier being able to learn anything about the secret value except for what is revealed by the claim itself.

*The LPN assumption.* The computationally hard problem underlying the security (i.e., the computational hiding property) of our commitment scheme is the *learning parity with noise* (LPN) assumption. This problem asks to distinguish “noisy” linear equations  $\mathbf{A} \cdot \mathbf{s} \oplus \mathbf{e}$  from uniformly random. Here  $\mathbf{A}$  is a “skinny” public random binary  $k \times \ell$  matrix,  $\mathbf{s}$  is a uniformly random  $\ell$  bit secret and  $\mathbf{e}$  is a random vector of low weight (the exact distribution of  $\mathbf{e}$  is discussed in §2.2). The LPN problem has found numerous applications as the assumption underlying provably secure cryptosystems, like secret-key [21,24,26,32] and public-key [36] authentication schemes or symmetric encryption [1,18].

LPN based cryptosystems are interesting for theoretical and practical reasons. On the one hand, the LPN problem is equivalent to the problem of decoding random linear codes, a problem that has been studied for over half a century [5,6,7,28,35]. The best known algorithms need  $2^{\Theta(\ell/\log \ell)}$  time and samples (the number of samples is given by the number  $k$  of rows of  $\mathbf{A}$ ) [7]. If  $k = \Theta(\ell)$  is linear in  $\ell$ , as it will be the case in this paper, the best algorithms need exponential  $2^{\Theta(\ell)}$  time. Furthermore, unlike most number-theoretic problems used in cryptography, the LPN problem is not known to become insecure against quantum algorithms. On the practical side, LPN based cryptosystems tend to be extremely simple and efficient, and thus are good candidates for weak devices like RFID tags, where existing cryptographic algorithms cannot be implemented due to constraints on code-size, running-time or memory.

### 1.1 Our Contributions

*Commitments from LPN.* In our scheme the commitment to a message  $\mathbf{m} \in \mathcal{I}^v$  (where  $\mathcal{I} \stackrel{\text{def}}{=} \{0, 1\}$ ) is simply

$$\text{Com}(\mathbf{m}) = \mathbf{A} \cdot (\mathbf{r} \parallel \mathbf{m}) \oplus \mathbf{e},$$

where  $\mathbf{A} = \mathbf{A}' \parallel \mathbf{A}'' \in \mathcal{I}^{k \times (\ell+v)}$  is a public random binary matrix,  $\mathbf{r} \in \mathcal{I}^\ell$  is a uniformly random vector and  $\mathbf{e} \in \mathcal{I}^k$  is a random low-weight vector. To open a commitment  $\sigma$ , one reveals  $\mathbf{r}, \mathbf{m}, \mathbf{e}$  and checks if  $\sigma \stackrel{?}{=} \mathbf{A} \cdot (\mathbf{r} \parallel \mathbf{m}) \oplus \mathbf{e}$  and  $\mathbf{e}$  is low weight. Here the length  $\ell = |\mathbf{r}|$  is chosen such that the LPN problem with secrets of length  $\ell$  is hard. The length  $v = |\mathbf{m}|$  of the message can be arbitrary, but for efficiency reasons it is best to choose it roughly of the same size as  $\ell$ .

Setting  $k = \Theta(v + \ell)$  large enough, the commitment scheme becomes computationally hiding and perfectly binding (with overwhelming probability over the choice of  $\mathbf{A}$ ). The binding property follows by the large distance of the code generated by the random matrix  $\mathbf{A}$ , the hiding property follows directly from the LPN assumption which implies that  $\mathbf{A}' \cdot \mathbf{r} \oplus \mathbf{e}$  is pseudorandom.

*Zero-knowledge protocols for arbitrary circuits.* We construct a zero-knowledge proof of knowledge, which is basically a so called  $\Sigma$ -protocol, that allows to prove knowledge of the message  $\mathbf{m}$  hidden inside a commitment without revealing anything about it. Furthermore, we give a protocol for proving that committed messages  $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2$  satisfy a linear relation  $\mathbf{m}_0 = \mathbf{X}_1 \cdot \mathbf{m}_1 \oplus \mathbf{X}_2 \cdot \mathbf{m}_2$  (for any square matrices  $\mathbf{X}_1, \mathbf{X}_2$ ). Based on this protocol, we construct proofs for any bitwise relations  $\mathbf{m}_0 = \mathbf{m}_1 \circ \mathbf{m}_2$ , where  $\circ$  can be any bitwise relation like AND, NAND, OR, NOR. As NAND is functionally complete, we can prove relations  $\mathbf{m}_0 = C(\mathbf{m}_1, \dots, \mathbf{m}_t)$  for any boolean circuit  $C$ .

For  $\mathbf{A} \in \mathcal{I}^{k \times m}$ , the communication complexity of our proofs is  $\Theta(k \log k)$ . Setting  $v = \ell$ , we can set  $k = \Theta(v + \ell) = \Theta(v)$ , thus the proofs are quasilinear in the length of the committed messages. The soundness error of our protocol is  $2/3$ . To get soundness errors of  $2^{-16}$  and  $2^{-32}$  as specified by the ISO/IEC-9798-5 standard we would need 28 and 55 repetitions, respectively.

As one application (which we bring up to compare our scheme to existing schemes in the related work section below) consider an  $\mathcal{NP}$  language  $L = \{x : \exists w : \mathcal{R}(x, w) = 1\}$ . Our scheme can be used to prove knowledge of a witness  $w$  for  $x \in L$  as follows: commit to  $\mathbf{m}_0 = w$  and  $\mathbf{m}_1 = 1$  and prove that the committed values satisfy the relation  $C_x(\mathbf{m}_0) = \mathbf{m}_1$  where  $C_x(\cdot)$  is the circuit computing the  $\mathcal{NP}$  relation  $\mathcal{R}(x, \cdot)$ . This proofs avoid expensive Karp reductions (to 3-coloring or Hamiltonian cycles) used in classical proofs.

### 1.2 Related Work

Our basic scheme for proving knowledge of a committed value is similar to Stern's [36] zero-knowledge proof of knowledge for the syndrome decoding problem, which can be seen as the "dual" of the LPN problem, and both are known to be  $\mathcal{NP}$ -complete [5]. Subsequent to Stern's work, Véron [37] proposed a  $\Sigma$ -protocol for proving knowledge of an LPN secret, but as we show in the full version of this paper [23], there is a gap in the proof of the ZK property of his protocol. Recently, several works have extended Stern's protocol to construct efficient identification schemes from various lattice-based and coding based assumptions (see [9,10,27] and references therein). In particular, Cayrel et al. [10] constructed an identification scheme with knowledge error  $1/2$  based on the  $q$ -ary syndrome decoding problem. However, this improvement in the knowledge error adds two additional rounds to the protocol, and thus their construction does not decrease the total number of rounds required to reach a specified knowledge error. Very recently, Asharov et al. [3] constructed  $\Sigma$ -protocols for various

learning with errors (LWE) related languages. We note that the ZK property of their protocols crucially relies on the ability to use large noise to “smudge out” small differences in distributions. Unfortunately, this technique does not extend to the setting of LPN (which is the focus of this work). We finally note that all the aforementioned works only construct ZK protocols for specific languages and, unlike our work, do not consider the general problem of constructing ZK proofs for circuit satisfiability.

There is a large body of work on efficient interactive or non-interactive zero-knowledge proofs and arguments, see, e.g., [8,13,14,15,20,22,25,30,31] and the references therein. For ZK *arguments* (as opposed to proofs), where the soundness property is only required to hold against computationally bounded malicious provers, one can construct schemes which asymptotically only require polylogarithmic communication (e.g., the interactive argument based on CRHFs [29] or the non-interactive argument in the random-oracle model [33]). These schemes rely on probabilistically checkable proofs (PCP), and are not really practical.

The beautiful work of Ishai et al. [22] on zero-knowledge proofs from secure multiparty computation aims at a similar goal as this work. They show how to construct ZK proofs from MPC; When instantiated with simple MPC protocols like GMW [19] they get ZK proofs for showing knowledge of a witness  $w$  such that  $\mathcal{R}(x, w) = 1$  with communication complexity  $O(ts)$ , where  $2^{-t}$  is the soundness error and  $s$  is the size of the circuit computing the relation  $\mathcal{R}(x, \cdot)$ , which is the same asymptotic behavior we get (as explained in the previous section). Using protocols relying on sophisticated secret sharing schemes for constant-size fields based on algebraic-geometric codes [11] they even get an asymptotic communication complexity of  $O(s) + \text{poly}(t, \log s)$ , but due to the large hidden constants in such codes this scheme will only be more efficient than the simpler scheme for very large circuits.

A ZK proof for any  $\mathcal{NP}$  relation can of course be used to prove any relation amongst *committed* values, but in general this would be rather expensive as the computation of the opening of the commitment must be part of the description of the relation. In contrast, our ZK proofs work directly on committed values, and we do not pay extra for this. Proving relations amongst *committed* values has been considered before, see [15] and references therein. These works give very efficient proofs for algebraic circuits over large fields, but are less suited for circuits over very small ones, in particular, for  $\mathbb{Z}_2$  as in boolean circuits. As an application, consider the case where we need to prove that committed values satisfy  $\mathbf{m}_0 = \text{AES}(\mathbf{m}_1, \mathbf{m}_2)$ , i.e.,  $\mathbf{m}_0$  is the output of the AES block-cipher under key  $\mathbf{m}_1$  on input  $\mathbf{m}_2$ .

