Amortized Complexity of Zero-Knowledge Proofs Revisited: Achieving Linear Soundness Slack

Ronald Cramer and Ivan Damgård*

CWI, Amsterdam & Mathematical Institute, Leiden University, and Department of Computer Science, Aarhus University

cramer@cwi.nl, ivan@cs.au.dk

Abstract. We propose a new zero-knowledge protocol for proving knowledge of short preimages under additively homomorphic functions that map integer vectors to an Abelian group. The protocol achieves amortized efficiency in that it only needs to send $O(n)$ auxiliary function values to prove knowledge of n preimages. Furthermore we significantly improve previous bounds on how short a secret we can extract from a dishonest prover, namely our bound is a factor $O(k)$ larger than the size of secret used by the honest prover. In the best previous result, the factor was $O(k^{\log k} n)$. Our main technique is derived from the theory of expanders.

Our protocol can be applied to give proofs of knowledge for plaintexts in (Ring-)LWE-based cryptosystems, knowledge of preimages of homomorphic hash functions as well as knowledge of committed values in some integer commitment schemes.

Keywords: Proofs of Plaintext Knowledge, Lattice-based Encryption, Homomorphic Hashing, Integer Commitments

1 Introduction

Proofs of Knowledge In a zero-knowledge protocol, a prover demonstrates that some claim is true (and in some cases that he knows a proof) while giving the verifier no other knowledge beyond the fact that the claim is true. Zero-knowledge protocols are essential tools in cryptographic protocol design. For instance, one needs zero-knowledge proofs of knowledge in multiparty computation to have a player demonstrate that he knows the input he is providing.

* Supported by The Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed; by the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed; and by the Advanced ERC grant MPCPRO.

In this work, we will consider the problem of proving knowledge of a preimage under a one-way functions $f : \mathbb{Z}^r \mapsto G$ where G is an Abelian group (written additively in the following), and where furthermore the function is additively homormorphic, i.e., $f(\boldsymbol{a})+f(\boldsymbol{b}) = f(\boldsymbol{a}+\boldsymbol{b})$. We will call such functions *ivOWF*'s (for homomorphic One-Way Functions over Integer Vectors). This problem was considered in several earlier works, in particular recently in [BDLN16], from where we have borrowed most of the notation and basic definitions we use in the following.
ivOWF turns out to be a very general notion. Examples of ivOWFs include:

– The encryption function of several (Ring-)LWE-based cryptosystems(such as the one introduced in [BGV12] and used in the so-called SPDZ protocol [DPSZ12]).
– The encryption function of any semi-homomorphic cryptosystem as defined in [BDOZ11].
– The commitment function in commitment schemes for committing to integer values (see, e.g., [DF02]).
– Hash functions based on lattice problems such as [GGH96,LMPR08], where it is hard to find a short preimage.

We will look at the scenario where a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ are given $y \in G$ and $\mathcal{P}$ holds a short preimage $\boldsymbol{x}$ of y, i.e., such that $||\boldsymbol{x}|| \leq \beta$ for some β. $\mathcal{P}$ wants to prove in zero-knowledge that he knows such an $\boldsymbol{x}$. When f is an encryption function and y is a ciphertext, this can be used to demonstrate that the ciphertext decrypts and $\mathcal{P}$ knows the plaintext. When f is a commitment function this can be used to show that one has committed to a number in a certain interval.
A well-known, simple but inefficient solution is the following protocol π:

(1) $\mathcal{P}$ chooses $\boldsymbol{r}$ at random such that $||\boldsymbol{r}|| \leq \tau \cdot \beta$ for some sufficiently large τ, the choice of which we return to below.
(2) $\mathcal{P}$ then sends $a = f(\boldsymbol{r})$ to $\mathcal{V}$.
(3) $\mathcal{V}$ sends a random challenge bit b.
(4) $\mathcal{P}$ responds with $\boldsymbol{z} = \boldsymbol{r} + b \cdot \boldsymbol{x}$.
(5) $\mathcal{V}$ checks that $f(\boldsymbol{z}) = a + b \cdot y$ and that $||\boldsymbol{z}|| \leq \tau \cdot \beta$.

If τ is sufficiently large, the distribution of $\boldsymbol{z}$ will be statistically independent of $\boldsymbol{x}$, and the protocol will be honest verifier statistical zero-knowledge[1]. On the other hand, we can extract a preimage of y from a cheating prover who can produce correct answers $\boldsymbol{z}_0, \boldsymbol{z}_1$ to $b = 0, b = 1$, namely $f(\boldsymbol{z}_1 - \boldsymbol{z}_0) = y$. Clearly, we have $||\boldsymbol{z}_1 - \boldsymbol{z}_0|| \leq 2 \cdot \tau \cdot \beta$. We will refer to the factor 2τ as the *soundness slack* of the protocol, because it measures the discrepancy between the interval used by the honest prover and what we can force a dishonest prover to do. The value of the soundness slack is important: if f is, e.g., an encryption function, then a

[1] We will only be interested in honest verifier zero-knowledge here. In applications one would get security for malicious verifiers by generating the challenge in a trusted way, e.g., using a maliciously sure coin-flip protocol.

large soundness slack will force us to use larger parameters for the underlying cryptosystem to ensure that the ciphertext decrypts even if the input is in the larger interval, and this will cost us in efficiency.

