There Are 10 Types of Vectors (and Polynomials)

Efficient Zero-Knowledge Proofs of "One-Hotness" via Polynomials with One Zero

William Black
University of Calgary
william.black@ucalgary.ca

Ryan Henry
University of Calgary
ryan.henry@ucalgary.ca

ABSTRACT

We present a new 4-move special honest-verifier zero-knowledge proof of knowledge system for proving that a vector of Pedersen commitments opens to a so-called "one-hot" vector (i.e., to a vector from the standard orthonormal basis) from $\mathbb{Z}_p^n$. The need for such proofs arises in the contexts of symmetric private information retrieval (SPIR), end-to-end verifiable voting (E2E), and privacy-preserving data aggregation and analytics, among others. The key insight underlying the new protocol is a simple observation regarding the paucity of roots of polynomials of bounded degree over a finite field. The new protocol is *fast* and yields *succinct* proofs: For vectors of length n, the prover evaluates $\Theta(\lg n)$ group operations plus $\Theta(n)$ field operations and sends just $\Theta(\lg n)$ group and field elements, while the verifier evaluates one n-base multiexponentiation plus $\Theta(\lg n)$ additional group operations and sends just $2(\lambda + \lg n)$ bits to obtain a soundness error less than $2^{-\lambda}$. (A 5-move variant of the protocol reduces prover upload to just $2\lambda + \lg n$ bits for the same soundness error.) We have implemented both our new protocol and its closest competitors from the literature; in accordance with our analytic results, experiments confirm that the new protocols handily outperform existing protocols for all but the shortest of vectors (roughly, for vectors with more than 16–32 elements).

CCS CONCEPTS

• **Security and privacy → Cryptography**; **Privacy-preserving protocols**; **Security protocols**;

KEYWORDS

Zero-knowledge proofs; efficiency; privacy-preserving protocols

1 INTRODUCTION

Zero-knowledge proofs provide a means for a prover to convince a verifier that some claim is true while communicating *nothing more*. The ability to prove statements while conveying zero information beyond their veracity has profound implications for cryptography and, especially, for its applications to the construction of privacy-enhancing technologies (PETs). Unfortunately, zero-knowledge does not come free and most of the common zero-knowledge techniques in the literature suffer poor scalability when applied to statements with "high fan-in", thus limiting their usefulness in many otherwise-promising application domains.

This paper addresses the problem of designing communication- and computation-efficient zero-knowledge proofs of knowledge with which a prover P can convince a verifier V of its ability to open a vector of Pedersen (or Pedersen-like) commitments to a so-called "one-hot" vector (i.e., to a vector from the standard orthonormal basis) over some prime-order field. The need for such proofs naturally arises in several contexts. For instance, Henry, Olumofin, and Goldberg [19] and, more recently, Damgård, Luo, Oechsner, Scholl, and Simkin [13] have observed that such proofs give rise to a generic (and rather elegant) transformation from "vector-matrix" style *private information retrieval (PIR)* to *symmetric PIR (SPIR)*, a strengthened variant of PIR which extend privacy protections beyond the queriers to also include the database holders. Similarly, schemes that rely on "vector-matrix" style *PIR-Writing* [7] with multiple clients—such as the Riposte system of Corrigan-Gibbs, Boneh, and Mazières [11] or the private ad-impression reporting protocol of Green, Ladd, and Miers [15]—require such proofs to ensure that malformed write requests from malicious clients cannot corrupt the entire database.

The focus of this paper is on the *design*, *formal analysis*, and *empirical evaluation* of an efficient new proof system for proving vectors of commitments open to standard basis vectors, as opposed to specific applications thereof; however, we hasten to note that our new proof system can serve as a drop-in replacement for those used in any of the above-mentioned systems, among myriad others. Moreover, our analytic and experimental results suggest that using our new proofs in place of existing proofs would yield increasingly dramatic speedups and bandwidth savings as the sizes the vectors in question grow large.

