Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise*

Abhishek Jain[1], Stephan Krenn[2], Krzysztof Pietrzak[2], Aris Tentes[3]

[1] Massachusetts Institute of Technology, USA, and
Boston University, USA
abhishek@csail.mit.edu
[2] Institute of Science and Technology Austria
{stephan.krenn,pietrzak}@ist.ac.at
[3] 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 Σ-protocol), but also for such proofs showing any kind of relation amongst committed values, i.e., proving that messages $\boldsymbol{m}_0, \ldots, \boldsymbol{m}_u$, are such that $\boldsymbol{m}_0 = C(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_u)$ 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 ℓ, 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 $\boldsymbol{m}$ by publishing a commitment σ, 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

* 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 $\boldsymbol{m}$ from the commitment σ, binding means that one cannot open a commitment σ to two different messages $\boldsymbol{m} \neq \boldsymbol{m}'$.

In our scheme, the commitment to a message $\boldsymbol{m}$ is simply the encoding of $\boldsymbol{m}$ 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 $\mathbf{A}.\boldsymbol{s} \oplus \boldsymbol{e}$ from uniformly random. Here $\mathbf{A}$ is a "skinny" public random binary $k \times \ell$ matrix, $\boldsymbol{s}$ is a uniformly random ℓ bit secret and $\boldsymbol{e}$ is a random vector of low weight (the exact distribution of $\boldsymbol{e}$ is discussed in §2.2). The LPN problem has found numerous applications as the assumption underlying provably secure cryptosystems, like symmetric encryption [ACPS09,GRS08] or secret-key [HB01,JW05,KSS10,KPC+11] and public-key [Ste93] authentication schemes.

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 [BMvT78,BFKL93,BKW03,Kea98,Reg05]. 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 $\mathbf{A}$) [BKW03]. If $k = \Theta(\ell)$ is linear in ℓ, 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 $\boldsymbol{m} \in \mathcal{I}^v$ (where $\mathcal{I} \stackrel{\mathsf{def}}{=} \{0,1\}$) is simply

$$\mathtt{Com}(\boldsymbol{m}) = \mathbf{A}.(\boldsymbol{r} \| \boldsymbol{m}) \oplus \boldsymbol{e},$$

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

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 $\mathbf{A}'.\boldsymbol{r} \oplus \boldsymbol{e}$ is pseudorandom.

Zero-knowledge protocols for arbitrary circuits. We construct a zero-knowledge proof of knowledge, which is basically a so called Σ-protocol, that allows to prove knowledge of the message $\boldsymbol{m}$ hidden inside a commitment without revealing anything about it. Furthermore, we give a protocol for proving that committed messages $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2$ satisfy a linear relation $\boldsymbol{m}_0 = \mathbf{X}_1.\boldsymbol{m}_1 \oplus \mathbf{X}_2.\boldsymbol{m}_2$ (for any square matrices $\mathbf{X}_1, \mathbf{X}_2$). Based on this protocol, we construct proofs for any bitwise relations $\boldsymbol{m}_0 = \boldsymbol{m}_1 \circ \boldsymbol{m}_2$, where $\circ$ can be any bitwise relation like AND, NAND, OR, NOR. As NAND is functionally complete, we can prove relations $\boldsymbol{m}_0 = C(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_t)$ 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 $\boldsymbol{m}_0 = w$ and $\boldsymbol{m}_1 = 1$ and prove that the committed values satisfy the relation $C_x(\boldsymbol{m}_0) = \boldsymbol{m}_1$ where $C_x(.)$ is the circuit computing the $\mathcal{NP}$ relation $\mathcal{R}(x, .)$. 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 [Ste93] 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 [BMvT78]. Subsequent to Stern's work, Véron [Vér96] proposed a Σ-protocol for proving knowledge of an LPN secret. However, as we will show in Appendix A, there is a gap in the proof of the zero-knowledge 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 [CLRS10,CVA10,KTX08] and references therein). In particular, Cayrel et al. [CVA10] 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. [AJLA+12] constructed Σ-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.

During the last decades, a large body of work on efficient interactive and non-interactive zero-knowledge proofs and arguments of knowledge has been published, see, e.g., [BDP00,CD97,CD98,CD09,GS08,IKOS07,KR06,KMO90,KP98] 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 [Kil92] or the non-interactive argument in the random-oracle model [Mic00]). These schemes rely on probabilistically checkable proofs (PCP), and are not really practical.

The beautiful work of Ishai et al. [IKOS07] 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 [GMW87] 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 $\mathcal{R}(x, .)$, 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 [CC06] they even get an asymptotic communication complexity of $O(s) + poly(t, \log s)$, 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 [CD09] 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 $\boldsymbol{m}_0 = \mathrm{AES}(\boldsymbol{m}_1, \boldsymbol{m}_2)$, i.e., $\boldsymbol{m}_0$ is the output of the AES block-cipher under key $\boldsymbol{m}_1$ on input $\boldsymbol{m}_2$.

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. A flaw in an earlier zero-knowledge proof for the LPN-problem is shown Appendix A.

2 Preliminaries

We use bold lower-case and upper-case letters like $\boldsymbol{a}, \mathbf{A}$ to denote vectors and matrices, respectively. Probabilistic polynomial time (PPT) algorithms are written by sans-serif letters like A. Calligraphic letters like $\mathcal{A}$ always denote sets. We write $a \overset{R}{\leftarrow} \mathcal{A}$ if a was drawn uniformly at random from set $\mathcal{A}$, $a \overset{R}{\leftarrow} \chi$ if a was drawn according to some probability distribution χ, and $a \overset{R}{\leftarrow} \mathsf{A}$ if a is the output of a randomized algorithm 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 $\boldsymbol{a} \in \mathcal{I}^k$ is denoted by $\|\boldsymbol{a}\|_1 = \sum_{i=1}^{k} \boldsymbol{a}[i]$. With $\mathcal{I}_w^k = \{\boldsymbol{a} \in \mathcal{I}^k \ : \ \|\boldsymbol{a}\|_1 = w\}$ 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 $\boldsymbol{a}$ and $\boldsymbol{b}$ is written as $\boldsymbol{a}\|\boldsymbol{b}$. 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 $\boldsymbol{a} \in \mathcal{I}^k$, $\pi(\boldsymbol{b})$ denotes the string $\boldsymbol{a}[i] = \boldsymbol{b}[\pi(i)]$.

2.1 Commitment Schemes

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

- *On input 1^ℓ, the key generation algorithm* KGen *outputs a public commitment key pk.*
- *The commitment algorithm* 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* 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*: Ver evaluates to 1 whenever the inputs were computed by an honest party, i.e.,

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

- *Perfect binding*: With overwhelming probability over the choice of the public key $pk \stackrel{R}{\leftarrow} \mathsf{KGen}(1^\ell)$, no commitment c can be opened in two different ways, i.e.,
$$(\mathsf{Ver}(pk, m, c, d) = 1) \wedge (\mathsf{Ver}(pk, m', c, d') = 1) \Rightarrow m = m'$$
- *Computational hiding*: A commitment c computationally hides the committed message: with overwhelming probability over the choice of $pk \stackrel{R}{\leftarrow} \mathsf{KGen}(1^\ell)$, for every $m, m' \in \mathcal{M}$ and $(c, d) \stackrel{R}{\leftarrow} \mathsf{Com}(m, pk)$, $(c', d') \stackrel{R}{\leftarrow} \mathsf{Com}(m', pk)$ 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 χ be an error distribution over $\mathcal{I}^k$. The* decisional (χ, ℓ, k)-LPN problem *is (s, ϵ)-hard if for every distinguisher* D *of size s:*