The naive protocol above requires an exponentially large slack to get zero-knowledge, but using Lyubachevsky's rejection sampling technique, the soundness slack can made polynomial or even constant (at least in the random oracle model).

The obvious problem with the naive solution is that one needs to repeat the protocol k times where k is the statistical security parameter, to get soundness error probability 2^{-k}. This means that one needs to generate $\Omega(k)$ auxiliary f-values. We will refer to this as the *overhead* of the protocol and use it as a measure of efficiency.

One wants, of course as small overhead and soundness slack as possible, but as long as we only want to give a proof for a single f-value, we do not know how to reduce the overhead dramatically in general. But if instead we want to give a proof for k or more f-values, then we know how to reduce the *amortised* overhead: Cramer and Damgård [CD09] show how to get amortised overhead $O(1)$, but unfortunately the soundness slack is $2^{\Omega(k)}$, even if rejection sampling is used. In [DKL+13] two protocols were suggested, where one is only covertly secure. The other one can achieve polynomial soundness slack with overhead $\Omega(k)$ and works only in the random oracle model[2]. This was improved in [BDLN16]: a protocol was obtained (without random oracles) that has $O(1)$ overhead and quasi polynomial soundness slack (proportional to $n \cdot (2k+1)^{\log(k)/2}$).

1.1 Contributions & Techniques

In this paper, we improve significantly the result from [BDLN16] and [DKL+13]: we obtain $O(1)$ overhead and soundness slack $O(k)$. All results hold in the standard model (no random oracles are needed).

Our protocol uses a high-level strategy similar to [BDLN16]:

(1) Do a cut-and-choose style protocol for the inputs $y_1, \ldots, y_n$. This is a relatively simple but imperfect proof of knowledge: It only guarantees that the prover knows almost all preimages.
(2) Let the verifier assign each y_i to one of several *buckets* following a randomised strategy.
(3) For each bucket, add all elements that landed in the bucket and do an imperfect proof of knowledge as in the first step, but now with all the bucket sums as input.

[2] The protocol in [DKL+13] is actually stated as a proof of plaintext knowledge for random ciphertexts, but generalizes to a protocol for ivOWFs. It actually offers a tradeoff between soundness slack s and overhead in the sense that the overhead is $M \cdot \log(k)$, where M has to be chosen such that the error probability $(1/s)^M$ is negligible. Thus to get exponentially small error probability in k as we do here, one can choose s to be $\mathsf{poly}(k)$ and hence M will be $\Omega(k/\log k)$.

The reason why one might hope this would work is as follows: as mentioned, the first step will ensure that we can extract *almost* all of the required n preimages, in fact we can extract all but k preimages (we assume throughout that $n >> k$). In the second step, since we only have k elements left that were "bad" in the sense that we could not yet extract a preimage, then if we have many more than k buckets, say $\Theta(n)$ and distribute them in buckets according to a carefully designed strategy, we may hope that with overwhelming probability, all the bad elements will be alone in one of those buckets for which we can extract a preimage of the bucket sum. This seems plausible because we can extract almost all such preimages. If indeed this happens, we can extract all remaining preimages by linearity of f: each bad element can be written as a sum of elements for which the extractor already knows a preimage.

Furthermore, the overall cost of doing the protocol would be $O(n)$, and the soundness slack will be limited by the maximal number of items in a bucket. In fact, if each bucket contains $O(k)$ elements, then the soundness slack is $O(k)$ as well. It turns out that we can derive a (randomised) strategy for assignment to buckets by modifying the construction of certain expander graphs and this is our main technical contribution, the intuition of which we explain shortly.

In comparison, the protocol from [BDLN16] also plays a "balls and bins" game. The difference is that they use only $O(k)$ buckets, but repeat the game $O(\log k)$ times. This means that their extraction takes place in $\Omega(\log k)$ stages, which leads to the larger soundness slack.

Our protocol is honest verifier zero-knowlegde and is sound in the sense of a standard proof of knowledge, i.e., we extract the prover's witness by rewinding. Nevertheless, the protocol can be readily used as a tool in a bigger protocol that is intended to be UC secure against malicious adversaries. Such a construction is already known from [DPSZ12].

We now explain how we arrive at our construction of the verifier's strategy for assigning elements to buckets: We define the buckets via a bipartite graph. Consider a finite, undirected, bipartite graph $G = (L, R, E)$ without multi-edges, where L denotes the set of vertices "on the left," R those "on the right" and E the set of edges. Write $n = |L|$ and $m = |R|$. Each vertex $w \in R$ on the right gives a "bucket of vertices" $N(\{w\}) \subset L$ on the left, where $N(\{w\})$ denotes the neighborhood of w.