Related work. A handful of recent works have studied efficient zero-knowledge proofs pertaining to "high fan-in" statements. In Sections 2.1 and 2.2, we review a pair of protocols that tackle the same problem as this work (i.e., proving that a vector of commitments opens to a standard basis vector). The second of those protocols builds on a logarithmic-size zero-knowledge proof, due to Groth and Kohlweiss [16], for proving that (at-least-)1-out-of-n commitments opens to zero, which itself builds on another logarithmic-size zero-knowledge proof, due to Bayer and Groth [2], for proving that a commitment opens to some element of a publicly known set. The Groth-Kohlweiss proof was later improved by Bootle, Cerulli, Chaidos, Ghadafi, Groth, and Petit [8]. Relatedly, Brands, Demuynck, and De Decker [9] give a $\sqrt{n}$-size proof that a committed value does *not* open to any element of a publicly known set.

WPES'19, November 11, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6830-8/19/11…$15.00
https://doi.org/10.1145/3338498.3358640

In another line of work, Peng, Boyd, and Dawson [21] and Gennaro, Leigh, Sundaram, and Yerazunis [14] proposed two alternative ways to prove simultaneous knowledge of n discrete-log representations, while Henry and Goldberg [17] extended the former approach to allow proving knowledge of (at-least-)k-out-of-n discrete-log representations. Whereas the protocols described in Sections 2.1 and 2.2 follow the same "small-exponents" technique as Peng et al., our new protocol will instead follow the "polynomial-evaluation" technique of Gennaro et al.

Finally, we note that there has been a recent flurry of work on SNARKs and SNARGs [5], IOPs [4], and the like. These approaches yield *constant-size* proofs with very efficient verifiers—but at *considerable* computational expense for the prover, which effectively renders them inapplicable for the scale of problems we target.

2 NOTATION AND PRELIMINARIES

Throughout, we denote by p a "large" (think, 256- or thereabout-bit) prime number; by $\mathbb{Z}_p$ the field of integers modulo p; and by $\mathbb{Z}_p[x]$ the ring of polynomials over $\mathbb{Z}_p$. We further denote by $\mathbb{G}$ a fixed, cyclic group of order p (for which we use multiplicative notation) having two canonical generators that we denote by g and h. We implicitly assume that the *discrete logarithm problem (DLP)* is intractable in $\mathbb{G}$ and that the generators g and h are selected in such a way that $\log_g h \in \mathbb{Z}_p$ is a uniform random, unknown quantity. With that said, all of the results we present herein hold *unconditionally*: We do not actually rely on the DLP nor on any other unproven intractability assumptions.[1] Rather, it is the cryptographic systems that are likely to *use* our results which must rely on such assumptions for their security and privacy guarantees to hold.

Cryptographic commitments: Each of our protocols takes as input one or more *cryptographic commitments* to elements from $\mathbb{Z}_p$. In practice, any linearly homomorphic commitment scheme will do; however, for simplicity and concreteness, we assume throughout that all commitments are Pedersen commitments [20] constructed using the above-defined bases g and h. Specifically, a commitment to $a \in \mathbb{Z}_p$ has the form $C = g^a h^r$ for uniformly selected $r \in_{\mathrm{R}} \mathbb{Z}_p$. Throughout the paper, we speak colloquially of the ability of a party to *open* a Pedersen commitment C to some value $a \in \mathbb{Z}_p$, by which we mean that the party can (be modified to) output the unique $r \in \mathbb{Z}_p$ for which $C = g^a h^r$. When the DLP is intractable in $\mathbb{G}$, one can reasonably surmise that the ability of a party to open C to a implies that the party *cannot* also open C to a' for any $a' \not\equiv a \bmod p$, as doing so would be tantamount to computing $x \equiv \log_g h \bmod p$.

Standard basis vectors: Our main contribution is an efficient protocol with which a *prover* P can demonstrate to a *verifier* V its ability to open a length-n vector of Pedersen commitments, say $\vec{E} \coloneqq \langle E_1, \ldots, E_n \rangle$, component-wise to a *standard basis vector* from $\mathbb{Z}_p^n$; that is, to a length-n vector comprising a *single* one, along with $n-1$ zeros. We refer to the length-n standard basis vector having its one in its ℓth coordinate as the *ℓth standard basis vector* of $\mathbb{Z}_p^n$ and we denote it by $\vec{e}_\ell$. We write $\vec{v}[i]$ to refer to the ith component of a vector $\vec{v}$; thus, $\vec{e}_\ell[i] = 1$ if and only if $\ell = i$ and $\vec{e}_\ell[i] = 0$ otherwise.