### 1.3 Outline

We introduce some notation and recapitulate the basic definitions required for this paper in Section 2. In Section 3 we present a very simple commitment scheme based on the hardness of the LPN problem. Protocols allowing one to

prove knowledge of the content of such commitments, and relations among them, are presented in Section 4. We finally conclude in Section 5.

## 2 Preliminaries

We use bold lower-case and upper-case letters like  $\mathbf{a}, \mathbf{A}$  to denote vectors and matrices, respectively. Probabilistic polynomial time (PPT) algorithms are written by sans-serif letters like  $\mathbf{A}$ . Calligraphic letters like  $\mathcal{A}$  always denote sets. We write  $a \stackrel{\mathcal{R}}{\leftarrow} \mathcal{A}$  if  $a$  was drawn uniformly at random from set  $\mathcal{A}$ ,  $a \stackrel{\mathcal{R}}{\leftarrow} \chi$  if  $a$  was drawn according to some probability distribution  $\chi$ , and  $a \stackrel{\mathcal{R}}{\leftarrow} \mathbf{A}$  if  $a$  is the output of a randomized algorithm  $\mathbf{A}$ .

We denote the set  $\{0, 1\}$  by  $\mathcal{I}$ , thus  $\mathcal{I}^k$  denotes the set of strings of length  $k$ . The Hamming weight of  $\mathbf{a} \in \mathcal{I}^k$  is denoted by  $\|\mathbf{a}\|_1 = \sum_{i=1}^k \mathbf{a}[i]$ . With  $\mathcal{I}_w^k = \{\mathbf{a} \in \mathcal{I}^k : \|\mathbf{a}\|_1 = w\}$  we denote the set of all  $k$ -bit vectors of weight exactly  $w$ . The all-zeros and all-ones vectors of length  $k$  are denoted by  $\mathbf{0}^k$  and  $\mathbf{1}^k$ , respectively. The concatenation of vectors  $\mathbf{a}$  and  $\mathbf{b}$  is written as  $\mathbf{a}\|\mathbf{b}$ . The symmetric group on  $k$  elements (i.e., the set of all permutations on  $k$  elements) is denoted  $\mathcal{S}_k$ . For  $\pi \in \mathcal{S}_k$  and  $\mathbf{a} \in \mathcal{I}^k$ ,  $\pi(\mathbf{b})$  denotes the string  $\mathbf{a}[i] = \mathbf{b}[\pi(i)]$ .

### 2.1 Commitment Schemes

**Definition 2.1.** *A triple of algorithms  $(\mathbf{KGen}, \mathbf{Com}, \mathbf{Ver})$  is called a commitment scheme if it satisfies the following:*

- On input  $1^\ell$ , the key generation algorithm  $\mathbf{KGen}$  outputs a public commitment key  $pk$ .
- The commitment algorithm  $\mathbf{Com}$  takes as inputs a message  $m$  from a message space  $\mathcal{M}$  and a commitment key  $pk$ , and outputs a commitment/opening pair  $(c, d)$ .
- The verification algorithm  $\mathbf{Ver}$  takes a key  $pk$ , a message  $m$ , a commitment  $c$  and an opening  $d$  and outputs 1 or 0.

The commitment scheme we construct satisfies the following security properties:

- *Correctness:*  $\mathbf{Ver}$  evaluates to 1 whenever the inputs were computed by an honest party, i.e.,

$$\Pr[\mathbf{Ver}(pk, m, c, d) = 1; pk \stackrel{\mathcal{R}}{\leftarrow} \mathbf{KGen}(1^\ell), m \in \mathcal{M}, (c, d) \stackrel{\mathcal{R}}{\leftarrow} \mathbf{Com}(m, pk)] = 1$$

- *Perfect binding:* With overwhelming probability over the choice of the public key  $pk \stackrel{\mathcal{R}}{\leftarrow} \mathbf{KGen}(1^\ell)$ , no commitment  $c$  can be opened in two different ways, i.e.,

$$(\mathbf{Ver}(pk, m, c, d) = 1) \wedge (\mathbf{Ver}(pk, m', c, d') = 1) \Rightarrow m = m'$$

- *Computational hiding:* A commitment  $c$  computationally hides the committed message: with overwhelming probability over the choice of  $pk \stackrel{\mathcal{R}}{\leftarrow} \mathbf{KGen}(1^\ell)$ , for every  $m, m' \in \mathcal{M}$  and  $(c, d) \stackrel{\mathcal{R}}{\leftarrow} \mathbf{Com}(m, pk), (c', d') \stackrel{\mathcal{R}}{\leftarrow} \mathbf{Com}(m', pk)$  the distributions  $c$  and  $c'$  are computationally indistinguishable.

## 2.2 Learning Parity with Noise

The computational assumption underlying all our constructions is the *learning parity with noise* (LPN) assumption. Below we define the decisional version of LPN in a general form, not yet specifying the error distribution.

**Definition 2.2.** For  $k, \ell \in \mathbb{N}$ , let  $\chi$  be an error distribution over  $\mathcal{I}^k$ . The decisional  $(\chi, \ell, k)$ -LPN problem is  $(s, \epsilon)$ -hard if for every distinguisher  $D$  of size  $s$ :

$$\left| \Pr_{\mathbf{x}, \mathbf{A}, \mathbf{e}} [\mathbf{D}(\mathbf{A}, \mathbf{A} \cdot \mathbf{x} \oplus \mathbf{e}) = 1] - \Pr_{\mathbf{r}, \mathbf{A}} [\mathbf{D}(\mathbf{A}, \mathbf{r}) = 1] \right| \leq \epsilon$$

where  $\mathbf{A} \xleftarrow{R} \mathcal{I}^{k \times \ell}$ ,  $\mathbf{e} \xleftarrow{R} \chi_k$ ,  $\mathbf{r} \xleftarrow{R} \mathcal{I}^k$ , and  $\mathbf{x} \xleftarrow{R} \mathcal{I}^\ell$  is fixed and secret. The search version is defined similarly, but we require that no  $D$  can find the secret  $\mathbf{x}$ :

$$\left| \Pr_{\mathbf{x}, \mathbf{A}, \mathbf{e}} [\mathbf{D}(\mathbf{A}, \mathbf{A} \cdot \mathbf{x} \oplus \mathbf{e}) = \mathbf{x}] \right| \leq \epsilon$$

In the standard definition of the LPN problem, the error distribution  $\chi$  is the Bernoulli distribution with some parameter  $0 < \tau < \frac{1}{2}$ , i.e., every bit  $e[i]$  is chosen independently and identically distributed with  $\Pr[e[i] = 1] = \tau$ , we will refer to this version as  $\text{LPN}_\tau$ . As mentioned in the introduction, for  $k = \Theta(\ell)$  as used in this paper, the search version of  $\text{LPN}_\tau$  is the same as the problem of decoding random linear codes, and is believed to be exponentially hard. The search and decision version of  $\text{LPN}_\tau$  are known to be equivalent [6,26], but to show this search to decision reduction, the number of samples  $k$  in the decision version must be much larger than in the search version (by a factor of  $\Omega(\ell/\epsilon)$ ). More recently, a sample preserving reduction has been shown [2, Lemma 4.4]. (cf. [34] for a more general treatment of sample preserving reductions).

*Exact LPN.* In this work we define a new version of the LPN problem, which we call *exact LPN* or  $\text{xLPN}$  for short. Similar to  $\text{LPN}_\tau$ ,  $\text{xLPN}$  is parameterized by some noise parameter  $0 < \tau < \frac{1}{2}$ , and the (search or decision)  $\text{xLPN}_\tau$  problem is defined exactly like  $\text{LPN}_\tau$ , except that the Hamming weight of the error vector is exactly  $\lfloor k\tau \rfloor$  (not of expected weight  $k\tau$  as in  $\text{LPN}_\tau$ ). That is,  $e$  is sampled uniformly at random from the set  $\mathcal{I}_{\lfloor k\tau \rfloor}^k$ .

In this work, we assume the hardness of *decisional*  $\text{xLPN}$ .<sup>1</sup> It is not hard to see that *search*  $\text{xLPN}_\tau$  is hard iff search  $\text{LPN}_\tau$  is hard.<sup>2</sup> Showing equivalence of *decisional*  $\text{xLPN}_\tau$  and  $\text{LPN}_\tau$  version is more tricky. The classical search to decision reduction for  $\text{LPN}_\tau$  from [6,26] does not work for  $\text{xLPN}_\tau$ , but the sample preserving reduction [2, Lemma 4.4] does. Summing up, we have

<sup>1</sup> The security of the basic commitment scheme can be based on decisional LPN, but our  $\Sigma$ -protocols to prove relations amongst committed values “leak” the weight of the error vectors. Thus, to be zero-knowledge, we need this value to be fixed.

<sup>2</sup> Any  $D$  who outputs  $\mathbf{x}$  with advantage  $\epsilon$  for  $\text{xLPN}_\tau$ , will output the secret  $\mathbf{x}$  with advantage at least  $\epsilon/\sqrt{k}$  of  $\text{LPN}_\tau$ , as the error vector sampled in  $\text{LPN}_\tau$  has weight  $\lfloor k\tau \rfloor$  with probability  $\geq 1/\sqrt{k}$ , and conditioned on this being the case, the error distribution is exactly the same as in  $\text{xLPN}_\tau$ .