We say that the bipartite graph G has the (f_1, f_2)-*strong unique neighbor property* if the following holds. For each set $N_1 \subset L$ with $|N_1| = f_1$, for each set $N_2 \subset R$ with $|N_2| = f_2$, and for each $i \in N_1$, there is $w \in R \setminus N_2$ such that $N_1 \cap N(\{w\}) = \{i\}$. Note that this property is anti-monotonous in the sense that if it holds for parameters (f_1, f_2) it also holds for parameters (f_1', f_2') with $f_1' \leq f_1$ and $f_2' \leq f_2$.

With f_1 corresponding to the failures in step 1 and f_2 corresponding to those in step 3, it should be clear that this property on (an infinite family of bipartite graphs) G, together with the conditions that $n = \mathrm{poly}(k)$, $m = O(n)$, $f_1 = O(k)$, $f_2 = O(k)$ and the condition that the right-degrees in G are all in $O(k)$, is sufficient to pull off our claimed result. Of course, in addition, this requires *efficient* construction of G.

Our pivotal observation for the construction of such bipartite graphs concerns a connection between our (f_1, f_2)-strong unique neighbor property and bipartite graphs with *excellent* expansion properties. Suppose the left-degrees in G are upper bounded by d. Then we say that a set $S \subset L$ expands ϵ-excellently if its neighborhood $N(S) \subset R$ satisfies $|N(S)| \geq (1-\epsilon)d|S|$ with $0 \leq \epsilon < 1/2$. Note that expansion in this regime is already known to imply the *unique neighbor property* for such an expanding set S, i.e., there exists a vertex $w \in R$ such that $|S \cap N(\{w\})| = 1$. Also note that $\epsilon < 1/2$ is essential for the implication. However, this implication is not sufficient for us. Instead, we show that G has the (f_1, f_2)-strong unique neighbor property if there is an ϵ such that $0 \leq \epsilon < 1/2$ and $2\epsilon f_1 + f_2 < d$ and such that each set $S \subset L$ of size f_1 expands ϵ-excellently. This generalization turns out to be sufficient for our purposes.

By the requirement that $f_1, f_2 \in O(k)$, it follows that the left-degrees of G cannot be upper bounded by a constant. Consequently, very well-known results on constant-degree lossless expanders [CRVW02] or unique-neighbor expanders [AC02] do not apply readily. [3]

Luckily, it turns out that a probabilistic construction of excellent bipartite expanders with parameters as required for our purposes can be based on the idea, originally due to Bassalygo [Bas81], of creating a bipartite expander graph from several random perfect-matchings. However, even though the probability of success is substantial (i.e., constant), this is certainly not large enough for our purposes.

The way out is that we can do with slightly weaker requirements. Namely, for our purposes, it suffices that there is an ϵ such that $0 \leq \epsilon < 1/2$ and $2\epsilon f_1 + f_2 < d$ and such that, for each *fixed* set $S \subset L$ of size f_1, the probability that S expands ϵ-excellently is exponentially close to 1. As we show, this is satisfied in our probabilistic approach. After an appropriate choice of parameters, we achieve $f_1 = c_1 k$ and $f_2 = c_2 k$ for arbitrary constants c_1, c_2, $n = m$ with $n \in O(k^3)$, and the right-degrees are each in $O(k)$.

Notation

Throughout this work we will format vectors such as $\boldsymbol{b}$ in lower-case bold face letters, whereas matrices such as $\boldsymbol{B}$ will be in upper case. We refer to the ith position of vector $\boldsymbol{b}$ as $\boldsymbol{b}[i]$, let $[r] := \{1, ..., r\}$ and define for $\boldsymbol{b} \in \mathbb{Z}^r$ that $||\boldsymbol{b}|| = \max_{i \in [r]}\{|\boldsymbol{b}[i]|\}$. To sample a variable g uniformly at random from a set G we use $g \xleftarrow{\$} G$. Throughout this work we will let λ be a computational and k be a statistical security parameter. Moreover, we use the standard definition for polynomial and negligible functions and denote those as $\mathsf{poly}(\cdot), \mathsf{negl}(\cdot)$.

[3] One may also derive excellent expansion from graphs with large girth. But this only applies to very small sets – too small for our purposes. Note that spectral properties of graphs do not guarantee excellent expansion. See [Kah92].

2 Homomorphic OWFs and Zero-Knowledge Proofs

We first define a primitive called *homomorphic one-way functions over integer vectors*. It is an extension of the standard definition of a OWF found in [KL14]. Let $\lambda \in \mathbb{N}$ be the security parameter, G be an Abelian group, $\beta, r \in \mathbb{N}$, $f : \mathbb{Z}^r \to G$ be a function and $\mathcal{A}$ be any algorithm. Consider the following game:

$\mathsf{Invert}_{\mathcal{A},f,\beta}(\lambda)$**:**

(1) Choose $\boldsymbol{x} \in \mathbb{Z}^r, ||\boldsymbol{x}|| \leq \beta$ and compute $y = f(\boldsymbol{x})$.
(2) On input $(1^\lambda, y)$ the algorithm $\mathcal{A}$ computes an $\boldsymbol{x}'$.
(3) Output 1 iff $f(\boldsymbol{x}') = y, ||\boldsymbol{x}'|| \leq \beta$, and 0 otherwise.