[1] One pseudo-caveat is our efficiency analyses in Sections 2.1–2.3 and Section 5.1, which take for granted the presumed infeasibility of computing $x \equiv \log_g h \bmod p$ and therefore consider only calculations that do not require knowledge of such an x.

(Note that we adopt the notational conventions that (i) indexing into sequences and vectors is always 1-based, and (ii) the index i always ranges over $[1..n]$, while index j ranges over $[1..k]$ for $k \coloneqq \lceil \lg n \rceil$, and index l elsewhere.)

2.1 The Henry-Olumofin-Goldberg Proof

The earliest instance of a proof of knowledge system allowing prover P to convince verifier V of its ability to open a vector of commitments component-wise to a standard basis vector appears to be due to Henry, Olumofin, and Goldberg [19, §3.5.2].[2] As with the protocol proposed herein, their protocol is a 4-move special honest-verifier zero-knowledge proof of knowledge system. It is based on the following two-part observation:

(1) if $\vec{v} = \vec{e}_\ell$ is the ℓth standard basis vector and $\vec{R} \in \mathbb{Z}_p^n$ is an arbitrary vector, then $\vec{v} \cdot \vec{R} \equiv \vec{R}[\ell] \bmod p$; whereas,
(2) if $\vec{v} \neq \vec{e}_\ell$ is *not* the ℓth standard basis vector and $\vec{R} \in_{\mathrm{R}} \mathbb{Z}_p^n$ is selected uniformly at random, then $\vec{v} \cdot \vec{R} \not\equiv \vec{R}[\ell] \bmod p$, except with a probability negligible in $\lg p$.

The above observation effectively reduces the problem of proving that $\vec{v}$ is a standard basis vector to that of proving that the scalar $\vec{v} \cdot \vec{R}$ is an element of the set $\{\vec{R}[1], \ldots, \vec{R}[n]\}$, which Henry et al. implement as a straightforward disjunctive ("OR") proof of partial knowledge using the standard techniques first proposed by Cramer, Damgård, and Schoenmakers [12].

In particular, at the outset of Henry et al.'s protocol, V chooses $\vec{R} \in_{\mathrm{R}} \mathbb{Z}_p^n$, and then it (i) computes

$$\begin{aligned}
\tilde{\vec{E}}(\vec{R}) &\coloneqq \textstyle\prod_{i=1}^{n} (E_i)^{\vec{R}[i]} \\
&= \textstyle\prod_{i=1}^{n} \left(g^{\vec{v}[i]} h^{r_i}\right)^{\vec{R}[i]} \\
&= \textstyle\prod_{i=1}^{n} g^{\vec{v}[i]\cdot\vec{R}[i]} h^{r_i \cdot \vec{R}[i]} \\
&= g^{\left(\sum_{i=1}^{n} \vec{v}[i]\cdot\vec{R}[i]\right)} h^{\left(\sum_{i=1}^{n} r_i \cdot \vec{R}[i]\right)} \\
&= g^{\vec{v}\cdot\vec{R}} h^{\left(\sum_{i=1}^{n} r_i \cdot \vec{R}[i]\right)} \\
&= g^{\vec{R}[\ell]} h^{\left(\sum_{i=1}^{n} r_i \cdot \vec{R}[i]\right)}
\end{aligned}$$

and (ii) sends $\vec{R}$ to P. From here, P engages V in the Σ-protocol denoted in Camenisch-Stadler notation [10] by

$$\mathrm{PK}\left\{ r : \textstyle\bigvee_{i=1}^{n} \left((\tilde{\vec{E}}(\vec{R}) / g^{\vec{R}[i]}) = h^r \right) \right\}$$

using $r \leftarrow \sum_{i=1}^{n} (r_i \cdot \vec{R}[i]) \bmod p$.