**Proposition 2.3.** *The hardness of decisional  $x\text{LPN}_\tau$  (used in this paper) is polynomially related to the hardness of search  $\text{LPN}_\tau$ .*

The sample preserving reduction [2, Lemma 4.4] relies on the Goldreich-Levin theorem, and as a consequence is not very tight. Although we do not know of significantly more efficient attacks against  $x\text{LPN}_\tau$  than against  $\text{LPN}_\tau$ , if one insists on basing the security of our schemes on the standard  $\text{LPN}_\tau$  assumption in a provable manner, one must take the loss in the reduction into account, which would result in rather large parameters. A protocol relying on the security of the standard decisional  $\text{LPN}_\tau$  assumption can be found in the full version of this paper. The protocol given there can be extended to prove arbitrary relations amongst committed values, in the same manner as in the case of  $x\text{LPN}_\tau$  assumption. However, this protocol is somewhat more complicated and has a worse soundness error ( $4/5$  as compared to  $2/3$ ), and thus requires roughly twice the number of repetitions in order to achieve the same knowledge error.

As suggested in [26, Section 5], replacing the LPN assumption with an assumption where we have a fixed upper bound on the weight of the error vector (like it is the case in  $x\text{LPN}$ ) would remove the completeness error (and thus allows for more efficient instantiations) also for other LPN based schemes, like HB type protocols. We thus think that investigating the exact hardness of the  $x\text{LPN}$ -problem is of interest beyond the realm of this work.

### 2.3 Zero-Knowledge Proofs of Knowledge and $\Sigma$ -Protocols

Informally, a zero-knowledge proof of knowledge is a two party protocol between a prover  $P$  and a verifier  $V$  which allows the former to convince the latter that it knows some secret piece of information without revealing anything about it. A bit more precisely, in a zero-knowledge proof for a binary relation  $\mathcal{R}$ , the parties have common input  $y$  and the prover has private input  $w$  such that  $(y, w) \in \mathcal{R}$ . The protocol must then satisfy the following three properties: (i) For an honest prover, the verifier always accepts (*completeness*). (ii) For every potentially malicious verifier  $V^*$  there exists a PPT simulator only taking  $y$  as an input whose output is indistinguishable from conversations of  $V^*$  with an honest prover (*zero-knowledge*). (iii) From every prover  $P^*$  which can make the verifier accept with a probability larger than a threshold  $\kappa$  (the *knowledge error*), a  $w'$  satisfying  $(y, w') \in \mathcal{R}$  can be extracted efficiently in a rewindable black-box way (*proof of knowledge*). For a formal definition we refer to Bellare and Goldreich [4].

The protocols we are going to design in the following are all instantiations of the following definition:

**Definition 2.4 ( $\Sigma$ -Protocol).** *Let  $(P, V)$  be a two-party protocol, where  $V$  is PPT, and let  $\mathcal{R}$  be a binary relation. Then  $(P, V)$  is called a  $\Sigma$ -protocol for  $\mathcal{R}$  with challenge set  $\mathcal{C}$ , public input  $y$  and private input  $w$ , if and only if it satisfies the following conditions:*

- **3-move form:** *The protocol is of the following form:*
  - *The prover  $P$  computes a commitment  $t$  and sends it to  $V$ .*

- The verifier  $\mathcal{V}$  draws a challenge  $c \xleftarrow{R} \mathcal{C}$  and sends it to  $\mathcal{P}$ .
- The prover sends a response  $s$  to the verifier.
- Depending on the protocol transcript  $(t, c, s)$ , the verifier accepts or rejects the proof.

The protocol transcript  $(t, c, s)$  is called accepting, if the verifier accepts the protocol run.

- **Completeness:** The verifier  $\mathcal{V}$  accepts whenever  $(y, w) \in \mathcal{R}$ .
- **Special soundness:** There exists a PPT algorithm  $\mathcal{E}$  (the knowledge extractor) which takes a set  $\{(t, c, s_c) : c \in \mathcal{C}\}$  of accepting transcripts with the same commitment as inputs, and outputs  $w'$  such that  $(y, w') \in \mathcal{R}$ .
- **Special honest-verifier zero-knowledge:** There exists a PPT algorithm  $\mathcal{S}$  (the simulator) taking  $y$  and  $c \in \mathcal{C}$  as inputs, and which outputs triples  $(t, c, s)$  whose distribution is (computationally) indistinguishable from accepting protocol transcripts generated by real protocol runs.

It is well known that every  $\Sigma$ -protocol is also a proof of knowledge for the same relation [16]. However, while in  $\Sigma$ -protocols the existence of a simulator is only required for the honest verifier, zero-knowledge proofs require this existence for arbitrary, potentially malicious, verifiers. This can be reached by applying generic standard techniques to  $\Sigma$ -protocols, e.g., Damgård et al. [17].

We note that our definition of  $\Sigma$ -protocols slightly differs from the standard definition found in the literature [12,16]. For the special soundness property, it is typically required that a valid witness can already be computed given any *two* accepting conversations with the same commitment but different challenges. We loosen this definition and only require that  $w'$  can be computed given valid responses to *all* challenges for a fixed commitment  $t$ . It can easily be seen that the aforementioned results showing that every  $\Sigma$ -protocol is also a proof of knowledge still hold true. However, while for the standard definition the knowledge error is given by  $1/\#\mathcal{C}$  it is only given by  $1 - 1/\#\mathcal{C}$  for Definition 2.4.

### 3 Perfectly Binding String Commitments from LPN

Our commitment scheme is parameterized by the main security parameter  $\ell \in \mathbb{N}$ ,  $0 < \tau < 0.25$ , the message length  $v \in \mathbb{N}$  and  $k \in \mathcal{O}(\ell + v)$ . Finally, we set  $w = \lfloor \tau k \rfloor$ . The algorithms of the commitment scheme are then given as follows:

- **KGen:** The public commitment key consists of the matrix  $\mathbf{A} = \mathbf{A}' \parallel \mathbf{A}'' \in \mathcal{I}^{k \times (\ell + v)}$ , where  $\mathbf{A}' \xleftarrow{R} \mathcal{I}^{k \times \ell}$  and  $\mathbf{A}'' \xleftarrow{R} \mathcal{I}^{k \times v}$ .
- **Com:** The commitment to a message  $\mathbf{m} \in \mathcal{I}^v$  is given by  $\mathbf{A} \cdot (\mathbf{r} \parallel \mathbf{m}) \oplus \mathbf{e}$ , where  $\mathbf{r} \xleftarrow{R} \mathcal{I}^\ell$  and  $\mathbf{e} \xleftarrow{R} \mathcal{I}_w^k$ . The opening of the commitment is given by  $\mathbf{m}$  and  $\mathbf{r}$ .
- **Ver:** Given a commitment  $\mathbf{c}$ , a message  $\mathbf{m}'$  and a randomness  $\mathbf{r}'$ , a verifier accepts if and only if  $\mathbf{e}' = \mathbf{c} \oplus \mathbf{A} \cdot (\mathbf{r}' \parallel \mathbf{m}')$  has weight  $w$ .

**Theorem 3.1.** *Let  $0 < \tau < 0.25$ , and  $\ell, k, v \in \mathbb{N}$  be such that the decisional  $x\text{LPN}_\tau$  problem (with secrets of length  $\ell$  and  $k$  samples) is hard. Let  $k = \Theta(\ell + v)$  be such that with overwhelming probability a randomly chosen generator matrix*



of a linear code  $\mathbf{A} \in \mathcal{I}^{k \times (\ell+v)}$  has distance larger than  $2w$ , i.e.,  $\|\mathbf{A} \cdot \mathbf{x}\|_1 > 2w$  for all  $\mathbf{x} \in \mathcal{I}^{\ell+v}$ . Then the above commitment scheme is perfectly binding and computationally hiding.

*Proof.* The required security properties can be seen as follows:

*Perfect binding.* Assume, by contraposition, that  $\mathbf{m}_i, \mathbf{r}_i, i = 1, 2$  are two different openings for a commitment  $\mathbf{c}$ . That is, we have that  $\mathbf{e}_i = \mathbf{c} \oplus \mathbf{A} \cdot (\mathbf{r}_i \| \mathbf{m}_i)$  has norm at most  $w$  for  $i = 1, 2$ . Thus we have that  $\mathbf{e}_1 \oplus \mathbf{e}_2 = \mathbf{A} \cdot (\mathbf{r}_1 \| \mathbf{m}_1 \oplus \mathbf{r}_2 \| \mathbf{m}_2)$  is a codeword of length  $\|\mathbf{e}_1 \oplus \mathbf{e}_2\|_1 \leq \|\mathbf{e}_1\|_1 + \|\mathbf{e}_2\|_1 \leq 2w$ , contradicting our assumption on the distance of the code generated by  $\mathbf{A}$ .

*Computational hiding.* We have that  $\mathbf{c} = \mathbf{A}' \cdot \mathbf{r} \oplus \mathbf{e} \oplus \mathbf{A}'' \cdot \mathbf{m}$ . By the  $\text{xLPN}_\tau$ -assumption  $\mathbf{A}' \cdot \mathbf{r} \oplus \mathbf{e}$ , and thus also  $\mathbf{c}$ , is pseudorandom.  $\square$

## 4 Zero-Knowledge Proofs of Knowledge

In this section we first construct a  $\Sigma$ -protocol, which on common input  $\mathbf{A}$  and  $\mathbf{y}$  allows the prover to prove knowledge of a valid opening of  $\mathbf{y}$  under the commitment scheme presented in Section 3. The protocol borrows some basic ideas from Stern [36], who gave a  $\Sigma$ -protocol for the syndrome decoding problem.