Definition 1 (Homomorphic OWF over Integer Vectors (ivOWF)). *A function $f : \mathbb{Z}^r \to G$ is called a homomorphic one-way function over the integers if the following conditions hold:*

(1) There exists a polynomial-time algorithm $eval_f$ such that $eval_f(\boldsymbol{x}) = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{Z}^r$.
(2) For all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{Z}^r$ it holds that $f(\boldsymbol{x}) + f(\boldsymbol{x}') = f(\boldsymbol{x} + \boldsymbol{x}')$.
(3) For every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr[\mathsf{Invert}_{\mathcal{A},f,\beta}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$$

As mentioned in the introduction, this abstraction captures, among other primitives, lattice-based encryption schemes such as [BGV12,GSW13,BV14] where the one-way property is implied by IND-CPA and β is as large as the plaintext space. Moreover it also captures hash functions such as [GGH96,LMPR08], where it is hard to find a preimage for all *sufficiently short* vectors that have norm smaller than β.

2.1 Proving Knowledge of Preimage

We consider two parties, the prover $\mathcal{P}$ and the verifier $\mathcal{V}$. $\mathcal{P}$ holds values $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathbb{Z}^r$, both parties have values $y_1, ..., y_n \in R$ and $\mathcal{P}$ wants to prove to $\mathcal{V}$ that $y_i = f(\boldsymbol{x}_i)$ and that $\boldsymbol{x}_i$ is *short*, while not giving no knowledge on the $\boldsymbol{x}_i$ away. More formally, the relation that we want to give a zero-knowledge proof of knowledge for is

$$R_{\text{KSP}} = \left\{ (v, w) \;\middle|\; \begin{array}{l} v = (y_1, ..., y_n) \wedge w = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \wedge \\ \left[y_i = f(\boldsymbol{x}_i) \wedge ||\boldsymbol{x}_i|| \leq \beta \right]_{i \in [n]} \end{array} \right\}$$

However, like all other protocols for this type of relation, we will have to live with a *soundness slack* τ as explained in the introduction. What this means more precisely is that there must exist a knowledge extractor with properties exactly as in the standard definition of knowledge soundness, but the extracted values only have to satisfy $[y_i = f(\boldsymbol{x}_i) \wedge ||\boldsymbol{x}_i|| \leq \tau \cdot \beta]_{i \in [n]}$.

3 Proofs of Preimage

3.1 Imperfect Proof of Knowledge

The first tool we need for our protocol is a subprotocol which we borrow from [BDLN16], a so-called *imperfect proof of knowledge*. This protocol is proof of knowledge for the above relation with a certain soundness slack, however, the knowledge extractor is only required to extract almost all preimages. We note that to show knowledge soundness later for our full protocol, Goldreich and Bellare [BG93] have shown that it is sufficient to consider deterministic provers, therefore we only need to consider deterministic provers in the following.

The idea for the protocol is that the prover constructs $T = 3n$ auxiliary values of form $z_i = f(\boldsymbol{r}_i)$ where $\boldsymbol{r}_i$ is random and short. The verifier asks the prover to open half the values (chosen at random) and aborts if the preimages received are not correct and short. One can show that this means the prover must know correct preimages of almost all the unopened values. The prover must now reveal, for each y_i in the input, a short preimage of the sum $y_i + z_j$ for some unopened z_j. By the homomorphic property of f this clearly means we can extract from the prover also a short preimage of most of the y_i's.

The reason one needs to have more than $2n$ auxiliary values is that the protocol makes use of Lyubashevsky's rejection sampling technique [Lyu08,Lyu09], where the prover is allowed to refuse to use some of the auxiliary values. This allows for a small soundness slack while still maintaining the zero-knowledge property. For technical reasons the use of rejection sampling means that the prover should not send the auxiliary values z_i in the clear at first but should commit to them, otherwise we cannot show zero-knowledge.

The following theorem is proved in [BDLN16] (their Theorem 1):

Theorem 1. *Let f be an ivOWF, k be a statistical security parameter, Assume we are given C_{aux}, a perfectly binding/computationally hiding commitment scheme over G, $\tau = 100 \cdot r$ and $T = 3 \cdot n, n \geq \max\{10, k\}$. Then there exists a protocol $\mathcal{P}_{\text{ImperfectProof}}$ with the following properties:*

Efficiency: *The protocol requires communication of at most T f-images and preimages.*

Correctness: *If $\mathcal{P}, \mathcal{V}$ are honest and run on an instance of R_{KSP}, then the protocol succeeds with probability at least $1 - \mathsf{negl}(k)$.*

Soundness: *For every deterministic prover $\hat{\mathcal{P}}$ that succeeds to run the protocol with probability $p > 2^{-k+1}$ one can extract at least $n - k$ values $\boldsymbol{x}'_i$ such that $f(\boldsymbol{x}'_i) = y_i$ and $||\boldsymbol{x}'_i|| \leq 2 \cdot \tau \cdot \beta$, in expected time $O(\mathsf{poly}(s) \cdot k^2/p)$ where s is the size of the input to the protocol.*

Zero-Knowledge: *The protocol is computational honest-verifier zero-knowledge.*

In the following we will use $\mathcal{P}_{\text{ImperfectProof}}(v, w, T, \tau, \beta)$ to denote an invocation of the protocol from this theorem with inputs $v = (y_1, ..., y_n), w = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ and parameters T, τ, β.