Both parties in this protocol have computation cost linear in the length n of $\vec{E}$: V evaluates an n-base multiexponentiation and an additional n exponentiation in $\mathbb{G}$ to produce the sequence of $\tilde{\vec{E}}(\vec{R}) / g^{\vec{R}[i]}$, plus a further $\Theta(n)$ 2-base multiexponentiations in $\mathbb{G}$ as part of the Σ-protocol; meanwhile, P evaluates $2n-1$ operations in $\mathbb{Z}_p$ to compute the witness r, plus a further $\Theta(n)$ exponentiations in $\mathbb{G}$ and operations in $\mathbb{Z}_p$ as part of the Σ-protocol. Similarly, both parties have upload cost linear in n: V sends the vector $\vec{R} \in \mathbb{Z}_p^n$ in the

[2] Shortly thereafter [18, Appendix], Henry, Huang, and Goldberg generalized Henry et al.'s protocol from a system allowing P to convince V of its ability to open a vector of commitments component-wise to a standard basis vector of $\mathbb{Z}_p^n$ into a system allowing P to convince V of its ability to open a vector of multi-secret commitments to a "rectangular permutation matrix" from $\mathbb{Z}_p^{m \times n}$ in which each individual row is a distinct standard basis vector of $\mathbb{Z}_p^n$.

first step, plus a single challenge from $\mathbb{Z}_p$ as part of the Σ-protocol; meanwhile, P sends n elements of $\mathbb{G}$ and a further n challenge-response pairs from $\mathbb{Z}_p \times \mathbb{Z}_p$ as part of the Σ-protocol.

2.2 The Groth-Kohlweiss Improvement

More recently, Groth and Kohlweiss [16, Figure 2] proposed a communication-efficient Σ-protocol allowing prover P to convince verifier V of its ability to open a commitment to some element of a fixed, finite set S. In contrast to the *linear* disjunctive proof of knowledge employed by Henry et al., Groth and Kohlweiss' proofs are *concise*: the entire (set membership sub-)protocol exhibits communication cost *logarithmic* in the cardinality of S. As observed by Green, Ladd, and Miers [15, §1.2 and §5], merely swapping Groth and Kohlweiss' protocol into the framework of Henry et al. yields a 4-move special honest-verifier zero-knowledge proof of knowledge system with notably reduced communication and computation costs—particularly when n is "large"—relative to the linear disjunctive proof. Nevertheless, with this approach, V must still sample and send to P the uniform random vector $\vec{R} \in \mathbb{Z}_p^n$ with which to define $\widetilde{\boldsymbol{E}}(\vec{R})$ and r prior to invoking Groth and Kohlweiss' protocol; thus, the overall $\Theta(n)$ communication cost of Henry et al.'s protocol persists, albeit with significantly smaller constants hidden within the big-Θ notation. (Looking ahead, we empirically measure the improvements relative to the linear disjunctive proof in Section 6.)

At the heart of Groth and Kohlweiss' protocol is a simple observation regarding the *Kronecker delta function* $\delta\colon [0\,..\,2^k-1] \times [0\,..\,2^k-1] \to \{0,1\}$, defined via $\delta(\ell,i)=1$ if $i=\ell$ and $\delta(\ell,i)=0$ otherwise; namely, that

$$\delta(\ell,i) = \prod_{j=1}^{k} \delta(\ell_j, i_j),$$

where ℓ_j and i_j respectively denote the jth-least-significant bits in the (k-bit) binary representations of ℓ and i.

To prove that $\widetilde{\boldsymbol{E}}(\vec{R})$ commits to some $\vec{R}[\ell] \in \{\vec{R}[1],\ldots,\vec{R}[n]\}$, P first commits to each bit ℓ_j in binary representation of ℓ, and then it constructs a specific length-n sequence of polynomials $f_1,\ldots,f_n \in \mathbb{Z}_p[x]$, with each f_i being defined in terms of (i) the bits ℓ_j of the target set element's index ℓ, (ii) the bits i_j of the polynomial's index i, and (iii) the randomness in the commitments to the ℓ_j just constructed. The polynomials f_i are defined in such a way that—owing to the above observation about $\delta(\ell,i)$—we have $\deg f_i = k$ if $i=\ell$ and $\deg f_i < k$ otherwise. (Recall that $k \coloneqq \lceil \lg n \rceil$ is the bitlength of ℓ and i.) The remainder of the protocol relates the unique degree-k monomial among the f_i to the position of an element within the set $\{\vec{R}[1],\ldots,\vec{R}[n]\}$. We refer the reader to Groth and Kohlweiss' paper [16, §3] for details of how this all works; for our purposes, it suffices to note that their approach requires P to evaluate $\Theta(n \ln n)$ operations in $\mathbb{Z}_p$ in order to construct and evalulate the requisite sequence of polynomials.