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

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

$$\left| \Pr_{\boldsymbol{x}, \mathbf{A}, \boldsymbol{e}}[\mathsf{D}(\mathbf{A}, \mathbf{A}.\boldsymbol{x} \oplus \boldsymbol{e}) = \boldsymbol{x} \right| \le \epsilon$$

In the standard definition of the LPN problem, the error distribution χ is the Bernoulli distribution with some parameter $0 < \tau < \frac{1}{2}$, i.e., every bit $\boldsymbol{e}[i]$ is chosen independently and identically distributed with $\Pr[\boldsymbol{e}[i] = 1] = \tau$, we will refer to this version as LPN_τ. As mentioned in the introduction, for $k = \Theta(\ell)$ as used in this paper, the search version of LPN_τ is the same as the problem of decoding random linear codes, and is believed to be exponentially hard. The search and decision version of LPN_τ are known to be equivalent [BFKL93,KSS10], 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 [AIK09, Lemma 4.4]. (cf. [MM11] 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 xLPN for short. Similar to LPN_τ, xLPN is parameterized by some noise parameter $0 < \tau < \frac{1}{2}$, and the (search or decision) xLPN_τ problem is defined exactly like LPN_τ, except that the Hamming weight of the error vector is exactly $\lfloor k\tau \rceil$ (not of expected weight $k\tau$ as in LPN_τ). That is, $\boldsymbol{e}$ is sampled uniformly at random from the set $\mathcal{I}^k_{\lfloor k\tau \rceil}$.

In this work, we assume the hardness of *decisional* xLPN.[4] It is not hard to see that *search* xLPN_τ is hard iff search LPN_τ is hard.[5] Showing equivalence of *decisional* xLPN_τ and LPN_τ version is more tricky. The classical search to decision reduction for LPN_τ from [BFKL93,KSS10] does not work for xLPN_τ, but the sample preserving reduction [AIK09, Lemma 4.4] does. Summing up, we have

Proposition 2.3. *The hardness of decisional* xLPN_τ *(used in this paper) is polynomially related to the hardness of search* LPN_τ.

The sample preserving reduction [AIK09, 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 xLPN_τ than against LPN_τ, if one insists on basing the security of our schemes on the standard LPN_τ assumption in a provable manner, one must take the loss in the reduction into account, which would result in rather large parameters. In Appendix B we show a protocol for proving knowledge of committed values whose security relies directly on the standard decisional LPN_τ assumption. The protocol given there can be extended to prove arbitrary relations amongst committed values, in the same manner as in the case of xLPN_τ 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 [KSS10, Section 5], replacing the LPN assumption with an assumption where we have a fixed upped bound on the weight of the error vector (like it is the case in xLPN) 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 xLPN-problem is of interest beyond the realm of this work.

2.3 Zero-Knowledge Proofs of Knowledge and Σ-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

[4] The security of the basic commitment scheme can be based on decisional LPN, but our Σ-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.

[5] Any D who outputs $\boldsymbol{x}$ with advantage ϵ for xLPN_τ, will output the secret $\boldsymbol{x}$ with advantage at least $\epsilon/\sqrt{k}$ of LPN_τ, as the error vector sampled in LPN_τ has weight $\lfloor k\tau \rceil$ with probability $\geq 1/\sqrt{k}$, and conditioned on this being the case, the error distribution is exactly the same as in xLPN_τ.

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 κ (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 [BG93].

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

Definition 2.4 (Σ-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* Σ-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* V *draws a challenge* $c \xleftarrow{R} \mathcal{C}$ *and sends it to* 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* V *accepts whenever* $(y, w) \in \mathcal{R}$.
- **Special soundness:** *There exists a PPT algorithm* 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* 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 Σ-protocol is also a proof of knowledge for the same relation [Dam04]. However, while in Σ-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 Σ-protocols, e.g., Damgård et al. [DGOW95].

We note that our definition of Σ-protocols slightly differs from the standard definition found in the literature [Cra97,Dam04]. 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 Σ-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 $w = \lfloor \tau k \rceil$. The algorithms of the commitment scheme are then given as follows:

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

Theorem 3.1. *Let $0 < \tau < 0.25$, and $\ell, k, v \in \mathbb{N}$ be such that the decisional xLPN_τ problem (with secrets of length ℓ 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., $\|\mathbf{A}.\boldsymbol{x}\|_1 > 2w$ for all $\boldsymbol{x} \in \mathcal{I}^{\ell+v}$. 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 $\boldsymbol{m}_i, \boldsymbol{r}_i, i = 1, 2$ are two different openings for a commitment $\boldsymbol{c}$. That is, we have that $\boldsymbol{e}_i = \boldsymbol{c} \oplus \mathbf{A}.(\boldsymbol{r}_i \| \boldsymbol{m}_i)$ has norm at most w for $i = 1, 2$. Thus we have that $\boldsymbol{e}_1 \oplus \boldsymbol{e}_2 = \mathbf{A}.(\boldsymbol{r}_1 \| \boldsymbol{m}_1 \oplus \boldsymbol{r}_2 \| \boldsymbol{m}_2)$ is a codeword of length $\|\boldsymbol{e}_1 \oplus \boldsymbol{e}_2\|_1 \leq \|\boldsymbol{e}_1\|_1 + \|\boldsymbol{e}_2\|_1 \leq 2w$, contradicting our assumption on the distance of the code generated by $\mathbf{A}$.

Computational hiding. We have that $\boldsymbol{c} = \mathbf{A}'.\boldsymbol{r} \oplus \boldsymbol{e} \oplus \mathbf{A}''.\boldsymbol{m}$. By the xLPN_τ-assumption $\mathbf{A}'.\boldsymbol{r} \oplus \boldsymbol{e}$, and thus also $\boldsymbol{c}$, is pseudorandom. ☐

4 Zero-Knowledge Proofs of Knowledge

In this section we first construct a Σ-protocol, which on common input $\mathbf{A}$ and $\boldsymbol{y}$ allows the prover to prove knowledge of a valid opening of $\boldsymbol{y}$ under the commitment scheme presented in Section 3. The protocol borrows some basic ideas from Stern [Ste93], who gave a Σ-protocol for the syndrome decoding problem.

After presenting this basic protocol, we give two further Σ-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 Σ-protocols for any relation amongst committed messages.

4.1 Proving Knowledge of a Valid Opening

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

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

If in this protocol $\boldsymbol{f}$ is sampled such that it has low weight, then $\boldsymbol{e} \oplus \boldsymbol{f}$ leaks information about $\boldsymbol{e}$, and the protocol is not zero-knowledge. On the other hand, if $\boldsymbol{f}$ is uniformly random (so $\boldsymbol{e} \oplus \boldsymbol{f}$ is independent of $\boldsymbol{e}$), the protocol is not sound (informally, all we can say is that from answers to both challenges we can extract $\boldsymbol{s}', \boldsymbol{e}'$ where $\boldsymbol{y} = \mathbf{A}.\boldsymbol{s}' \oplus \boldsymbol{e}'$, but $\boldsymbol{e}'$ can have arbitrary weight, and finding such a solution is trivial). In our protocol $\boldsymbol{f}$ is chosen uniformly at random, and to ensure soundness we use a trick from Stern [Ste93]. We additionally commit to a random permutation $\pi \in \mathcal{S}_k$ and to $\pi(\boldsymbol{f})$, $\pi(\boldsymbol{f} \oplus \boldsymbol{e})$. On challenge $c = 0$ and $c = 1$ we now additionally make sure the openings are consistent with the committed errors by opening π and either $\pi(\boldsymbol{f})$ (if $c = 0$) or $\pi(\boldsymbol{f} \oplus \boldsymbol{e})$ (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 $\pi(\boldsymbol{f}) \oplus \pi(\boldsymbol{f} \oplus \boldsymbol{e}) = \pi(\boldsymbol{e})$ (and thus $\boldsymbol{e}$) is small, this will ensure soundness, as from valid answers to all three challenges we can extract $\boldsymbol{s}', \boldsymbol{e}'$ where $\boldsymbol{y} = \mathbf{A}.\boldsymbol{s}' \oplus \boldsymbol{e}'$ and $\boldsymbol{e}'$ has low weight. Opening the commitments to $\pi(\boldsymbol{f}), \pi(\boldsymbol{f} \oplus \boldsymbol{e})$ on $c = 2$ does not hurt the ZK property, as $\pi(\boldsymbol{f}), \pi(\boldsymbol{f} \oplus \boldsymbol{e})$ contains no information about $\boldsymbol{e}$ except its weight.