3.2 The Full Proof of Knowledge

The above imperfect protocol will be used as a building block. After executing it with the $(\boldsymbol{x}_i, y_i)$ as input, we may assume that a preimage of most of the y_i's (in fact, all but k) can be extracted from the prover.

The strategy for the last part of the protocol is as follows: we let the verifier randomly assign each y_i to one of several *buckets*. Then, for each bucket, we add all elements that landed in the bucket and have the prover demonstrate that he knows a preimage of the sum. The observation (made in [BDLN16]) is that we can now extract a preimage of every bad elements that is alone in a bucket. The question, however, is how we distribute items in buckets to maximize our chance of extracting all the missing preimages, and how many buckets we should use. One solution to this was given in [BDLN16], but it requires repeating the experiment $\log k$ times before all bad elements have been handled with good probability.

Here we propose a new strategy that achieves much better results: we need just one repetition of the game and each bucket will contain only $O(k)$ items which gives us the soundness slack of $O(k)$.

Before we can describe the protocol, we need to define a combinatorial object we use in the protocol, namely a good set system distribution:

Definition 2. *A set system distribution $\mathcal{D}$ with parameters n, m is a probability distribution that outputs m index sets $B_1, ..., B_m$, where each $B_j \subset [n]$, and $[n] = \{1, ..., n\}$. Both n and m depend on a security parameter k and we require that the distribution can be efficiently sampled (polynomial time in k). The set system distribution is* good *if the maximal size of a set B_j is $O(k)$, m is $O(n)$ and if for every set $N_1 \subset [n]$ of size k, the following event $Good_{N_1}$ occurs, except with probability at most $2^{-\Omega(k)}$. $Good_{N_1}$ occurs if after choosing the B_j's, for every set $N_2 \subset [m]$ of size k and every $i \in N_1$, there exists $j \in [m] - N_2$ such that $B_j \cap N_1 = \{i\}$.*

The idea in the definition is that $\mathcal{D}$ will describe the verifier's choice of buckets $\{B_j\}$. Then, if the distribution is good, and if we can extract preimage sums over all bucket except k, then we will be in business.

Theorem 2. *Let f be an ivOWF, k be a statistical security parameter, β be a given upper bound and $n \in \Theta(k^3)$. If $\mathcal{P}_{\text{CompleteProof}}$ is executed using a good set system distribution $\mathcal{D}$, then it is an interactive honest-verifier zero-knowledge proof of the relation R_{KSP} with knowledge error 2^{-k+1}. More specifically, it has the following properties:*

Efficiency *The protocol has overhead $O(1)$.*

Correctness: *If $\mathcal{P}, \mathcal{V}$ are honest then the protocol succeeds with probability at least $1 - 2^{-O(k)}$.*

Soundness: *For every deterministic prover $\hat{\mathcal{P}}$ that succeeds to run the protocol with probability $p > 2^{-k+1}$ one can extract n values $\boldsymbol{x}'_i$ such that $f(\boldsymbol{x}'_i) = y_i$ and $||\boldsymbol{x}'_i|| \leq O((k \cdot r \cdot \beta)$ except with negligible probability, in expected time $\mathsf{poly}(s, k)/p$, where s is the size of the input to the protocol.*

Procedure $\mathcal{P}_{\text{CompleteProof}}$

Let f be an ivOWF. $\mathcal{P}$ inputs w to the procedure and $\mathcal{V}$ inputs v. We assume that good set system distribution $\mathcal{D}$ is given with parameters n, m.

proof(v, w, β) :

(1) Let $v = (y_1, ..., y_n), w = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$. Run $\mathcal{P}_{\text{ImperfectProof}}(v, w, 3n, 100r, \beta)$. If $\mathcal{V}$ in $\mathcal{P}_{\text{ImperfectProof}}$ aborts then abort, otherwise continue.

(2) $\mathcal{V}$ chooses $B_1, ..., B_m$ according to $\mathcal{D}$ and sends the specification of the B_j's to $\mathcal{P}$.

(3) For $j = 1, \ldots, m$, both players compute $\gamma_j = \sum_{i \in B_j} v_i$ and $\mathcal{P}$ also computes $\boldsymbol{\delta}_j = \sum_{i \in B_j} \boldsymbol{x}_i$. Let h be the maximal size of a bucket set B_j, and set $\gamma = (\gamma_1, \ldots, \gamma_m)$, $\delta = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_m)$.