Thus, both parties in the resulting protocol have computation cost that is (at least) linear in n: V still evaluates an n-base multiexponentiation and n exponentiations in $\mathbb{G}$ to produce the sequence of $\widetilde{\boldsymbol{E}}(\vec{R})/g^{\vec{R}[i]}$, plus a further $\Theta(\lg n)$ operations in $\mathbb{G}$ and $\mathbb{Z}_p$ as part of the Σ-protocol; meanwhile, P still evaluates $2n-1$ operations in $\mathbb{Z}_p$ to compute the witness r, plus a further $\Theta(n \lg n)$ operations in $\mathbb{Z}_p$ and $\Theta(\lg n)$ operations in $\mathbb{G}$ as part of the Σ-protocol. However, now it is only V that has upload cost linear in n, while P's has been reduced to logarithmic in n: V still sends the vector $\vec{R} \in \mathbb{Z}_p^n$ in the first step, plus a single challenge from $\mathbb{Z}_p$ as part of the Σ-protocol; meanwhile, P sends just $4\lceil \lg n \rceil$ elements from $\mathbb{G}$ and and $3\lceil \lg n \rceil - 1$ elements from $\mathbb{Z}_p$ as part of the Σ-protocol—a *dramatic* improvement over the linear disjunctive proof.

2.3 The new protocol

Our new protocol tackles the problem of allowing prover P to convince verifier V of its ability to open a vector of commitments to a standard basis vector "head on", without reducing the problem to that of proving set membership. In doing so, we are able to eliminate the random vector $\vec{R} \in \mathbb{Z}_p^n$ and, thereby, obtain a protocol with only *logarithmic communication in both directions* (indeed, as we will see, it is natural to treat V's upload as constant).

The new protocol is based on the following two-part observation, which is reminiscent of Henry et al.'s observation as stated in Section 2.1, just with the vector $\vec{R} \in_{\mathrm{R}} \mathbb{Z}_p^n$ replaced by a vector of the form $\langle 1, t, t^2, \ldots, t^{n-1}\rangle$ for some $t \in_{\mathrm{R}} \mathbb{Z}_p$:

(1) if $\vec{v} = \vec{\boldsymbol{e}}_\ell$ is the ℓth standard basis vector and $t \in \mathbb{Z}_p$, then $\sum_{i=1}^{n} \vec{v}[i] \cdot t^{i-1} \equiv t^{\ell-1} \bmod p$; whereas,
(2) if $\vec{v} \neq \vec{\boldsymbol{e}}_\ell$ is *not* the ℓth standard basis vector and $t \in_{\mathrm{R}} \mathbb{Z}_p$ is selected uniformly at random, then $\sum_{i=1}^{n} \vec{v}[i] \cdot t^{i-1} \not\equiv t^{\ell-1} \bmod p$, except with negligible probability.

Indeed, $\sum_{i=1}^{n}(\vec{v}[i] \cdot t^{i-1}) \equiv t^{\ell-1} \bmod p$ if and only if t is a root of the polynomial $V_\ell(x) \coloneqq \left(\sum_{i=1}^{n}(\vec{v}[i] \cdot x^{i-1})\right) - x^{\ell-1} \in \mathbb{Z}_p[x]$, which is exceedingly unlikely for uniformly selected $t \in_{\mathrm{R}} \mathbb{Z}_p$ unless $V_\ell(x) = 0$ is the zero polynomial (i.e., unless $\vec{v} = \vec{\boldsymbol{e}}_\ell$). We provide much-needed rigor to the above argument in Section 3, specifically in Corollary 3.4 and the pair of theorems that precede it.

The above observation effectively reduces the problem of proving that $\vec{v}$ is a standard basis vector to that of proving that $V(x) \coloneqq \sum_{i=1}^{n}(\vec{v}[i] \cdot x^{i-1})$ is a monic monomial of degree less than n.[3] To this end, we propose an efficient Σ-protocol for proving knowledge of a so-called *double discrete logarithm (double DL)* with a publicly known base and a publicly known upper bound on the bitlength of the secret exponent.