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

– P samples a permutation $\pi \xleftarrow{\text{R}} \mathcal{S}_k$ at random.
  It then draws $\boldsymbol{v} \xleftarrow{\text{R}} \mathcal{I}^{\ell+v}$, $\boldsymbol{f} \xleftarrow{\text{R}} \mathcal{I}^k$, and then sends the following commitments to the verifier V:
$$C_0 \leftarrow \texttt{Com}(\pi' = \pi, \boldsymbol{t}_0 = \mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{f})$$
$$C_1 \leftarrow \texttt{Com}(\boldsymbol{t}_1 = \pi(\boldsymbol{f}))$$
$$C_2 \leftarrow \texttt{Com}(\boldsymbol{t}_2 = \pi(\boldsymbol{f} \oplus \boldsymbol{e}))$$
– The verifier draws $c \xleftarrow{\text{R}} \mathbb{Z}_3$ 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 $\pi', \boldsymbol{t}_0, \boldsymbol{t}_1$ and the associated random coins.
  1. P opens C_0, C_2 by sending $\pi', \boldsymbol{t}_0, \boldsymbol{t}_2$ and the associated random coins.
  2. P opens C_1, C_2 by sending $\boldsymbol{t}_1, \boldsymbol{t}_2$ 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 $\boldsymbol{t}_0 \oplus \pi'^{-1}(\boldsymbol{t}_1) \overset{?}{\in} \operatorname{img} \mathbf{A}$ and $\pi' \overset{?}{\in} \mathcal{S}_k$.

1. V accepts, iff $\boldsymbol{t}_0 \oplus \pi'^{-1}(\boldsymbol{t}_2) \oplus \boldsymbol{y} \overset{?}{\in} \operatorname{img} \mathbf{A}$.
2. V accepts, iff $\|\boldsymbol{t}_1 \oplus \boldsymbol{t}_2\|_1 = w$.

Theorem 4.1. *The above protocol is a Σ-protocol for the following relation:*

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

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

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 $\texttt{Com}(.)$, 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 $\pi'^{-1}(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2) \oplus \boldsymbol{y} \in \operatorname{img} \mathbf{A}$ and thus that $\boldsymbol{y} = \mathbf{A}.\boldsymbol{s}' \oplus \pi'^{-1}(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2)$, where $\boldsymbol{s}' = (\boldsymbol{r}'\|\boldsymbol{m}')$ is easy to compute. Now, using that $\|\boldsymbol{t}_1 \oplus \boldsymbol{t}_2\|_1 = w$, we have a valid witness of $(\mathbf{A}, \boldsymbol{y})$ is thus given by $(\boldsymbol{r}', \boldsymbol{m}', \pi'^{-1}(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2))$.

Honest-Verifier Zero-Knowledge. In the following we describe an efficient simulator 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 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 $C_0, C_1, \pi', \boldsymbol{t}_0, \boldsymbol{t}_1$ is identical to that in real protocol transcripts. Furthermore, by the computational hiding property of the commitment scheme $\texttt{Com}(.)$, the distribution of C_2 is computationally indistinguishable from that in real protocol runs.
1. For $c = 1$, the simulator draws $\pi \overset{R}{\leftarrow} \mathcal{S}_k$, $\boldsymbol{a} \overset{R}{\leftarrow} \mathcal{I}^k$ and $\boldsymbol{b} \overset{R}{\leftarrow} \mathcal{I}^{\ell+v}$. It sets $C_0 = \texttt{Com}(\pi, \mathbf{A}.\boldsymbol{b} \oplus \boldsymbol{y} \oplus \boldsymbol{a})$ and $C_2 = \texttt{Com}(\pi(\boldsymbol{a}))$. The value of C_1 is computed as commitments to 0. It easy to see that the openings of C_0, C_2 pass the verification equations. To see the correctness of their distributions note that $\boldsymbol{t}_2$ in the real protocol run and $\pi(\boldsymbol{a})$ 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 $\boldsymbol{t}_0 = \mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{f}$, where $\boldsymbol{v}$ is uniformly at random, and $\boldsymbol{f} = \pi^{-1}(\boldsymbol{t}_2 \oplus \boldsymbol{e})$; in the simulated transcript the content of C_0 is given by $\mathbf{A}.(\boldsymbol{b} \oplus \boldsymbol{s}) \oplus (\boldsymbol{a} \oplus \boldsymbol{e})$. Now, $\boldsymbol{v}$ and $\boldsymbol{b} \oplus \boldsymbol{s}$ are both uniformly random, and the terms $\boldsymbol{f}$ and $\boldsymbol{a} \oplus \boldsymbol{e}$ 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 $\mathtt{Com}(.)$.

2. Finally, for $c = 2$, the simulator draws $\boldsymbol{a} \stackrel{\mathrm{R}}{\leftarrow} \mathcal{I}^k$ and $\boldsymbol{b} \leftarrow \mathcal{I}^k_w$ uniformly at random. It computed C_0 as a commitment to 0, $C_1 = \mathtt{Com}(\boldsymbol{a})$ and $C_5 = \mathtt{Com}(\boldsymbol{a} \oplus \boldsymbol{b})$. As before, the distributions of C_0 is computationally indistinguishable from real protocol runs by the binding property of $\mathtt{Com}(.)$, and C_1 and C_2 as well as their openings can easily be seen to perfectly simulate the behavior of an honest prover. □

4.2 Proving Linear Relations

We next describe a Σ-protocol which allows to prove that the messages hidden within commitments $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_2$ (where $\boldsymbol{y}_i = \mathbf{A}.(\boldsymbol{r}_i \| \boldsymbol{m}_i) \oplus \boldsymbol{e}_i$) satisfy arbitrary linear relations. That is, $\mathbf{X}_1.\boldsymbol{m}_1 \oplus \mathbf{X}_2.\boldsymbol{m}_2 = \boldsymbol{m}_3$ 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. P and V run the protocol from the previous section to prove knowledge of $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3$ 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: P commits to three random messages $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$, and later opens the $\boldsymbol{v}_i$'s (if $c = 0$) or $\boldsymbol{v}_i \oplus \boldsymbol{m}_i$ (if $c = 1$). We change this protocol now a bit, and instead choosing $\boldsymbol{v}_3$ at random we compute it as $\boldsymbol{v}_3 = \mathbf{X}_1.\boldsymbol{v}_1 \oplus \mathbf{X}_2.\boldsymbol{v}_2$. Moreover the verifier now additionally checks if $\boldsymbol{v}_3 \stackrel{?}{=} \mathbf{X}_1.\boldsymbol{v}_1 \oplus \mathbf{X}_2.\boldsymbol{v}_2$ (if $c = 0$) and if $(\boldsymbol{v}_3 \oplus \boldsymbol{m}_3) \stackrel{?}{=} \mathbf{X}_1.(\boldsymbol{v}_1 \oplus \boldsymbol{m}_1) \oplus \mathbf{X}_2.(\boldsymbol{v}_2 \oplus \boldsymbol{m}_2)$ (if $c = 1$). With these changes, we get a stronger soundness property: not only can we extract the committed messages $\boldsymbol{m}_i$ from accepting answers to both challenges, but they will also satisfy $\boldsymbol{m}_3 = \mathbf{X}_1.\boldsymbol{m}_1 \oplus \mathbf{X}_2.\boldsymbol{m}_2$. At the same time the zero-knowledge property is not weakened, except of course for leaking the fact that the $\boldsymbol{m}_i$'s satisfy this linear relation.