(4) Run $\mathcal{P}_{\text{ImperfectProof}}(\gamma, \delta, 3m, 100r, h\beta)$. If $\mathcal{V}$ in $\mathcal{P}_{\text{ImperfectProof}}$ aborts then abort, otherwise accept.

Fig. 1. A protocol to prove the relation R_{KSP}

Zero-Knowledge: *The protocol is computational honest-verifier zero-knowledge.*

Proof. Efficiency is immediate from Theorem 1 and the fact that we use a good set system, so that m is $O(n)$. Note also that the verifier can specify the set system for the prover using $O(m \cdot k \cdot \log n)$ bits. This will be dominated by the communication of m preimages if a preimage is larger than $k \log n$ bits, which will be the case for any realistic setting.
Correctness is immediate from correctness of $\mathcal{P}_{\text{ImperfectProof}}$.
The extractor required for knowlege soundness will simply run the extractor for $\mathcal{P}_{\text{ImperfectProof}}$ twice, corresponding to the 2 invocations of $\mathcal{P}_{\text{ImperfectProof}}$. Let N_1 be the set of k preimages we fail to extract in the first invocation, and let N_2 be the set of bucket sums we fail to extract in the second invocation. The properties of a good set system distribution now guarantee that no matter what set N_2 turns out to be, we can find, for each $i \in N_1$, a set B_j where we know a preimage of the sum over the bucket ($j \in [m] - N_2$), and furthermore $B_j \cap N_1 = \{i\}$. Concretely, we know $\boldsymbol{\delta}_j$ such that $f(\boldsymbol{\delta}_j) = \sum_{l \in B_j} y_l$ and we know preimages of all summands except for y_i. By the homomorphic property of f we can solve for a preimages of y_i, and the size of the preimage found follows immediately from Theorem 1 and the fact that buckets have size $O(k)$.
Honest-verifier zero-knowledge follows immediately from Theorem 1. We do the simulation by first invoking the simulator $\mathcal{P}_{\text{ImperfectProof}}$ with the input parameters for the first step. We then sample according to $\mathcal{D}$, compute the inout parameters for the second invocation and run the simulator for $\mathcal{P}_{\text{ImperfectProof}}$ again. □

To make this theorem be useful, we need of course that good set system distributions exist. This is taken care of in the following theorem.

Theorem 3. *Good set system distributions exist with parameters $n, m \in O(k^3)$.*

4 Proof of Theorem 3

Let $G = (L, R, V)$ be a finite, undirected bipartite graph. For simplicity we also assume G has no multi-edges. Here, L denotes the set of vertices "on the left," R the set of vertices "on the right" and V the set of edges.

Suppose $S \subset L$ is nonempty. The *neighborhood* of S is denoted $N(S)$. Note that $N(S) \subset R$ since G is bipartite. The neighborhood $N(T) \subset L$ of a nonempty set $T \subset R$ is defined similarly. The *unique neighbor set* $U(S) \subset R$ of the set $S \subset L$ consists of all $w \in R$ such that

$$|N(\{w\}) \cap S| = 1.$$

In general, $U(S)$ may be empty. We may similarly define $U(T)$ for nonempty $T \subset R$, but we will not need this.

We say that graph G is *d-left-bounded* if, for each $v \in L$, it holds that $|N(\{v\})| \leq d$. Similarly for *d'-right-bounded*. The graph G is *(d, d')-bi-bounded* if it is d-left-bounded and d'-right-bounded.

Write $n = |L|$ and $m = |R|$. Let d be an integer with $1 \leq d \leq n$ and let δ, ϵ be real numbers with $0 \leq \delta, \epsilon \leq 1$. Let $S \subset L$ be nonempty. Let f_1 be an integer with $1 \leq f_1 \leq n$ and let f_2, f be integers with $0 \leq f_2, f \leq m$.

Definition 3 (Unique Neighbor Property). *The set S has the* unique neighbor property *if it holds that $U(S) \neq \emptyset$.*

Definition 4 (Strong Unique Neighbor Property of a Set). *The set S has the* strong unique neighbor property *if, for each $i \in S$, there is $w \in R$ such that $N(\{w\}) \cap S = \{i\}$.*

Definition 5 (f-Strong Unique Neighbor Property of a Set). *The set S has the* f-strong unique neighbor property *if, for each $i \in S$ and for each $Z \subset R$ with $|Z| = f$, there is $w \in R \setminus Z$ such that $N(\{w\}) \cap S = \{i\}$.*

The unique neighbor property has been widely considered before and it has many known applications. We are presently not aware of previous consideration of our f-strong unique neighbor property even though it is a very natural extension.

Definition 6 ((f_1, f_2)-Strong Unique Neighbor Property of a Graph G). *The bipartite graph $G = (L, R, E)$ has the* (f_1, f_2)-strong unique neighbor property *if each set $S \subset L$ with $|S| = f_1$ has the f_2-strong unique neighbor property.*

The following lemma is well-known.