After presenting this basic protocol, we give two further  $\Sigma$ -protocols. The first can be used to prove that committed strings satisfy any linear relation. The second protocol can be used to show that committed strings satisfy any bitwise relation like bitwise AND, NAND, OR or NOR. As NAND is functionally complete, using this protocol we can construct  $\Sigma$ -protocols for any relation amongst committed messages.

### 4.1 Proving Knowledge of a Valid Opening

The following  $\Sigma$ -protocol proves knowledge of a valid opening for commitments of the form described in the previous section, i.e., it shows possession of  $\mathbf{r}, \mathbf{m}, \mathbf{e}$  such that  $\mathbf{y} = \mathbf{A} \cdot (\mathbf{r} \| \mathbf{m}) \oplus \mathbf{e}$  for an error satisfying  $\|\mathbf{e}\|_1 = w$ . For notational convenience we will sometimes write  $\mathbf{s}$  to denote the vector  $\mathbf{r} \| \mathbf{m}$ .

A first idea for such a protocol (which will not quite work) is to mimic Schnorr’s protocol as follows: (1) the prover  $\mathbf{P}$  commits to some value  $\mathbf{t}_0 = \mathbf{A} \cdot \mathbf{v} \oplus \mathbf{f}$ , (2) the verifier  $\mathbf{V}$  sends a challenge  $c \stackrel{\mathbb{R}}{\leftarrow} \{0, 1\}$ , (3) the prover opens  $\mathbf{t}_0$  (i.e., sends  $\mathbf{v}, \mathbf{f}$ ) if  $c = 0$  and opens  $\mathbf{t}_0 \oplus \mathbf{y}$  (i.e., sends  $\mathbf{v} \oplus \mathbf{s}, \mathbf{f} \oplus \mathbf{e}$ ) if  $c = 1$ .

If in this protocol  $\mathbf{f}$  is sampled such that it has low weight, then  $\mathbf{e} \oplus \mathbf{f}$  leaks information about  $\mathbf{e}$ , and the protocol is not zero-knowledge. On the other hand, if  $\mathbf{f}$  is uniformly random (so  $\mathbf{e} \oplus \mathbf{f}$  is independent of  $\mathbf{e}$ ), the protocol is not sound (informally, all we can say is that from answers to both challenges we can extract  $\mathbf{s}', \mathbf{e}'$  where  $\mathbf{y} = \mathbf{A} \cdot \mathbf{s}' \oplus \mathbf{e}'$ , but  $\mathbf{e}'$  can have arbitrary weight, and finding such a solution is trivial). In our protocol  $\mathbf{f}$  is chosen uniformly at random, and to ensure soundness we use a trick from Stern [36]. We additionally commit to a

random permutation  $\pi \in \mathcal{S}_k$  and to  $\pi(\mathbf{f}), \pi(\mathbf{f} \oplus \mathbf{e})$ . On challenge  $c = 0$  and  $c = 1$  we now additionally make sure the openings are consistent with the committed errors by opening  $\pi$  and either  $\pi(\mathbf{f})$  (if  $c = 0$ ) or  $\pi(\mathbf{f} \oplus \mathbf{e})$  (if  $c = 1$ ). Moreover we extend the challenge space from two to three. The extra challenge  $c = 2$  is used to verify that the weight of  $\pi(\mathbf{f}) \oplus \pi(\mathbf{f} \oplus \mathbf{e}) = \pi(\mathbf{e})$  (and thus  $\mathbf{e}$ ) is small, this will ensure soundness, as from valid answers to all three challenges we can extract  $\mathbf{s}', \mathbf{e}'$  where  $\mathbf{y} = \mathbf{A}.\mathbf{s}' \oplus \mathbf{e}'$  and  $\mathbf{e}'$  has low weight. Opening the commitments to  $\pi(\mathbf{f}), \pi(\mathbf{f} \oplus \mathbf{e})$  on  $c = 2$  does not hurt the ZK property, as  $\pi(\mathbf{f}), \pi(\mathbf{f} \oplus \mathbf{e})$  contains no information about  $\mathbf{e}$  except its weight.

The common input to P, V is  $\mathbf{A}$  and  $\mathbf{y}$ , P's secret input is  $(\mathbf{e}, \mathbf{s})$ . The protocol flow is then given as follows, where the commitment scheme  $\text{Com}(\cdot)$  can be instantiated by an arbitrary perfectly binding string commitment scheme, potentially the scheme presented in Section 3 itself.

- P samples a permutation  $\pi \xleftarrow{\mathbb{R}} \mathcal{S}_k$  at random.  
 It then draws  $\mathbf{v} \xleftarrow{\mathbb{R}} \mathcal{I}^{\ell+v}, \mathbf{f} \xleftarrow{\mathbb{R}} \mathcal{I}^k$ , and then sends the following commitments to the verifier V:
 
$$\begin{aligned} C_0 &\leftarrow \text{Com}(\pi' = \pi, \mathbf{t}_0 = \mathbf{A}.\mathbf{v} \oplus \mathbf{f}) \\ C_1 &\leftarrow \text{Com}(\mathbf{t}_1 = \pi(\mathbf{f})) \\ C_2 &\leftarrow \text{Com}(\mathbf{t}_2 = \pi(\mathbf{f} \oplus \mathbf{e})) \end{aligned}$$
- The verifier draws  $c \xleftarrow{\mathbb{R}} \mathbb{Z}_3$  and sends it to P.
- Depending on the value of  $c$ , P opens the following commitments:
  0. P opens  $C_0, C_1$  by sending  $\pi', \mathbf{t}_0, \mathbf{t}_1$  and the associated random coins.
  1. P opens  $C_0, C_2$  by sending  $\pi', \mathbf{t}_0, \mathbf{t}_2$  and the associated random coins.
  2. P opens  $C_1, C_2$  by sending  $\mathbf{t}_1, \mathbf{t}_2$  and the associated random coins.
- The verifier verifies the correctness of the openings received from the prover, and additionally performs the following checks depending on the challenge  $c$ :
  0. V accepts, iff  $\mathbf{t}_0 \oplus \pi'^{-1}(\mathbf{t}_1) \stackrel{?}{\in} \text{img } \mathbf{A}$  and  $\pi' \stackrel{?}{\in} \mathcal{S}_k$ .
  1. V accepts, iff  $\mathbf{t}_0 \oplus \pi'^{-1}(\mathbf{t}_2) \oplus \mathbf{y} \stackrel{?}{\in} \text{img } \mathbf{A}$ .
  2. V accepts, iff  $\|\mathbf{t}_1 \oplus \mathbf{t}_2\|_1 = w$ .

**Theorem 4.1.** *The above protocol is a  $\Sigma$ -protocol for the following relation:*

$$\mathcal{R}_{LPN} = \{((\mathbf{A}, \mathbf{y}), (\mathbf{r}, \mathbf{m}, \mathbf{e})) : \mathbf{y} = \mathbf{A}.\mathbf{r} \parallel \mathbf{m} \oplus \mathbf{e} \quad \wedge \quad \|\mathbf{e}\|_1 = w\}$$

*Proof.* The 3-move form required for Definition 2.4 is clear. The remaining properties can be seen as follows.

*Completeness.* It is easy to see that an honest prover can always convince the verifier. Depending on the challenge  $c$ , we get:

0.  $\mathbf{t}_0 \oplus \pi'^{-1}(\mathbf{t}_1) = (\mathbf{A}.\mathbf{v} \oplus \mathbf{f}) \oplus \pi^{-1}(\pi(\mathbf{f})) = \mathbf{A}.\mathbf{v} \in \text{img } \mathbf{A}$  and  $\pi$  is a permutation.
1.  $\mathbf{t}_0 \oplus \pi'^{-1}(\mathbf{t}_2) \oplus \mathbf{y} = (\mathbf{A}.\mathbf{v} \oplus \mathbf{f}) \oplus \pi^{-1}(\pi(\mathbf{f} \oplus \mathbf{e})) \oplus (\mathbf{A}.\mathbf{s} \oplus \mathbf{e}) = \mathbf{A}.\mathbf{v} \oplus \mathbf{s} \in \text{img } \mathbf{A}$ .
2.  $\|\mathbf{t}_1 \oplus \mathbf{t}_2\|_1 = \|\pi(\mathbf{f}) \oplus \pi(\mathbf{f} \oplus \mathbf{e})\|_1 = \|\mathbf{f} \oplus \mathbf{f} \oplus \mathbf{e}\|_1 = \|\mathbf{e}\|_1 = w$ .

*Special Soundness.* Assume that we have fixed values  $C_0, C_1, C_2$  and openings for all challenges  $c \in \mathbb{Z}_3$ , such that the verifier accepts on all of them. Then, by the assumed perfect binding property of the underlying commitment scheme  $\text{Com}(\cdot)$ , we know that the openings to identical commitments must be identical across different challenges.

By adding the verification equations for  $c = 0$  and  $c = 1$  we get that  $\pi'^{-1}(\mathbf{t}_1 \oplus \mathbf{t}_2) \oplus \mathbf{y} \in \text{img } \mathbf{A}$  and thus that  $\mathbf{y} = \mathbf{A} \cdot \mathbf{s}' \oplus \pi'^{-1}(\mathbf{t}_1 \oplus \mathbf{t}_2)$ , where  $\mathbf{s}' = (\mathbf{r}' \parallel \mathbf{m}')$  is easy to compute. Now, using that  $\|\mathbf{t}_1 \oplus \mathbf{t}_2\|_1 = w$ , we have a valid witness of  $(\mathbf{A}, \mathbf{y})$  is thus given by  $(\mathbf{r}', \mathbf{m}', \pi'^{-1}(\mathbf{t}_1 \oplus \mathbf{t}_2))$ .