The protocol flow is defined as follows:

– P samples permutations π_1, π_2, π_3 at random.
  It then draws $\boldsymbol{v}_1, \boldsymbol{v}_2 \stackrel{\mathrm{R}}{\leftarrow} \mathcal{I}^v$, $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3 \stackrel{\mathrm{R}}{\leftarrow} \mathcal{I}^\ell$, $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3 \stackrel{\mathrm{R}}{\leftarrow} \mathcal{I}^k$, sets $\boldsymbol{v}_3 = \mathbf{X}_1.\boldsymbol{v}_1 \oplus \mathbf{X}_2.\boldsymbol{v}_2$ and then sends the following commitments for $i = 1, 2, 3$ to the verifier V:
$$C_{i0} \leftarrow \mathtt{Com}(\pi'_i = \pi_i, \boldsymbol{t}_{i0} = \mathbf{A}.(\boldsymbol{u}_i \| \boldsymbol{v}_i) \oplus \boldsymbol{f}_i)$$
$$C_{i1} \leftarrow \mathtt{Com}(\boldsymbol{t}_{i1} = \pi_i(\boldsymbol{f}_i))$$
$$C_{i2} \leftarrow \mathtt{Com}(\boldsymbol{t}_{i2} = \pi_i(\boldsymbol{f}_i \oplus \boldsymbol{e}_i))$$
– The verifier draws $c \stackrel{\mathrm{R}}{\leftarrow} \mathbb{Z}_3$ and sends it to P.
– Depending on the value of c, P opens the following commitments:
  0. P opens C_{i0}, C_{i1} by sending $\pi'_i, \boldsymbol{t}_{i0}, \boldsymbol{t}_{i1}$ and the associated random coins.

1. P opens C_{i0}, C_{i2} by sending $\pi'_i, \boldsymbol{t}_{i0}, \boldsymbol{t}_{i2}$ and the associated random coins.
2. P opens C_{i1}, C_{i2} by sending $\boldsymbol{t}_{i1}, \boldsymbol{t}_{i2}$ 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. V accepts, iff $\pi'_i \stackrel{?}{\in} \mathcal{S}_k$, there exist solutions $(\boldsymbol{a}_i, \boldsymbol{b}_i) \in \mathcal{I}^\ell \times \mathcal{I}^v$ to the equations $\boldsymbol{t}_{i0} \oplus \pi_i'^{-1}(\boldsymbol{t}_{i1}) = \mathbf{A}.(\boldsymbol{a}_i \| \boldsymbol{b}_i)$ and they satisfy $\boldsymbol{b}_3 = \mathbf{X}_1.\boldsymbol{b}_1 \oplus \mathbf{X}_2.\boldsymbol{b}_2$.
  1. V accepts, iff there exist solutions $(\boldsymbol{c}_i, \boldsymbol{d}_i) \in \mathcal{I}^\ell \times \mathcal{I}^v$ to the equations $\boldsymbol{t}_{i0} \oplus \pi_i'^{-1}(\boldsymbol{t}_{i2}) \oplus \boldsymbol{y}_i = \mathbf{A}.(\boldsymbol{c}_i \| \boldsymbol{d}_i)$ and they satisfy $\boldsymbol{d}_3 = \mathbf{X}_1.\boldsymbol{d}_1 \oplus \mathbf{X}_2.\boldsymbol{d}_2$.
  2. V accepts, iff $\|\boldsymbol{t}_{i1} \oplus \boldsymbol{t}_{i2}\|_1 \stackrel{?}{=} w$.

Theorem 4.2. *The above protocol is a Σ-protocol for the following relation:*

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

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

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

In particular, this allows to prove that $\boldsymbol{m}_1 = \mathbf{1}^v \oplus \boldsymbol{m}_2$, i.e., $\boldsymbol{m}_1$ is the bitwise negation of $\boldsymbol{m}_2$. 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 $\boldsymbol{y}_i = \mathbf{A}.(\boldsymbol{a}_i \oplus \boldsymbol{c}_i \| \boldsymbol{b}_i \oplus \boldsymbol{d}_i) \oplus \pi_i'^{-1}(\boldsymbol{t}_{i1} \oplus \boldsymbol{t}_{i2})$, where the second addend has low weight by the same arguments as earlier. Using the linear relations among the $\boldsymbol{b}_i$ and the $\boldsymbol{d}_i$ we further get $(\boldsymbol{b}_3 \oplus \boldsymbol{d}_3) = \mathbf{X}_1.(\boldsymbol{b}_1 \oplus \boldsymbol{d}_1) \oplus \mathbf{X}_2.(\boldsymbol{b}_2 \oplus \boldsymbol{d}_2)$. Thus, a valid witness is given by $\boldsymbol{r}'_i = \boldsymbol{a}_i \oplus \boldsymbol{c}_i$, $\boldsymbol{m}'_i = \boldsymbol{b}_i \oplus \boldsymbol{d}_i$ and $\boldsymbol{e}'_i = \pi_i'^{-1}(\boldsymbol{t}_{i1} \oplus \boldsymbol{t}_{i2})$.

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 $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3$ (where $\boldsymbol{y}_i = \mathbf{A}.(\boldsymbol{r}_i \| \boldsymbol{m}_i) \oplus \boldsymbol{e}_i$). That is, it allows one to prove that the messages satisfy $\boldsymbol{m}_3 = \boldsymbol{m}_1 \circ \boldsymbol{m}_2$. 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 $\tilde{\boldsymbol{m}}_1, \tilde{\boldsymbol{m}}_2, \tilde{\boldsymbol{m}}_3 \stackrel{R}{\leftarrow} \mathcal{I}^{4v}$ such that (1) $\tilde{\boldsymbol{m}}_3 = \tilde{\boldsymbol{m}}_1 \circ \tilde{\boldsymbol{m}}_2$ and (2) for all $(a,b) \in \mathcal{I}^2$ the number of indices $j \in \{1, \ldots, 4v\}$ satisfying $(\tilde{\boldsymbol{m}}_1[j], \tilde{\boldsymbol{m}}_2[j]) = (a,b)$ is exactly v. Further, the prover draws a random matrix $\mathbf{R} \stackrel{R}{\leftarrow} \mathcal{I}^{v \times 4v}$ with full rank such that each row has Hamming weight exactly 1 and such that $\mathbf{R}.\tilde{\boldsymbol{m}}_i = \boldsymbol{m}_i$ for $i = 1,2,3$ (so $\mathbf{R}$ is a $v \times v$ permutation matrix with $3v$ additional zero columns).

Now P and V basically run the protocol from the previous section to prove the linear relation $\mathbf{R}.\tilde{\boldsymbol{m}}_i = \boldsymbol{m}_i$, with the crucial difference that the relation $\mathbf{R}$ is not know to V, instead the prover additionally sends a commitment to $\mathbf{R}$ with the first message. Moreover P sends commitments to the $\tilde{\boldsymbol{m}}_i$'s to 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 $\mathbf{R}.\tilde{\boldsymbol{m}}_i = \boldsymbol{m}_i$ for the given c. If $c = 3$, the prover opens the commitments to the $\tilde{\boldsymbol{m}}_i$'s, and V checks if $\tilde{\boldsymbol{m}}_3 \stackrel{?}{=} \tilde{\boldsymbol{m}}_1 \circ \tilde{\boldsymbol{m}}_2$.