Lemma 1. *Suppose G is d-left-bounded. If $N(S) \geq (1 - \epsilon)d|S|$, then $|U(S)| \geq (1 - 2\epsilon)d|S|$.*

Proof. Since G is d-left-bounded, there are at most $d|S|$ edges "emanating" from S and "arriving" at $N(S)$. Write m_1 for the number of vertices $w \in N(S)$ with $|S \cap N(\{w\})| = 1$. Then we have the obvious bound

$$m_1 + 2(|N(S)| - m_1) \leq d|S|.$$

Therefore,

$$m_1 \geq 2|N(S)| - d|S|.$$

Since $|N(S)| \geq (1-\epsilon)d|S|$, it follows that

$$m_1 \geq (1-2\epsilon)d|S|,$$

as desired. △

Using a "greedy argument" we now show how the f-strong unique neighbor property for a set is implied by a large unique neighbor set.

Lemma 2. *Suppose G is d-left-bounded. If*

(1) $|U(S)| \geq (1-\delta)d|S|$ *and*
(2) $|S| < \frac{1}{\delta}(1 - \frac{f}{d})$

then S has the f-strong unique neighbor property. As this property is anti-monotone, it then also holds for all nonempty subsets $S' \subset S$.

Proof. Since

$$|U(S)| \geq (1-\delta)d|S|,$$

it follows, by a pigeonhole argument, that, if

$$\frac{(1-\delta)d|S| - f}{|S| - 1} > d,$$

the set S has the f-strong unique neighbor property. Indeed, if the property would fail on some $i \in S$, the inequality implies that there is some element in $S \setminus \{i\}$ with degree greater than d, which contradicts the fact that G is d-left-bounded. Note that the previous inequality simplifies to $|S| < 1/\delta(1 - f/d)$. △

By combining Lemmas 1 and 2 we get the following sufficient condition for the f-strong unique neighbor property of a set $S \subset L$.

Corollary 1. *Suppose G is d-left-bounded. If*

(1) $N(S) \geq (1-\epsilon)d|S|$ *and*
(2) $|S| < \frac{1}{2\epsilon}(1 - \frac{f}{d})$,

then S (and all of its nonempty subsets) have the f-strong unique neighbor property.

Remark 1. For this to be nontrivial, it is necessary that $\epsilon < 1/2$ and $f < d$.

We now give a probabilistic construction. Suppose $|L| = |R| = n$. Write $L = \{v_1, \ldots, v_n\}$ and $R = \{w_1, \ldots, w_n\}$. For a permutation π on $\{1, \ldots, n\}$, define $E(\pi) \subset L \times R$ as the set of edges

$$\{(v_1, w_{\pi(1)}), \ldots, (v_n, w_{\pi(n)})\}.$$

Suppose $1 \leq d \leq n$. For a d-vector $\Pi = (\pi_1, \ldots, \pi_d)$ of (not-necessarily distinct) permutations on $\{1, \ldots, n\}$, define the set

$$E(\Pi) = \bigcup_{j=1}^{d} E(\pi_j) \subset L \times R$$

and define the bipartite graph

$$G(\Pi) = (L, R, E(\Pi)).$$

Note that G is a (d, d)-bi-bounded (undirected) bipartite graph (without multi-edges). We have the following lemma.

Proposition 1. *Let $G = (L, R, E)$ be a random (d, d)-bi-bounded bipartite graph with $|L| = |R| = n$ as described above. Let α be a real number with $0 < \alpha < 1$. Then, for any* **fixed** *set $S \subset L$ with $|S| = \alpha n$, it holds that*

$$N(S) \geq (d-2)|S|,$$

except with probability

$$p'_S \leq \left(\frac{d^2 \alpha e}{2(1-\alpha)} \right)^{2\alpha n},$$

where e denotes Euler's constant.

Proof. Choose the d permutations $\pi_1, \ldots, \pi_d$ sequentially. For convenience, write $S = \{1, \ldots, s\}$. For $i = 1, \ldots, s$ and $j = 1, \ldots, d$, consider the random variables

$$X_i^j,$$

the image of $i \in S$ under the permutation π_j. We now think of these as “ordered” $X_1^1, \ldots, X_s^1, X_1^2, \ldots, X_s^2, \ldots$, “increasing” from left to right.
For given X_i^j, condition on all “prior” random variables in the ordering. The probability that X_i^j is a *repeat*, i.e., it lands in what is $N(S)$-so-far is at most

$$\frac{d|S|}{n-i+1} \leq \frac{d|S|}{n-|S|}.$$

Here the denominator on the LHS is due to the fact that when choosing the image of i, the $i-1$ distinct images of $1, \ldots, i-1$ are already taken. Hence, the probability p'_S that the event $|N(S)| \leq (d-2)|S|$ occurs is at most the probability of the event that there are $2|S|$ repeats. By the union bound, the latter probability is clearly at most

$$\binom{d|S|}{2|S|}\left(\frac{d|S|}{n-|S|}\right)^{2|S|}$$

Therefore, [4]

$$p'_S \leq \binom{d|S|}{2|S|}\left(\frac{d|S|}{n-|S|}\right)^{2|S|} \leq \left(\frac{de}{2}\right)^{2|S|}\left(\frac{d|S|}{n-|S|}\right)^{2|S|} = \left(\frac{d^2\alpha e}{2(1-\alpha)}\right)^{2\alpha n}.$$