*Honest-Verifier Zero-Knowledge.* In the following we describe an efficient simulator  $\mathbf{S}$ , which for each challenge  $c \in \mathbb{Z}_3$  outputs an accepting protocol transcript the distribution of which is computationally indistinguishable from real protocol transactions with an honest prover for challenge  $c$ .

0. The simulator  $\mathbf{S}$  computes  $C_0$  and  $C_1$  like an honest prover, and computes  $C_2$  as a commitment to 0. Then, clearly, the distribution of  $C_0, C_1, \pi', \mathbf{t}_0, \mathbf{t}_1$  is identical to that in real protocol transcripts. Furthermore, by the computational hiding property of the commitment scheme  $\text{Com}(\cdot)$ , the distribution of  $C_2$  is computationally indistinguishable from that in real protocol runs.
1. For  $c = 1$ , the simulator draws  $\pi \stackrel{\mathbb{R}}{\leftarrow} \mathcal{S}_k$ ,  $\mathbf{a} \stackrel{\mathbb{R}}{\leftarrow} \mathcal{I}^k$  and  $\mathbf{b} \stackrel{\mathbb{R}}{\leftarrow} \mathcal{I}^{\ell+v}$ . It sets  $C_0 = \text{Com}(\pi, \mathbf{A} \cdot \mathbf{b} \oplus \mathbf{y} \oplus \mathbf{a})$  and  $C_2 = \text{Com}(\pi(\mathbf{a}))$ . The value of  $C_1$  is computed as commitments to 0. It is easy to see that the openings of  $C_0, C_2$  pass the verification equations. To see the correctness of their distributions note that  $\mathbf{t}_2$  in the real protocol run and  $\pi(\mathbf{a})$  in the simulated run are perfectly uniform in  $\mathcal{I}^k$ , and the permutations are also equally distributed both times. Concerning the opening of  $C_0$ , note the following: in the real protocol run, we have  $\mathbf{t}_0 = \mathbf{A} \cdot \mathbf{v} \oplus \mathbf{f}$ , where  $\mathbf{v}$  is uniformly at random, and  $\mathbf{f} = \pi^{-1}(\mathbf{t}_2 \oplus \mathbf{e})$ ; in the simulated transcript the content of  $C_0$  is given by  $\mathbf{A} \cdot (\mathbf{b} \oplus \mathbf{s}) \oplus (\mathbf{a} \oplus \mathbf{e})$ . Now,  $\mathbf{v}$  and  $\mathbf{b} \oplus \mathbf{s}$  are both uniformly random, and the terms  $\mathbf{f}$  and  $\mathbf{a} \oplus \mathbf{e}$  are uniquely determined by the contents of  $C_0$  and  $C_2$ . Thus, the distributions of  $C_0, C_2$  and their openings are perfectly simulated. The distribution of  $C_1$  is computationally indistinguishable by the assumed hiding property of  $\text{Com}(\cdot)$ .
2. Finally, for  $c = 2$ , the simulator draws  $\mathbf{a} \stackrel{\mathbb{R}}{\leftarrow} \mathcal{I}^k$  and  $\mathbf{b} \leftarrow \mathcal{I}_w^k$  uniformly at random. It computes  $C_0$  as a commitment to 0,  $C_1 = \text{Com}(\mathbf{a})$  and  $C_2 = \text{Com}(\mathbf{a} \oplus \mathbf{b})$ . As before, the distributions of  $C_0$  is computationally indistinguishable from real protocol runs by the binding property of  $\text{Com}(\cdot)$ , and  $C_1$  and  $C_2$  as well as their openings can easily be seen to perfectly simulate the behavior of an honest prover.  $\square$

## 4.2 Proving Linear Relations

We next describe a  $\Sigma$ -protocol which allows to prove that the messages hidden within commitments  $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_2$  (where  $\mathbf{y}_i = \mathbf{A} \cdot (\mathbf{r}_i \parallel \mathbf{m}_i) \oplus \mathbf{e}_i$ ) satisfy arbitrary linear relations. That is,  $\mathbf{X}_1 \cdot \mathbf{m}_1 \oplus \mathbf{X}_2 \cdot \mathbf{m}_2 = \mathbf{m}_3$  for arbitrary matrices

$\mathbf{X}_1, \mathbf{X}_2 \in \mathcal{I}^{v \times v}$ . The computational and communication complexity of the protocol is roughly the same as for proving the knowledge of the three committed messages using the protocol from the previous section, proving that they also satisfy the linear relation comes almost for free.

The high level idea of the protocol is as follows.  $\mathbf{P}$  and  $\mathbf{V}$  run the protocol from the previous section to prove knowledge of  $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$  for all the messages in parallel (but using the same challenge for all three). Recall that (oversimplifying a bit by ignoring the issue with the errors, i.e., the challenge  $c = 2$ ) this protocol goes as follows:  $\mathbf{P}$  commits to three random messages  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ , and later opens the  $\mathbf{v}_i$ 's (if  $c = 0$ ) or  $\mathbf{v}_i \oplus \mathbf{m}_i$  (if  $c = 1$ ). We change this protocol now a bit, and instead choosing  $\mathbf{v}_3$  at random we compute it as  $\mathbf{v}_3 = \mathbf{X}_1.\mathbf{v}_1 \oplus \mathbf{X}_2.\mathbf{v}_2$ . Moreover the verifier now additionally checks if  $\mathbf{v}_3 \stackrel{?}{=} \mathbf{X}_1.\mathbf{v}_1 \oplus \mathbf{X}_2.\mathbf{v}_2$  (if  $c = 0$ ) and if  $(\mathbf{v}_3 \oplus \mathbf{m}_3) \stackrel{?}{=} \mathbf{X}_1.(\mathbf{v}_1 \oplus \mathbf{m}_1) \oplus \mathbf{X}_2.(\mathbf{v}_2 \oplus \mathbf{m}_2)$  (if  $c = 1$ ). With these changes, we get a stronger soundness property: not only can we extract the committed messages  $\mathbf{m}_i$  from accepting answers to both challenges, but they will also satisfy  $\mathbf{m}_3 = \mathbf{X}_1.\mathbf{m}_1 \oplus \mathbf{X}_2.\mathbf{m}_2$ . At the same time the zero-knowledge property is not weakened, except of course for leaking the fact that the  $\mathbf{m}_i$ 's satisfy this linear relation.

The protocol flow is defined as follows:

- $\mathbf{P}$  samples permutations  $\pi_1, \pi_2, \pi_3$  at random.

It then draws  $\mathbf{v}_1, \mathbf{v}_2 \stackrel{\mathbf{R}}{\leftarrow} \mathcal{I}^v, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \stackrel{\mathbf{R}}{\leftarrow} \mathcal{I}^\ell, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3 \stackrel{\mathbf{R}}{\leftarrow} \mathcal{I}^k$ , sets  $\mathbf{v}_3 = \mathbf{X}_1.\mathbf{v}_1 \oplus \mathbf{X}_2.\mathbf{v}_2$  and then sends the following commitments for  $i = 1, 2, 3$  to the verifier  $\mathbf{V}$ :

$$\begin{aligned} C_{i0} &\leftarrow \text{Com}(\pi'_i = \pi_i, \mathbf{t}_{i0} = \mathbf{A}.\mathbf{(u}_i \parallel \mathbf{v}_i) \oplus \mathbf{f}_i) \\ C_{i1} &\leftarrow \text{Com}(\mathbf{t}_{i1} = \pi_i(\mathbf{f}_i)) \\ C_{i2} &\leftarrow \text{Com}(\mathbf{t}_{i2} = \pi_i(\mathbf{f}_i \oplus \mathbf{e}_i)) \end{aligned}$$

- The verifier draws  $c \stackrel{\mathbf{R}}{\leftarrow} \mathbb{Z}_3$  and sends it to  $\mathbf{P}$ .
- Depending on the value of  $c$ ,  $\mathbf{P}$  opens the following commitments:
  0.  $\mathbf{P}$  opens  $C_{i0}, C_{i1}$  by sending  $\pi'_i, \mathbf{t}_{i0}, \mathbf{t}_{i1}$  and the associated random coins.
  1.  $\mathbf{P}$  opens  $C_{i0}, C_{i2}$  by sending  $\pi'_i, \mathbf{t}_{i0}, \mathbf{t}_{i2}$  and the associated random coins.
  2.  $\mathbf{P}$  opens  $C_{i1}, C_{i2}$  by sending  $\mathbf{t}_{i1}, \mathbf{t}_{i2}$  and the associated random coins.
- The verifier verifies the correctness of the openings received from the prover, and additionally performs the following checks depending on the challenge  $c$ :
  0.  $\mathbf{V}$  accepts, iff  $\pi'_i \in \mathcal{S}_k$ , there exist solutions  $(\mathbf{a}_i, \mathbf{b}_i) \in \mathcal{I}^\ell \times \mathcal{I}^v$  to the equations  $\mathbf{t}_{i0} \oplus \pi_i^{-1}(\mathbf{t}_{i1}) = \mathbf{A}.\mathbf{(a}_i \parallel \mathbf{b}_i)$  and they satisfy  $\mathbf{b}_3 = \mathbf{X}_1.\mathbf{b}_1 \oplus \mathbf{X}_2.\mathbf{b}_2$ .
  1.  $\mathbf{V}$  accepts, iff there exist solutions  $(\mathbf{c}_i, \mathbf{d}_i) \in \mathcal{I}^\ell \times \mathcal{I}^v$  to the equations  $\mathbf{t}_{i0} \oplus \pi_i^{-1}(\mathbf{t}_{i2}) \oplus \mathbf{y}_i = \mathbf{A}.\mathbf{(c}_i \parallel \mathbf{d}_i)$  and they satisfy  $\mathbf{d}_3 = \mathbf{X}_1.\mathbf{d}_1 \oplus \mathbf{X}_2.\mathbf{d}_2$ .
  2.  $\mathbf{V}$  accepts, iff  $\|\mathbf{t}_{i1} \oplus \mathbf{t}_{i2}\|_1 \stackrel{?}{=} w$ .