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

The zero knowledge property holds as even though $\mathbf{R}$ together with the $\tilde{\boldsymbol{m}}_i$'s determines the $\boldsymbol{m}_i$'s, each by itself is completely independent of the $\boldsymbol{m}_i$'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:

– P samples $\tilde{\boldsymbol{m}}_1, \tilde{\boldsymbol{m}}_2, \tilde{\boldsymbol{m}}_3 \stackrel{R}{\leftarrow} \mathcal{I}^{4v}$ such that $\tilde{\boldsymbol{m}}_3 = \tilde{\boldsymbol{m}}_1 \circ \tilde{\boldsymbol{m}}_2$ and such that for all $(a,b) \in \mathcal{I}^2$ the number of indices $j \in \{1, \ldots, 4v\}$ satisfying $(\tilde{\boldsymbol{m}}_1[j], \tilde{\boldsymbol{m}}_2[j]) = (a,b)$ is exactly v. Further, the prover draws a random matrix $\mathbf{R} \stackrel{R}{\leftarrow} \mathcal{I}^{v \times 4v}$ with full rank such that each row has at most Hamming weight 1 and such that $\mathbf{R}.\tilde{\boldsymbol{m}}_i = \boldsymbol{m}_i$ for $i = 1,2,3$.
  In the following we denote the j^{th} v-bit block of $\tilde{\boldsymbol{m}}_i$ by $\tilde{\boldsymbol{m}}_i^j$, i.e., $\tilde{\boldsymbol{m}}_i^j = (\tilde{\boldsymbol{m}}_{iu})_{u=(j-1)v+1}^{jv}$. 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.
  P draws $\tilde{\boldsymbol{r}}_i^j \stackrel{R}{\leftarrow} \mathcal{I}^{\ell}$ and defines auxiliary images as $\tilde{\boldsymbol{y}}_i^j = \mathbf{A}.(\tilde{\boldsymbol{r}}_i^j \| \tilde{\boldsymbol{m}}_i^j) \oplus \tilde{\boldsymbol{e}}_i^j$ for $\tilde{\boldsymbol{e}}_i^j \stackrel{R}{\leftarrow} \mathcal{I}_w^k$. It then samples permutations $\pi_i, \pi_i^j \leftarrow \mathcal{S}_k$ at random.

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

$$\tilde{C} \leftarrow \mathtt{Com}(\tilde{\boldsymbol{y}}_1'^1 = \tilde{\boldsymbol{y}}_1^1, \dots, \tilde{\boldsymbol{y}}_3'^4 = \tilde{\boldsymbol{y}}_3^4) \qquad C_{\mathbf{R}} \leftarrow \mathtt{Com}(\mathbf{R}' = \mathbf{R})$$

$$C_{i0} \leftarrow \mathtt{Com}(\pi_i' = \pi_i, \boldsymbol{t}_{i0} = \mathbf{A}.(\boldsymbol{u}_i \| \boldsymbol{v}_i) \oplus \boldsymbol{f}_i)$$

$$C_{i1} \leftarrow \mathtt{Com}(\boldsymbol{t}_{i1} = \pi_i(\boldsymbol{f}_i)) \qquad C_{i2} \leftarrow \mathtt{Com}(\boldsymbol{t}_{i2} = \pi_i(\boldsymbol{f}_i \oplus \boldsymbol{e}_i))$$

$$C_{i0}^j \leftarrow \mathtt{Com}(\pi_i'^j = \pi_i^j, \boldsymbol{t}_{i0}^j = \mathbf{A}.(\boldsymbol{u}_i^j \| \boldsymbol{v}_i^j) \oplus \boldsymbol{f}_i^j)$$

$$C_{i1}^j \leftarrow \mathtt{Com}(\boldsymbol{t}_{i1}^j = \pi_i^j(\boldsymbol{f}_i^j)) \qquad C_{i2}^j \leftarrow \mathtt{Com}(\boldsymbol{t}_{i2}^j = \pi_i^j(\boldsymbol{f}_i^j \oplus \tilde{\boldsymbol{e}}_i^j))$$

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

Theorem 4.3. *The above protocol is a Σ-protocol for the following relation:*

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

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

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 $\tilde{\boldsymbol{m}}_i^j$ are uniformly random in their domain, and do not leak any information about the $\boldsymbol{m}_i$, as long as the matrix $\mathbf{R}$ is kept secret. Similarly, if the $\tilde{\boldsymbol{m}}_i^j$ 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 $\tilde{\boldsymbol{y}}_i^j$ 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 $\boldsymbol{m}_0, \boldsymbol{m}_1$ satisfy $\boldsymbol{m}_0 = C(\boldsymbol{m}_1)$ for an arbitrary circuit C. Let $C_1, \ldots, C_d$ denote the layers of C, i.e., $C(\boldsymbol{m}_1) = C_d(\ldots C_1(\boldsymbol{m}_1)\ldots)$, 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 ℓ, where ℓ 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ where $\boldsymbol{x}_1 = \boldsymbol{m}_1, \boldsymbol{x}_2 = C_1(\boldsymbol{m}_1), \ldots, \boldsymbol{x}_d = C(\boldsymbol{m}_1)$ (note that we already have commitments to $\boldsymbol{x}_1 = \boldsymbol{m}_1$ and $\boldsymbol{x}_d = \boldsymbol{m}_0$). Now we use our Σ-protocols to prove that $\boldsymbol{x}_{i+1} = C_i(\boldsymbol{x}_i)$ for $i = 1 \ldots d-1$.

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 Σ-protocols which allow one to prove arbitrary relations among secret values $\boldsymbol{m}_i$, i.e., $\boldsymbol{m}_0 = C(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_u)$ 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

ACPS09. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009.

AIK09. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography with Constant Input Locality. *Journal of Cryptology*, 22(4):429–469, 2009.

AJLA+12. G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 483–501. Springer, 2012.

BDP00. J. Boyar, I. Damgård, and R. Peralta. Short Non-Interactive Cryptographic Proofs. *Journal of Cryptology*, 13(4):449–472, 2000.

BFKL93. A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In D. R. Stinson, editor, *CRYPTO 93*, volume 773 of *LNCS*, pages 278–291. Springer, 1993.

BG93. M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. In E. F. Brickell, editor, *CRYPTO 92*, volume 740 of *LNCS*, pages 390–420. Springer, 1993.

BKW03. A. Blum, A. Kalai, and H. Wasserman. Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. *Journal of the ACM*, 50(4):506–519, 2003.

BMvT78. E. Berlekamp, R. McEliece, and H. van Tilborg. On the Inherent Intractability of Certain Coding Problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

CC06. H. Chen and R. Cramer. Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computations over Small Fields. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 521–536. Springer, 2006.

CD97. R. Cramer and I. Damgård. Linear Zero-Knowledge - A Note on Efficient Zero-Knowledge Proofs and Arguments. In F. T. Leighton and P. W. Shor, editors, *STOC 97*, pages 436–445. ACM, 1997.

CD98. R. Cramer and I. Damgård. Zero-Knowledge Proofs for Finite Field Arithmetic; or: Can Zero-Knowledge Be for Free? In H. Krawczyk, editor, *CRYPTO 98*, volume 1462 of *LNCS*, pages 424–441. Springer, 1998.

CD09. R. Cramer and I. Damgård. On the Amortized Complexity of Zero-Knowledge Protocols. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 177–191. Springer, 2009.