$\triangle$

The lemma and its proof are adapted from an expander graph construction due to Bassalygo [Bas81]. Our exposition follows (part of) the proof of Theorem 4.4 in Salil Vadhan's textbook on Pseudorandomness [Vad12]. The reason we do not apply the Bassalygo result directly is that the success probability of the construction of an excellent expander is high (i.e., constant) but still much too small for our purposes. Fortunately, we can do with the slightly weaker requirement on G that, for any *fixed* set S of precisely the dictated size, the probability that the set S does not expand excellently is negligibly small. As this saves two applications of the union bound, one to quantify over all sets S of the dictated size and one to quantify over the subsets of size smaller than the dictated size, we get exponentially small failure probability instead of constant.

In conclusion, combining Proposition 1 and Corollary 1, we obtain the following result on set systems suitable for our purposes.

Theorem 4. *Let k be the security parameter. Let c_1, c_2 be arbitrary positive integers. Set*

(1) $f_1 = c_1 k,\ f_2 = c_2 k.$
(2) $d = c_3 k$ *with* $c_3 = c_1 + c_2 + 1.$
(3) $\alpha = \frac{1}{d^2 e+1}.$
(4) $n = m = \frac{c_1}{\alpha}k = (d^2 e + 1)c_1 k = (c_3^2 e k^2 + 1)c_1 k = c_1 c_3^2 e k^3 + c_1 k.$

Then there is good set system distribution with parameters as above and with error probability p' satisfying

$$p' \leq \left(\frac{1}{2}\right)^{2c_1 k}$$

Proof. For each set S of size $K = \alpha n = c_1 k = f_1$, it holds that $N(S) \geq (d-2)|S|$. Note that $\epsilon = 2/d$ here. This means that the second condition for the f_2-strong unique neighbor property of sets of this size is $f_1 + f_2 < d$. This is satisfied by definition. Note that efficiency of the construction is obvious. $\triangle$

This proves Theorem 3 - note that f_1, f_2 are upper bounds on the sizes of the "failure sets" N_1, N_2 from Definition 2, so this result is in fact more general because any constant times k is allowed. We get Theorem 3 by setting $c_1 = c_2 = 1$.

[4] Note that $\left(\frac{r}{s}\right)^s \leq \binom{r}{s} \leq \left(\frac{re}{s}\right)^s$.

5 Acknowledgements

We are grateful to Gilles Zémor for helpful discussions.

References

AC02. Noga Alon and Michael Capalbo. Explicit unique-neighbor expanders. In *FOCS*, pages 73–, 2002.

Bas81. Leonid Bassalygo. Asymptotically optimal switching circuits. *Problems in Information Transmission*, 17(3):81–88, 1981.

BDLN16. Carsten Baum, Ivan Damgard, Kasper Larsen, and Michael Nielsen. How to prove knowledge of small secrets. Cryptology ePrint Archive, Report 2016/538, 2016. To appear in Crypto 2016.

BDOZ11. Rikke Bendlin, Ivan Damgård, Claudio Orlandi, and Sarah Zakarias. Semi-homomorphic encryption and multiparty computation. In *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 169–188. Springer Berlin Heidelberg, 2011.

BG93. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology—CRYPTO'92*, pages 390–420. Springer, 1993.

BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

BV14. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

CD09. Ronald Cramer and Ivan Damgård. On the amortized complexity of zero-knowledge protocols. In *Advances in Cryptology-CRYPTO 2009*, pages 177–191. Springer, 2009.

CRVW02. Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *STOC*, pages 659–668, 2002.

DF02. Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *Advances in Cryptology—ASIACRYPT 2002*, pages 125–142. Springer, 2002.

DKL$^+$13. Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel Smart. Practical covertly secure mpc for dishonest majority - or: Breaking the spdz limits. In *ESORICS*, pages 1–18, 2013.

DPSZ12. Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 643–662. Springer, Berlin, Germany, 2012.

GGH96. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 3, pages 236–241, 1996.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*, pages 75–92. Springer, 2013.

Kah92. Nabil Kahale. On the second eigenvalue and linear expansion of regular graphs. In *FOCS*, pages 296–303, 1992.

KL14. Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC Press, 2014.

LMPR08. Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. Swifft: A modest proposal for fft hashing. In *Fast Software Encryption*, pages 54–72. Springer, 2008.

Lyu08. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography–PKC 2008*, pages 162–179. Springer, 2008.

Lyu09. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology–ASIACRYPT 2009*, pages 598–616. Springer, 2009.

Vad12. Salil Vadhan. *Pseudorandomness*. Now publishers, 2012.