**Theorem 4.2.** *The above protocol is a  $\Sigma$ -protocol for the following relation:*

$$\mathcal{R}_{LLPN} = \left\{ \left( (\mathbf{A}, \mathbf{X}_1, \mathbf{X}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3), (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \right) : \right. \\ \left. \bigwedge_{i=1}^3 \left( \mathbf{y}_i = \mathbf{A} \cdot (\mathbf{r}_i \parallel \mathbf{m}_i) \oplus \mathbf{e}_i \wedge \|\mathbf{e}_i\|_1 = w \right) \wedge \mathbf{m}_3 = \mathbf{X}_1 \cdot \mathbf{m}_1 \oplus \mathbf{X}_2 \cdot \mathbf{m}_2 \right\}.$$

As we can turn any commitment  $\mathbf{y} = \mathbf{A} \cdot (\mathbf{r} \parallel \mathbf{m}) \oplus \mathbf{e}$  for an (unknown) message  $\mathbf{m}$  into a commitment for the message  $\mathbf{m} \oplus \mathbf{x}$  as  $\mathbf{y} \oplus \mathbf{A} \cdot (0^\ell \parallel \mathbf{x}) = \mathbf{A} \cdot (\mathbf{r} \parallel (\mathbf{m} \oplus \mathbf{x})) \oplus \mathbf{e}$ . Our protocol directly implies a protocol for affine relations

$$\mathcal{R}_{ALPN} = \left\{ \left( (\mathbf{A}, \mathbf{X}_1, \mathbf{X}_2, \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^3), (\{\mathbf{r}_i, \mathbf{m}_i, \mathbf{e}_i\}_{i=1}^3) \right) : \right. \\ \left. \bigwedge_{i=1}^3 \left( \mathbf{y}_i = \mathbf{A} \cdot (\mathbf{r}_i \parallel \mathbf{m}_i) \oplus \mathbf{e}_i \wedge \|\mathbf{e}_i\|_1 = w \right) \right. \\ \left. \wedge (\mathbf{m}_3 \oplus \mathbf{x}_3) = \mathbf{X}_1 \cdot (\mathbf{m}_1 \oplus \mathbf{x}_1) \oplus \mathbf{X}_2 \cdot (\mathbf{m}_2 \oplus \mathbf{x}_2) \right\}.$$

In particular, this allows to prove that  $\mathbf{m}_1 = \mathbf{1}^v \oplus \mathbf{m}_2$ , i.e.,  $\mathbf{m}_1$  is the bitwise negation of  $\mathbf{m}_2$ . Furthermore, the protocol can be seen to directly generalize to relations among more than 3 secret messages as well.

*Proof.* We do not give a full proof here, as it is very similar to that of Theorem 4.1. Besides technicalities, the only difference is to prove that the extracted witnesses indeed satisfy the required linear relation.

This can be seen as follows. From the verification equations of  $c = 0$  and  $c = 1$  we first get that  $\mathbf{y}_i = \mathbf{A} \cdot (\mathbf{a}_i \oplus \mathbf{c}_i \parallel \mathbf{b}_i \oplus \mathbf{d}_i) \oplus \pi_i'^{-1}(\mathbf{t}_{i1} \oplus \mathbf{t}_{i2})$ , where the second addend has low weight by the same arguments as earlier. Using the linear relations among the  $\mathbf{b}_i$  and the  $\mathbf{d}_i$  we further get  $(\mathbf{b}_3 \oplus \mathbf{d}_3) = \mathbf{X}_1 \cdot (\mathbf{b}_1 \oplus \mathbf{d}_1) \oplus \mathbf{X}_2 \cdot (\mathbf{b}_2 \oplus \mathbf{d}_2)$ . Thus, a valid witness is given by  $\mathbf{r}'_i = \mathbf{a}_i \oplus \mathbf{c}_i$ ,  $\mathbf{m}'_i = \mathbf{b}_i \oplus \mathbf{d}_i$  and  $\mathbf{e}'_i = \pi_i'^{-1}(\mathbf{t}_{i1} \oplus \mathbf{t}_{i2})$ .

To see that the protocol is still honest-verifier zero-knowledge it suffices to note that the only additional information the verifier learns is that the random coins used in the protocol and the secret witnesses satisfy the linear relation which is already part of the description of the relation  $\mathcal{R}_{LLPN}$ . The rest of the protocol is just a parallel execution of independent instances of the protocol for  $\mathcal{R}_{LPN}$ .  $\square$

### 4.3 Proving Multiplicative Relations

Finally, we present a protocol which can be used to prove a bitwise relation amongst commitments  $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$  (where  $\mathbf{y}_i = \mathbf{A} \cdot (\mathbf{r}_i \parallel \mathbf{m}_i) \oplus \mathbf{e}_i$ ). That is, it allows one to prove that the messages satisfy  $\mathbf{m}_3 = \mathbf{m}_1 \circ \mathbf{m}_2$ . The main idea of the protocol is to reduce the task of proving this multiplicative relation to a linear one, which we showed how to solve in the last section.

In the protocol, which is given in detail below, the prover P first samples vectors  $\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \tilde{\mathbf{m}}_3 \stackrel{\mathcal{R}}{\leftarrow} \mathcal{I}^{4v}$  such that (1)  $\tilde{\mathbf{m}}_3 = \tilde{\mathbf{m}}_1 \circ \tilde{\mathbf{m}}_2$  and (2) for all  $(a, b) \in \mathcal{I}^2$

the number of indices  $j \in \{1, \dots, 4v\}$  satisfying  $(\tilde{\mathbf{m}}_1[j], \tilde{\mathbf{m}}_2[j]) = (a, b)$  is exactly  $v$ . Further, the prover draws a random matrix  $\mathbf{R} \xleftarrow{\mathcal{R}} \mathcal{I}^{v \times 4v}$  with full rank such that each row has Hamming weight exactly 1 and such that  $\mathbf{R} \cdot \tilde{\mathbf{m}}_i = \mathbf{m}_i$  for  $i = 1, 2, 3$  (so  $\mathbf{R}$  is a  $v \times v$  permutation matrix with  $3v$  additional zero columns).

Now  $\mathbf{P}$  and  $\mathbf{V}$  basically run the protocol from the previous section to prove the linear relation  $\mathbf{R} \cdot \tilde{\mathbf{m}}_i = \mathbf{m}_i$ , with the crucial difference that the relation  $\mathbf{R}$  is not known to  $\mathbf{V}$ , instead the prover additionally sends a commitment to  $\mathbf{R}$  with the first message. Moreover  $\mathbf{P}$  sends commitments to the  $\tilde{\mathbf{m}}_i$ 's to  $\mathbf{V}$ .

The challenge space is extended from  $\mathbb{Z}_3$  to  $\mathbb{Z}_4$  (but will later merge  $c = 2$  and  $c = 3$  and get back to 3). If  $c \in \{0, 1, 2\}$ , the prover opens the commitment to  $\mathbf{R}$  and sends the same answer as he would in the the protocol for proving the linear relation  $\mathbf{R} \cdot \tilde{\mathbf{m}}_i = \mathbf{m}_i$  for the given  $c$ . If  $c = 3$ , the prover opens the commitments to the  $\tilde{\mathbf{m}}_i$ 's, and  $\mathbf{V}$  checks if  $\tilde{\mathbf{m}}_3 \stackrel{?}{=} \tilde{\mathbf{m}}_1 \circ \tilde{\mathbf{m}}_2$ .

The soundness of this protocol follows as  $\mathbf{R} \cdot \tilde{\mathbf{m}}_i = \mathbf{m}_i$  and  $\tilde{\mathbf{m}}_3 = \tilde{\mathbf{m}}_1 \circ \tilde{\mathbf{m}}_2$  together imply the claimed statement  $\mathbf{m}_3 = \mathbf{m}_1 \circ \mathbf{m}_2$ .

The zero knowledge property holds as even though  $\mathbf{R}$  together with the  $\tilde{\mathbf{m}}_i$ 's determines the  $\mathbf{m}_i$ 's, each by itself is completely independent of the  $\mathbf{m}_i$ 's, and we never open both.

Finally, we observe that in our protocol for proving linear relations, the verifier does not need to know the linear relation  $\mathbf{R}$  if  $c = 2$ . So we can collapse the challenges  $c = 2$  and  $c = 3$  as described above, but not open  $\mathbf{R}$  in this case.

Formally, the protocol flow is defined by the following algorithms:

- $\mathbf{P}$  samples  $\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \tilde{\mathbf{m}}_3 \xleftarrow{\mathcal{R}} \mathcal{I}^{4v}$  such that  $\tilde{\mathbf{m}}_3 = \tilde{\mathbf{m}}_1 \circ \tilde{\mathbf{m}}_2$  and such that for all  $(a, b) \in \mathcal{I}^2$  the number of indices  $j \in \{1, \dots, 4v\}$  satisfying  $(\tilde{\mathbf{m}}_1[j], \tilde{\mathbf{m}}_2[j]) = (a, b)$  is exactly  $v$ . Further, the prover draws a random matrix  $\mathbf{R} \xleftarrow{\mathcal{R}} \mathcal{I}^{v \times 4v}$  with full rank such that each row has at most Hamming weight 1 and such that  $\mathbf{R} \cdot \tilde{\mathbf{m}}_i = \mathbf{m}_i$  for  $i = 1, 2, 3$ .