CLRS10. P.-L. Cayrel, R. Lindner, M. Rückert, and R. Silva. Improved Zero-Knowledge Identification with Lattices. In *ProvSec*, pages 1–17, 2010.

Cra97. R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and University of Amsterdam, 1997.

CVA10. P.-L. Cayrel, P. Véron, and S. M. El Yousfi Alaoui. A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem. In A. Biryukov, G. Gong, and D. R. Stinson, editors, *Selected Areas in Cryptography – SAC 2010*, volume 6544 of *LNCS*, pages 171–186. Springer, 2010.

Dam04. I. Damgård. On Σ-Protocols. Lecture on Cryptologic Protocol Theory; Faculty of Science, University of Aarhus, 2004.

DGOW95. I. Damgård, O. Goldreich, T. Okamoto, and A. Wigderson. Honest Verifier vs Dishonest Verifier in Public Coin Zero-Knowledge Proofs. In D. Coppersmith, editor, *CRYPTO 95*, volume 963 of *LNCS*, pages 325–338. Springer, 1995.

GMW87. O. Goldreich, S. Micali, and A. Wigderson. How to Play Any Mental Game, or A Completeness Theorem for Protocols with Honest Majority. In A. V. Aho, editor, *STOC 87*, pages 218–229. ACM, 1987.

GRS08. H. Gilbert, M. J. B. Robshaw, and Y. Seurin. How to Encrypt with the LPN Problem. In *Automata, Languages and Programming – ICALP 2008*, LNCS, pages 679–690. Springer, 2008.

GS08. J. Groth and A. Sahai. Efficient Non-interactive Proof Systems for Bilinear Groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.

HB01. N. J. Hopper and M. Blum. Secure Human Identification Protocols. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 52–66. Springer, 2001.

IKOS07. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge from secure multiparty computation. In D. S. Johnson and U. Feige, editors, *STOC 2007*, pages 21–30. ACM, 2007.

JW05. A. Juels and S. A. Weis. Authenticating Pervasive Devices with Human Protocols. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 293–308. Springer, 2005.

Kea98. M. J. Kearns. Efficient Noise-Tolerant Learning from Statistical Queries. *Journal of the ACM*, 45(6):983–1006, 1998.

Kil92. J. Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In *STOC 92*, pages 723–732, 1992.

KMO90. J. Kilian, S. Micali, and R. Ostrovsky. Minimum Resource Zero-Knowledge Proofs (Extended Abstract). In G. Brassard, editor, *CRYPTO 89*, volume 435 of *LNCS*, pages 545–546. Springer, 1990.

KP98. J. Kilian and E. Petrank. An Efficient Noninteractive Zero-Knowledge Proof System for NP with General Assumptions. *Journal of Cryptology*, 11(1):1–27, 1998.

KPC+11. E. Kiltz, K. Pietrzak, D. Cash, A. Jain, and D. Venturi. Efficient Authentication from Hard Learning Problems. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 7–26. Springer, 2011.

KR06. Y. T. Kalai and R. Raz. Succinct Non-Interactive Zero-Knowledge Proofs with Preprocessing for LOGSNP. In *FOCS 2006*, pages 355–366. IEEE Computer Society, 2006.

KSS10. J. Katz, J. S. Shin, and A. Smith. Parallel and Concurrent Security of the HB and HB^+ Protocols. *Journal of Cryptology*, 23(3):402–421, 2010.

KTX08. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.

Mic00. S. Micali. Computationally Sound Proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

MM11. D. Micciancio and P. Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, 2011.

Reg05. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.

Ste93. J. Stern. A New Identification Scheme Based on Syndrome Decoding. In D. R. Stinson, editor, *CRYPTO 93*, volume 773 of *LNCS*, pages 13–21. Springer, 1993.

Vér96. P. Véron. Improved Identification Schemes Based on Error-Correcting Codes. *Applicable Algebra in Engineering, Communication and Computing*, 8(1):57–69, 1996.

A A Gap in the Proof of Véron's Protocol [Vér96]

We now briefly present the protocol of Véron [Vér96] and show where the gap in his proof is. We here write $\mathcal{I}_w^k$ to denote the subset of $\mathcal{I}^k$ with Hamming weight equal to w.

Let P, V get common input $\mathbf{A} \in \mathcal{I}^{k \times \ell}, \boldsymbol{y} \in \mathcal{I}^k$, and P gets $\boldsymbol{s} \in \mathcal{I}^\ell, \boldsymbol{e} \in \mathcal{I}_w^k$ such that $\mathbf{A}.\boldsymbol{s} \oplus \boldsymbol{e} = \boldsymbol{y}$ as a private input. The protocol flow is then given as follows:

- P samples $\boldsymbol{v} \xleftarrow{\text{R}} \mathcal{I}^\ell, \boldsymbol{f} \xleftarrow{\text{R}} \mathcal{I}^k$ (note that $\boldsymbol{f}$ is uniformly random) and a random permutation $\sigma \xleftarrow{\text{R}} \mathcal{S}^k$. P then sends commitments C_0, C_1, C_2 to V, where
$$C_0 \leftarrow \mathtt{Com}(\sigma' = \sigma)$$
$$C_1 \leftarrow \mathtt{Com}(\boldsymbol{t}_1 = \sigma(\mathbf{A}.(\boldsymbol{v} \oplus \boldsymbol{s})))$$
$$C_2 \leftarrow \mathtt{Com}(\boldsymbol{t}_2 = \sigma(\mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{y}))$$
- The verifier draws $c \xleftarrow{\text{R}} \mathbb{Z}_3$ 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 $\sigma', \boldsymbol{t}_1$ and the associated random coins.
  1. P opens C_0, C_2 by sending $\sigma', \boldsymbol{t}_2$ and the associated random coins.
  2. P opens C_1, C_2 by sending $\boldsymbol{t}_1, \boldsymbol{t}_2$ 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 $\sigma'^{-1}(\boldsymbol{t}_1) \overset{?}{\in} \operatorname{img} \mathbf{A}$.
  1. V accepts, iff $\sigma'^{-1}(\boldsymbol{t}_2) \oplus \boldsymbol{y} \overset{?}{\in} \operatorname{img} \mathbf{A}$.
  2. V accepts, iff $\|\boldsymbol{t}_1 \oplus \boldsymbol{t}_2\|_1 \overset{?}{=} w$.

The gap. The gap in the proof of security of this protocol is in the part of the zero-knowledge property. More precisely, the simulator in the case of $c = 2$ samples $\hat{\sigma} \xleftarrow{\text{R}} \mathcal{S}^k$, $\boldsymbol{a} \xleftarrow{\text{R}} \mathcal{I}^\ell$ and $\hat{\boldsymbol{e}} \xleftarrow{\text{R}} \mathcal{I}_w^k$ and sets $\hat{\boldsymbol{t}}_1 = \hat{\sigma}(\mathbf{A}.\boldsymbol{a})$ and $\hat{\boldsymbol{t}}_2 := \boldsymbol{t}_1 \oplus \hat{\sigma}(\hat{\boldsymbol{e}})$. In the paper it is argued that given $\mathbf{A}$ and $\boldsymbol{y}$ both $\hat{\boldsymbol{t}}_1$ and $\hat{\boldsymbol{t}}_2$ follow the same distribution as $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ in real protocol runs. While this is true for each one individually (as shown in the paper), it is not proven for the combined distribution of $(\hat{\boldsymbol{t}}_1, \hat{\boldsymbol{t}}_2)$.

At a more intuitive level, in the case of $c = 2$, the verifier is given $\boldsymbol{t}_1 = \sigma(\mathbf{A}.(\boldsymbol{v} \oplus \boldsymbol{s}))$ and $\boldsymbol{t}_2 = \sigma(\mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{y})$ and therefore $\sigma(\boldsymbol{e})$ as well. However, the former gives some information about σ because $\mathbf{A}.(\boldsymbol{v} \oplus \boldsymbol{s})$ is not a random vector for a fixed $\mathbf{A}$, as for instance $\mathbf{A}$ might be such that all its images have their last coordinates equal to 0. This leaks some information about σ and thus about $\boldsymbol{e}$ as well, and therefore we cannot prove perfect zero knowledge for this case as claimed.

B A Protocol for the Standard LPN-Problem

As mentioned in the main part of the paper, the hardness of the xLPN-problem holds assuming the hardness of the standard LPN-problem, but the reduction is not tight, as it is based on the Goldreich-Levin theorem. In the following we thus present an alternative to our commitment scheme and protocol, whose security is directly based on the standard LPN-problem. The protocol has a knowledge error of 4/5, and thus running the protocol twice in parallel roughly achieves the same knowledge error as the protocols in Section 4. Depending on the concrete parameters being used, either of the solutions may be more efficient.

We first note that under the LPN_τ-problem still remains hard if the weight of the error $\boldsymbol{e}$ is limited by twice its expectation value, where Ber_τ denotes the Bernoulli distribution with parameter τ:

Lemma B.1. *The following two probability distributions are statistically indistinguishable:*

$$\{(\mathbf{A}, \mathbf{A}.\boldsymbol{x} \oplus \boldsymbol{e})\} \quad \textit{and} \quad \{(\mathbf{A}, \mathbf{A}.\boldsymbol{x} \oplus \boldsymbol{f})\},$$

where $\mathbf{A} \xleftarrow{R} \mathcal{I}^{k \times m}$, $\boldsymbol{e} \xleftarrow{R} \mathsf{Ber}_\tau^k$, $\boldsymbol{f} \xleftarrow{R} \mathsf{Ber}_\tau^k$ *conditioned on* $\|\boldsymbol{f}\|_1 \leq 2\tau k$, *and* $\boldsymbol{x} \xleftarrow{R} \mathcal{I}^m$ *is fixed and secret.*

Proof. Let $\boldsymbol{r} \leftarrow \mathsf{Ber}_\tau^k$. Hoeffding's inequality then states that $Pr[\|\boldsymbol{r}\|_1 \geq (\tau + \varepsilon)k] \leq e^{-2\varepsilon^2 k}$. Setting $\varepsilon = \tau$ we get that $Pr[\|\boldsymbol{r}\|_1 \geq 2\tau k] \leq e^{-2\tau^2 k}$. Thus the statistical difference of the two random ensembles is bounded above by $2e^{-2\tau^2 k}$, which is negligible for any fixed $0 < \tau \leq 1$. □

Having this, it is straightforward to see that Theorem 3.1 literally holds true even if the error of the commitment scheme is drawn as $\boldsymbol{e} \xleftarrow{R} \mathsf{Ber}_\tau^k$ conditioned on $\|\boldsymbol{f}\|_1 \leq 2\tau k$, as long as the matrix $\mathbf{A}$ has minimal distance at least $4\lfloor \tau k \rceil$ (instead of $2\lfloor \tau k \rceil$ stated there).

We next describe the Σ-protocol for proving knowledge of a valid opening of a commitment under this variant. The constructions for linear and multiplicative relations from Sections 4.2 and 4.3 can easily be adapted for this setting as well.

In contrast to the protocols given in the main part of this paper, the protocol has knowledge error 4/5 instead of 2/3, which again can be reduced arbitrarily by parallel repetition.

In the description we write $\mathcal{S}_w^k$ for the set of all $k \times k$-matrices of rank $k - w$, where each column and each row has at most Hamming weight 1. One can think of $\mathcal{S}_w^k$ as all mappings that send w indices to 0, and permute the remaining $k - w$ ones.te the remaining $k - w$ ones.

– P samples a $\tilde{\sigma} \xleftarrow{R} \mathcal{S}_w^k$ satisfying $\tilde{\sigma}(\boldsymbol{e}) = \mathbf{0}$ and a permutation π at random. It defines $\sigma = \tilde{\sigma}.\pi^{-1}$.
  It then draws $\boldsymbol{v} \xleftarrow{R} \mathcal{I}^{\ell+v}$, $\boldsymbol{f} \xleftarrow{R} \mathcal{I}^k$, and then sends the following commitments to the verifier V:

$$\begin{array}{ll} C_0 \leftarrow \mathtt{Com}(\pi' = \pi, \boldsymbol{t}_0 = \mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{f}) & C_1 \leftarrow \mathtt{Com}(\boldsymbol{t}_1 = \pi(\boldsymbol{f})) \\ C_2 \leftarrow \mathtt{Com}(\boldsymbol{t}_2 = \pi(\boldsymbol{f} \oplus \boldsymbol{e})) & C_3 \leftarrow \mathtt{Com}(\sigma' = \sigma) \\ C_4 \leftarrow \mathtt{Com}(\boldsymbol{t}_4 = \tilde{\sigma}(\boldsymbol{f})) & C_5 \leftarrow \mathtt{Com}(\boldsymbol{t}_5 = \tilde{\sigma}(\boldsymbol{f} \oplus \boldsymbol{e})) \end{array}$$

– The verifier draws $c \xleftarrow{R} \mathbb{Z}_5$ 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 $\pi', \boldsymbol{t}_0, \boldsymbol{t}_1$ and the associated random coins.
  1. P opens C_0, C_2 by sending $\pi', \boldsymbol{t}_0, \boldsymbol{t}_2$ and the associated random coins.
  2. P opens C_1, C_3, C_4 by sending $\boldsymbol{t}_1, \sigma', \boldsymbol{t}_4$ and the associated random coins.
  3. P opens C_2, C_3, C_5 by sending $\boldsymbol{t}_2, \sigma', \boldsymbol{t}_5$ and the associated random coins.
  4. P opens C_4, C_5 by sending $\boldsymbol{t}_4, \boldsymbol{t}_5$ 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 $\boldsymbol{t}_0 \oplus \pi'^{-1}(\boldsymbol{t}_1) \stackrel{?}{\in} \operatorname{img} \mathbf{A}$ and $\pi' \stackrel{?}{\in} \mathcal{S}^k$.
  1. V accepts, iff $\boldsymbol{t}_0 \oplus \pi'^{-1}(\boldsymbol{t}_2) \oplus \boldsymbol{y} \stackrel{?}{\in} \operatorname{img} \mathbf{A}$.
  2. V accepts, iff $\sigma'(\boldsymbol{t}_1) \stackrel{?}{=} \boldsymbol{t}_4$ and $\sigma' \stackrel{?}{\in} \mathcal{S}^k_w$.
  3. V accepts, iff $\sigma'(\boldsymbol{t}_2) \stackrel{?}{=} \boldsymbol{t}_5$.
  4. V accepts, iff $\|\boldsymbol{t}_4 \oplus \boldsymbol{t}_5\|_1 \stackrel{?}{=} 0$.

Theorem B.2. *The above protocol is a Σ-protocol for the following relation:*

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

We note that in a straightforward manner our protocol can be modified to prove claims of the more general form $w_1 \leq \|\boldsymbol{e}\|_1 \leq w_2$ for $0 \leq w_1 \leq w_2 \leq k$ as well.