In the following we denote the  $j^{\text{th}}$   $v$ -bit block of  $\tilde{\mathbf{m}}_i$  by  $\tilde{\mathbf{m}}_i^j$ , i.e.,  $\tilde{\mathbf{m}}_i^j = (\tilde{\mathbf{m}}_{iu})_{u=(j-1)v+1}^{jv}$ . Similarly,  $\mathbf{R}^j$  denotes the matrix given by columns  $(j-1)v+1$  to  $jv$  of  $\mathbf{R}$ .

In the remainder of this protocol description all computations are done for  $i = 1, 2, 3$  and  $j = 1, 2, 3, 4$ , respectively.

$\mathbf{P}$  draws  $\tilde{\mathbf{r}}_i^j \xleftarrow{\mathcal{R}} \mathcal{I}^\ell$  and defines auxiliary images as  $\tilde{\mathbf{y}}_i^j = \mathbf{A} \cdot (\tilde{\mathbf{r}}_i^j \parallel \tilde{\mathbf{m}}_i^j) \oplus \tilde{\mathbf{e}}_i^j$  for  $\tilde{\mathbf{e}}_i^j \xleftarrow{\mathcal{R}} \mathcal{I}_w^k$ . It then samples permutations  $\pi_i, \pi_i^j \xleftarrow{\mathcal{R}} \mathcal{S}_k$  at random.

It then draws  $\mathbf{v}_i^j \xleftarrow{\mathcal{R}} \mathcal{I}^v$ ,  $\mathbf{u}_i, \mathbf{u}_i^j \xleftarrow{\mathcal{R}} \mathcal{I}^\ell$ ,  $\mathbf{f}_i, \mathbf{f}_i^j \xleftarrow{\mathcal{R}} \mathcal{I}^k$ , sets  $\mathbf{v}_i = \sum_{j=1}^4 \mathbf{R}^j \cdot \mathbf{v}_i^j$  and then sends the following commitments to the verifier  $\mathbf{V}$ :

$$\begin{aligned}
 \tilde{C} &\leftarrow \text{Com}(\tilde{\mathbf{y}}_1^1 = \tilde{\mathbf{y}}_1^1, \dots, \tilde{\mathbf{y}}_3^4 = \tilde{\mathbf{y}}_3^4) & C_{\mathbf{R}} &\leftarrow \text{Com}(\mathbf{R}' = \mathbf{R}) \\
 C_{i0} &\leftarrow \text{Com}(\pi_i^t = \pi_i, \mathbf{t}_{i0} = \mathbf{A} \cdot (\mathbf{u}_i \parallel \mathbf{v}_i) \oplus \mathbf{f}_i) \\
 C_{i1} &\leftarrow \text{Com}(\mathbf{t}_{i1} = \pi_i(\mathbf{f}_i)) & C_{i2} &\leftarrow \text{Com}(\mathbf{t}_{i2} = \pi_i(\mathbf{f}_i \oplus \mathbf{e}_i)) \\
 C_{i0}^j &\leftarrow \text{Com}(\pi_i^{tj} = \pi_i^j, \mathbf{t}_{i0}^j = \mathbf{A} \cdot (\mathbf{u}_i^j \parallel \mathbf{v}_i^j) \oplus \mathbf{f}_i^j) \\
 C_{i1}^j &\leftarrow \text{Com}(\mathbf{t}_{i1}^j = \pi_i^j(\mathbf{f}_i^j)) & C_{i2}^j &\leftarrow \text{Com}(\mathbf{t}_{i2}^j = \pi_i^j(\mathbf{f}_i^j \oplus \tilde{\mathbf{e}}_i^j))
 \end{aligned}$$

- The verifier draws  $c \xleftarrow{\mathcal{R}} \mathbb{Z}_3$  and sends it to  $\mathbf{P}$ .

- Depending on the value of  $c$ ,  $\mathsf{P}$  opens the following commitments:
  0.  $\mathsf{P}$  opens  $C_{i0}, C_{i1}, C_{i0}^j, C_{i1}^j, C_{\mathbf{R}}$  by sending  $\pi_i', \mathbf{t}_{i0}, \mathbf{t}_{i1}, \pi_i^j, \mathbf{t}_{i0}^j, \mathbf{t}_{i1}^j, \mathbf{R}'$  and the associated random coins.
  1.  $\mathsf{P}$  opens  $C_{i0}, C_{i2}, C_{i0}^j, C_{i2}^j, \tilde{C}, C_{\mathbf{R}}$  by sending  $\pi_i', \mathbf{t}_{i0}, \mathbf{t}_{i2}, \pi_i^j, \mathbf{t}_{i0}^j, \mathbf{t}_{i2}^j, \tilde{\mathbf{y}}_i^j, \mathbf{R}'$  and the associated random coins.
  2.  $\mathsf{P}$  opens  $C_{i1}, C_{i2}, C_{i1}^j, C_{i2}^j, \tilde{C}$  by sending  $\mathbf{t}_{i1}, \mathbf{t}_{i2}, \mathbf{t}_{i1}^j, \mathbf{t}_{i2}^j, \tilde{\mathbf{y}}_i^j$  and the associated random coins.
- The verifier verifies the correctness of the openings received from the prover, and additionally performs the following checks depending on the challenge  $c$ :
  0.  $\mathsf{V}$  accepts, iff  $\pi_i', \pi_i^j \stackrel{?}{\in} \mathcal{S}_k$ , there exist solutions  $(\mathbf{a}_i, \mathbf{b}_i), (\mathbf{a}_i^j, \mathbf{b}_i^j) \in \mathcal{I}^\ell \times \mathcal{I}^v$  to the equations  $\mathbf{t}_{i0} \oplus \pi_i'^{-1}(\mathbf{t}_{i1}) = \mathbf{A} \cdot (\mathbf{a}_i \parallel \mathbf{b}_i)$  and  $\mathbf{t}_{i0}^j \oplus (\pi_i^j)^{-1}(\mathbf{t}_{i1}^j) = \mathbf{A} \cdot (\mathbf{a}_i^j \parallel \mathbf{b}_i^j)$ , respectively, which satisfy  $\mathbf{b}_i = \sum_{j=1}^4 \mathbf{R}^j \cdot \mathbf{b}_i^j$ .
  1.  $\mathsf{V}$  accepts, iff  $\mathbf{R}'$  has full rank and each row has Hamming weight at most 1, and iff there exist solutions  $(\mathbf{a}_i, \mathbf{b}_i), (\mathbf{a}_i^j, \mathbf{b}_i^j) \in \mathcal{I}^\ell \times \mathcal{I}^v$  to the equations  $\mathbf{t}_{i0} \oplus \pi_i'^{-1}(\mathbf{t}_{i2}) \oplus \mathbf{y}_i = \mathbf{A} \cdot (\mathbf{a}_i \parallel \mathbf{b}_i)$  and  $\mathbf{t}_{i0}^j \oplus (\pi_i^j)^{-1}(\mathbf{t}_{i2}^j) \oplus \tilde{\mathbf{y}}_i^j = \mathbf{A} \cdot (\mathbf{a}_i^j \parallel \mathbf{b}_i^j)$ , respectively, which satisfy  $\mathbf{b}_i = \sum_{j=1}^4 \mathbf{R}^j \cdot \mathbf{b}_i^j$ .
  2.  $\mathsf{V}$  accepts, iff  $\|\mathbf{t}_{i1} \oplus \mathbf{t}_{i2}\|_1 = \|\mathbf{t}_{i1}^j \oplus \mathbf{t}_{i2}^j\|_1 \stackrel{?}{=} w$ ,  $\tilde{\mathbf{y}}_i^j \stackrel{?}{=} \mathbf{A} \cdot (\tilde{\mathbf{r}}_i^j \parallel \tilde{\mathbf{m}}_i^j) \oplus \tilde{\mathbf{e}}_i^j$ ,  $\|\tilde{\mathbf{e}}_i^j\|_1 \stackrel{?}{=} w$  and  $\tilde{\mathbf{m}}_1^j \circ \tilde{\mathbf{m}}_2^j \stackrel{?}{=} \tilde{\mathbf{m}}_3^j$ .

**Theorem 4.3.** *The above protocol is a  $\Sigma$ -protocol for the following relation:*

$$\mathcal{R}_{MLPN} = \left\{ ((\mathbf{A}, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3), (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)) : \bigwedge_{i=1}^3 (\mathbf{y}_i = \mathbf{A} \cdot (\mathbf{r}_i \parallel \mathbf{m}_i) \oplus \mathbf{e}_i \wedge \|\mathbf{e}_i\|_1 = w) \wedge \mathbf{m}_3 = \mathbf{m}_1 \circ \mathbf{m}_2 \right\}.$$

*Proof.* The 3-move form of the protocol is easy to see. Furthermore, completeness directly follows from the construction and can easily be verified.

*Special soundness.* Concerning the special soundness of the protocol, note the following. Using the same arguments as in the proof of Theorem 4.1, we can extract openings  $\mathbf{m}'_i, \mathbf{r}'_i$  and  $\mathbf{e}'_i$  of the  $\mathbf{y}_i$ , and similarly, we get  $\tilde{\mathbf{m}}_i^j, \tilde{\mathbf{r}}_i^j$  and  $\tilde{\mathbf{e}}_i^j$  which are valid openings for the  $\tilde{\mathbf{y}}_i^j$ . Now, by the same arguments as in Theorem 4.2 we can further infer that  $\mathbf{m}'_i = \sum_{j=1}^4 \mathbf{R}^j \cdot \tilde{\mathbf{m}}_i^j = \mathbf{R}' \cdot \tilde{\mathbf{m}}_i'$ . Furthermore, we know that  $\tilde{\mathbf{m}}_1' \circ \tilde{\mathbf{m}}_2' = \tilde{\mathbf{m}}_3'$ . Now, because of the special form of  $\mathbf{R}'$ , we can finally infer that the same relation must also be true for the  $\mathbf{m}'_i$ .

*Honest-verifier zero-knowledge.* We do not give a full simulator here, but only give the intuition why the protocol is zero-knowledge. Clearly, the  $\tilde{\mathbf{m}}_i^j$  are uniformly random in their domain, and do not leak any information about the  $\mathbf{m}_i$ , as long as the matrix  $\mathbf{R}$  is kept secret. Similarly, if the  $\tilde{\mathbf{m}}_i^j$  are kept secret, the matrix  $\mathbf{R}$  itself is a uniformly random matrix of full rank with the specified restriction on the weight of its rows. Computationally this still holds true even if the  $\tilde{\mathbf{y}}_i^j$  are revealed, as they are pseudorandom by Theorem 3.1. The zero-knowledge property now follows from that of the protocol for  $\mathcal{R}_{LLPN}$ .  $\square$

#### 4.4 Proving Arbitrary Relations

We finally briefly explain how one can use the protocols presented in this section to prove that committed values  $\mathbf{m}_0, \mathbf{m}_1$  satisfy  $\mathbf{m}_0 = C(\mathbf{m}_1)$  for an arbitrary circuit  $C$ . Let  $C_1, \dots, C_d$  denote the layers of  $C$ , i.e.,  $C(\mathbf{m}_1) = C_d(\dots C_1(\mathbf{m}_1)\dots)$ , where we assume that each  $C_i$  is either a linear function or a bitwise operation (e.g., bitwise NAND). For simplicity we assume the number of input and output wires to each  $C_i$  is  $\ell$ , where  $\ell$  is the length of the underlying LPN problem.

We use our string commitment scheme to commit to the values in the intermediate layers, i.e., to strings  $\mathbf{x}_1, \dots, \mathbf{x}_d$  where  $\mathbf{x}_1 = \mathbf{m}_1, \mathbf{x}_2 = C_1(\mathbf{m}_1), \dots, \mathbf{x}_d = C(\mathbf{m}_1)$  (note that we already have commitments to  $\mathbf{x}_1 = \mathbf{m}_1$  and  $\mathbf{x}_d = \mathbf{m}_0$ ). Now we use our  $\Sigma$ -protocols to prove that  $\mathbf{x}_{i+1} = C_i(\mathbf{x}_i)$  for  $i = 1 \dots d - 1$ .

The total communication complexity of this protocol is  $\Theta(\sum |C_i| \ell \log \ell) = \Theta(|C| \ell \log \ell)$ , the soundness error is  $2/3$ , and thus for most applications must be lowered by (parallel) repetition.

### 5 Conclusions and Open Problems

We presented a very simple and efficient string commitment scheme, whose security is based on the hardness of the LPN-problem, or, equivalently, on the hardness of decoding random linear codes. We further presented  $\Sigma$ -protocols which allow one to prove arbitrary relations among secret values  $\mathbf{m}_i$ , i.e.,  $\mathbf{m}_0 = C(\mathbf{m}_1, \dots, \mathbf{m}_u)$  for any circuit  $C$ . The size of a proof is only quasi-linear in the length of the committed messages.

We introduced an “exact” version of the LPN-problem which is polynomially equivalent to the standard LPN problem. This new assumption might be of independent interest as basing existing LPN based schemes on this new assumptions removes the completeness error (cf. §2 for a discussion).

It would be interesting to find protocols which already achieve a small knowledge error in only run, and do not rely on repetitions. Furthermore, a tighter reduction for the hardness of the decisional xLPN problem, in particular not relying on the Goldreich-Levin theorem, would be desirable.

**Acknowledgment.** We are grateful to Petros Mol for helpful discussions on the reduction for the hardness of the xLPN problem.

### References

1. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
2. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography with Constant Input Locality. *Journal of Cryptology* 22(4), 429–469 (2009)
3. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (2012)



4. Bellare, M., Goldreich, O.: On Defining Proofs of Knowledge. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 390–420. Springer, Heidelberg (1993)
5. Berlekamp, E., McEliece, R., van Tilborg, H.: On the Inherent Intractability of Certain Coding Problems. *IEEE Transactions on Information Theory* 24(3), 384–386 (1978)
6. Blum, A., Furst, M.L., Kearns, M., Lipton, R.J.: Cryptographic Primitives Based on Hard Learning Problems. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg (1994)
7. Blum, A., Kalai, A., Wasserman, H.: Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. *Journal of the ACM* 50(4), 506–519 (2003)
8. Boyar, J., Damgård, I., Peralta, R.: Short Non-Interactive Cryptographic Proofs. *Journal of Cryptology* 13(4), 449–472 (2000)
9. Cayrel, P.-L., Lindner, R., Rückert, M., Silva, R.: Improved Zero-Knowledge Identification with Lattices. In: Heng, S.-H., Kurosawa, K. (eds.) ProvSec 2010. LNCS, vol. 6402, pp. 1–17. Springer, Heidelberg (2010)
10. Cayrel, P.-L., Véron, P., El Yousfi Alaoui, S.M.: A Zero-Knowledge Identification Scheme Based on the  $q$ -ary Syndrome Decoding Problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 171–186. Springer, Heidelberg (2011)
11. Chen, H., Cramer, R.: Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computations over Small Fields. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 521–536. Springer, Heidelberg (2006)
12. Cramer, R.: Modular Design of Secure yet Practical Cryptographic Protocols. PhD thesis, CWI and University of Amsterdam (1997)
13. Cramer, R., Damgård, I.: Linear Zero-Knowledge - A Note on Efficient Zero-Knowledge Proofs and Arguments. In: Leighton, F.T., Shor, P.W. (eds.) STOC 1997, pp. 436–445. ACM (1997)
14. Cramer, R., Damgård, I.: Zero-Knowledge Proofs for Finite Field Arithmetic or: Can Zero-Knowledge Be for Free? In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 424–441. Springer, Heidelberg (1998)
15. Cramer, R., Damgård, I.: On the Amortized Complexity of Zero-Knowledge Protocols. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 177–191. Springer, Heidelberg (2009)
16. Damgård, I.: On  $\Sigma$ -Protocols. Lecture on Cryptologic Protocol Theory; Faculty of Science. University of Aarhus (2004)
17. Damgård, I., Goldreich, O., Okamoto, T., Wigderson, A.: Honest Verifier vs Dishonest Verifier in Public Coin Zero-Knowledge Proofs. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 325–338. Springer, Heidelberg (1995)
18. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: How to Encrypt with the LPN Problem. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfssdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 679–690. Springer, Heidelberg (2008)
19. Goldreich, O., Micali, S., Wigderson, A.: How to Play Any Mental Game, or A Completeness Theorem for Protocols with Honest Majority. In: Aho, A.V. (ed.) STOC 1987, pp. 218–229. ACM (1987)
20. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
21. Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)

22. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Johnson, D.S., Feige, U. (eds.) STOC 2007, pp. 21–30. ACM (2007)
23. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and Efficient Zero-Knowledge Proofs from Hard Learning Problems. Cryptology ePrint Archive, Report 2012/513 (2012), <http://eprint.iacr.org/>
24. Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
25. Kalai, Y.T., Raz, R.: Succinct Non-Interactive Zero-Knowledge Proofs with Pre-processing for LOGSNP. In: FOCS 2006, pp. 355–366. IEEE Computer Society (2006)
26. Katz, J., Shin, J.S., Smith, A.: Parallel and Concurrent Security of the HB and HB<sup>+</sup> Protocols. *Journal of Cryptology* 23(3), 402–421 (2010)
27. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 372–389. Springer, Heidelberg (2008)
28. Kearns, M.J.: Efficient Noise-Tolerant Learning from Statistical Queries. *Journal of the ACM* 45(6), 983–1006 (1998)
29. Kilian, J.: A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In: STOC 1992, pp. 723–732 (1992)
30. Kilian, J., Micali, S., Ostrovsky, R.: Minimum Resource Zero-Knowledge Proofs (Extended Abstract). In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 545–546. Springer, Heidelberg (1990)
31. Kilian, J., Petrank, E.: An Efficient Noninteractive Zero-Knowledge Proof System for NP with General Assumptions. *Journal of Cryptology* 11(1), 1–27 (1998)
32. Kiltz, E., Pietrzak, K., Cash, D., Jain, A., Venturi, D.: Efficient Authentication from Hard Learning Problems. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 7–26. Springer, Heidelberg (2011)
33. Micali, S.: Computationally Sound Proofs. *SIAM Journal on Computing* 30(4), 1253–1298 (2000)
34. Micciancio, D., Mol, P.: Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
35. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: STOC 2005, pp. 84–93. ACM (2005)
36. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)
37. Véron, P.: Improved Identification Schemes Based on Error-Correcting Codes. *Applicable Algebra in Engineering, Communication and Computing* 8(1), 57–69 (1996)