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. $\boldsymbol{t}_0 \oplus \pi'^{-1}(\boldsymbol{t}_1) = (\mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{f}) \oplus \pi^{-1}(\pi(\boldsymbol{f})) = \mathbf{A}.\boldsymbol{v} \in \operatorname{img} \mathbf{A}$ and π is a permutation.
1. $\boldsymbol{t}_0 \oplus \pi'^{-1}(\boldsymbol{t}_2) \oplus \boldsymbol{y} = (\mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{f}) \oplus \pi^{-1}(\pi(\boldsymbol{f} \oplus \boldsymbol{e})) \oplus (\mathbf{A}.\boldsymbol{s} \oplus \boldsymbol{e}) = \mathbf{A}.(\boldsymbol{v} \oplus \boldsymbol{s}) \in \operatorname{img} \mathbf{A}$.
2. $\sigma'(\boldsymbol{t}_1) = \tilde{\sigma}(\pi^{-1}(\pi(\boldsymbol{f}))) = \tilde{\sigma}(\boldsymbol{f}) = \boldsymbol{t}_4$ and $\tilde{\sigma}$ has the correct form.
3. $\sigma'(\boldsymbol{t}_2) = \tilde{\sigma}(\pi^{-1}(\pi(\boldsymbol{f} \oplus \boldsymbol{e}))) = \tilde{\sigma}(\boldsymbol{f} \oplus \boldsymbol{e}) = \boldsymbol{t}_5$.
4. $\|\boldsymbol{t}_4 \oplus \boldsymbol{t}_5\|_1 = \|\tilde{\sigma}(\boldsymbol{f}) \oplus \tilde{\sigma}(\boldsymbol{f} \oplus \boldsymbol{e})\|_1 = \|\tilde{\sigma}(\boldsymbol{f} \oplus \boldsymbol{f} \oplus \boldsymbol{e})\|_1 = \|\tilde{\sigma}(\boldsymbol{e})\|_1 = \|\mathbf{0}\|_1 = 0$.

Special Soundness. Assume that we have fixed values $C_0, \ldots, C_5$ and openings for all challenges $c \in \mathbb{Z}_5$, such that the verifier accepts on all of them. Then, by the assumed perfect binding property of the underlying commitment scheme `Com`(.), 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 $\pi'^{-1}(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2) \oplus \boldsymbol{y} \in \operatorname{img} \mathbf{A}$ and thus that $\boldsymbol{y} = \mathbf{A}.\boldsymbol{s}' \oplus \pi'^{-1}(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2)$, where $\boldsymbol{s}' = (\boldsymbol{r}'\|\boldsymbol{m}')$ is easy to compute. Furthermore, by adding the verification equations for $c = 2$ and $c = 3$ we get that $\sigma'(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2) = \boldsymbol{t}_4 \oplus \boldsymbol{t}_5$. Now, using that $\|\boldsymbol{t}_4 \oplus \boldsymbol{t}_5\|_1 = 0$ (from $c = 4$) and the fact that σ' is a partial permutation that sends exactly w coordinates to 0, we can infer that every preimage of $\boldsymbol{t}_4 \oplus \boldsymbol{t}_5$ under σ' has at most norm w, resulting in $\|\boldsymbol{t}_1 \oplus \boldsymbol{t}_2\|_1 \leq w$.

A valid witness of $(\mathbf{A}, \boldsymbol{y})$ is thus given by $(\boldsymbol{r}', \boldsymbol{m}', \pi'^{-1}(\boldsymbol{t}_1 \oplus \boldsymbol{t}_2))$.

Honest-Verifier Zero-Knowledge. In the following we describe an efficient simulator S, which for each challenge $c \in \mathbb{Z}_5$ 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 S computes C_0 and C_1 like an honest prover, and computes $C_2, \ldots, C_5$ as independent commitments to 0. Then, clearly, the distribution of $C_0, C_1, \pi', \boldsymbol{t}_0, \boldsymbol{t}_1$ is identical to that in real protocol transcripts. Furthermore, by the computational blinding property of the commitment scheme $\mathtt{Com}(.)$, the distribution of $C_2, \ldots, C_5$ is computationally indistinguishable from that in real protocol runs.
1. For $c = 1$, the simulator draws $\pi \xleftarrow{R} \mathcal{S}^k$, $\boldsymbol{a} \xleftarrow{R} \mathcal{I}^k$ and $\boldsymbol{b} \xleftarrow{R} \mathcal{I}^{\ell+v}$. It sets $C_0 = \mathtt{Com}(\pi, \mathbf{A}.\boldsymbol{b} \oplus \boldsymbol{y} \oplus \boldsymbol{a})$ and $C_2 = \mathtt{Com}(\pi(\boldsymbol{a}))$. All other commitments are computed as commitments to 0. It easy to see that the openings of C_0, C_2 pass the verification equations. To see the correctness of their distributions note that $\boldsymbol{t}_2$ in the real protocol run and $\pi(\boldsymbol{a})$ 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 $\boldsymbol{t}_0 = \mathbf{A}.\boldsymbol{v} \oplus \boldsymbol{f}$, where $\boldsymbol{v}$ is uniformly at random, and $\boldsymbol{f} = \pi^{-1}(\boldsymbol{t}_2 \oplus \boldsymbol{e})$; in the simulated transcript the content of C_0 is given by $\mathbf{A}.(\boldsymbol{b} \oplus \boldsymbol{s}) \oplus (\boldsymbol{a} \oplus \boldsymbol{e})$. Now, $\boldsymbol{v}$ and $\boldsymbol{b} \oplus \boldsymbol{s}$ are both uniformly random, and the terms $\boldsymbol{f}$ and $\boldsymbol{a} \oplus \boldsymbol{e}$ 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. Again, the distribution of the remaining commitments is computationally indistinguishable by the assumed blinding property of $\mathtt{Com}(.)$.
2. First, S draws $\sigma' \xleftarrow{R} \mathcal{S}_w^k$ and $\boldsymbol{a} \leftarrow \mathcal{I}^k$. It sets $C_1 = \mathtt{Com}(\boldsymbol{a})$, $C_3 = \mathtt{Com}(\sigma')$ and $C_4 = \mathtt{Com}(\sigma'(\boldsymbol{a}))$. All other commitments are computed as commitments to 0. Clearly, the distribution C_1, C_3, C_4 and the openings are perfectly correct. As before, the C_0, C_2, C_5 are computationally indistinguishable from those in real protocol runs by the properties of $\mathtt{Com}(.)$.
3. Here, S draws $\sigma' \xleftarrow{R} \mathcal{S}_w^k$ and $\boldsymbol{a} \leftarrow \mathcal{I}^k$. It sets $C_1 = \mathtt{Com}(\boldsymbol{a})$, $C_2 = \mathtt{Com}(\sigma')$ and $C_5 = \mathtt{Com}(\sigma'(\boldsymbol{a}))$. All other commitments are computed as commitments to 0. Clearly, the distribution C_2, C_3, C_5 and the openings are perfectly correct. As before, the C_0, C_1, C_4 are computationally indistinguishable from those in real protocol runs by the hiding property of $\mathtt{Com}(.)$.
4. Finally, for $c = 4$, the simulator draws $\sigma' \xleftarrow{R} \mathcal{S}_w^k$, and draws $\boldsymbol{a} \xleftarrow{R} \mathcal{I}^k$ and $\boldsymbol{b} \xleftarrow{R} \mathcal{I}_w^k$ such that $\sigma'(\boldsymbol{b}) = \mathbf{0}$. It computed $C_0, \ldots, C_3$ as commitments to 0, $C_4 = \mathtt{Com}(\sigma'(\boldsymbol{a}))$ and $C_5 = \mathtt{Com}(\sigma'(\boldsymbol{a} \oplus \boldsymbol{b}))$. As before, the distributions of $C_0, \ldots, C_3$ are computationally indistinguishable from real protocol runs by the binding property of $\mathtt{Com}(.)$, and C_4 and C_5 as well as their openings can easily be seen to perfectly simulate the behavior of an honest prover. $\square$