More concretely, at the outset of our protocol, V chooses $t \in_{\mathrm{R}} \mathbb{Z}_p$, and then it (i) computes the vector $\vec{t} \coloneqq \langle 1, t, t^2, \ldots, t^{n-1}\rangle \in \mathbb{Z}_p^n$, (ii) uses $\vec{t}$ to compute

$$\begin{aligned}
\widetilde{\boldsymbol{E}}(t) &\coloneqq \textstyle\prod_{i=1}^{n}(E_i)^{t^{i-1}} \\
&= \textstyle\prod_{i=1}^{n}\left(g^{\vec{v}[i]}h^{r_i}\right)^{t^{i-1}} \\
&= \textstyle\prod_{i=1}^{n}\left(g^{\vec{v}[i]\cdot t^{i-1}}h^{r_i \cdot t^{i-1}}\right) \\
&= g^{\left(\sum_{i=1}^{n}\vec{v}[i]\cdot t^{i-1}\right)}h^{\left(\sum_{i=1}^{n} r_i \cdot t^{i-1}\right)} \\
&= g^{\vec{v}\cdot\vec{t}}h^{\left(\sum_{i=1}^{n} r_i \cdot t^{i-1}\right)} \\
&= g^{t^{\ell-1}}h^{\left(\sum_{i=1}^{n} r_i \cdot t^{i-1}\right)}
\end{aligned}$$

as an n-base multiexponentiation, and (iii) sends $t \in \mathbb{Z}_p$ to P.

[3] In order to avoid costly range proofs, we relax the latter requirement and prove only that the degree of $V(x)$ is less than $2^{\lceil \lg n \rceil}$; our analysis in Section 3 establishes that this relaxation imposes an at-most-negligible penalty to the soundness of the protocol.

Table 1: Computation and communication complexities for the prover and verifier in the protocols of Henry, Olumofin, and Goldberg ("HenryOG11") [19]; its variant with Groth and Kohlweiss' improvement ("GrothK15") [16]; and the protocol proposed in this work ("This work"). The color underlining each protocol label references the corresponding plot color for that protocol in Figures 1 and 2 (in Section 6). In each column, the lowest-cost cells are shaded green, whilst the highest-cost cells are shaded red; when all costs in a column are asymptotically equal, the cells are shaded yellow.

	Prover						Verifier					
	Computation			Communication			Computation					Communication
	Exps.		$\mathbb{Z}_p$	$\mathbb{G}$		$\mathbb{Z}_p$	Exps.		n-Multiexps.		$\mathbb{Z}_p$	$\mathbb{Z}_p$
"HenryOG11"	$\Theta(n)$	+	$\Theta(n)$	$\Theta(n)$	+	$\Theta(n)$	$\Theta(n)$	+	$\Theta(1)$	+	$\Theta(n)$	$\Theta(n)$
"GrothK15"	$\Theta(\lg n)$	+	$\Theta(n \lg n)$	$\Theta(\lg n)$	+	$\Theta(\lg n)$	$\Theta(n)$	+	$\Theta(1)$	+	$\Theta(n)$	$\Theta(n)$
"This work"	$\Theta(\lg n)$	+	$\Theta(n)$	$\Theta(\lg n)$	+	$\Theta(\lg n)$	$\Theta(\lg n)$	+	$\Theta(1)$	+	$\Theta(n)$	$\Theta(1)$

From here, P engages V in the Σ-protocol denoted in Camenisch-Stadler notation [10] by

$$\mathrm{PK}\{(\ell, r) : C = g^{t^{\ell-1}} h^r \wedge \ell \in [1..2^k]\},$$

using $r \leftarrow \sum_{i=1}^{n} (r_i \cdot t^{i-1}) \bmod p$. An efficient instantiation of the latter Σ-protocol appears in Section 4.4.

The result is a 4-move special honest-verifier zero-knowledge proof of knowledge system that, relative to the protocols of Henry et al. (§2.1) and its variant with Groth and Kohlweiss' improvement (§2.2), exhibits superior asymptotics *and* superior concrete costs for all but the shortest of vectors. Specifically, both parties in the resulting protocol have computation cost that is linear in n: V computes powers of t using $n-2$ multiplications in $\mathbb{Z}_p$ and then it evaluates an n-base multiexponentiation in $\mathbb{G}$, plus a further $\Theta(\lg n)$ operations in each of $\mathbb{G}$ and $\mathbb{Z}_p$ as part of the Σ-protocol; meanwhile, P evaluates $2n-1$ operations $\mathbb{Z}_p$ to compute its witness r using Horner's method, plus a further $\Theta(\lg n)$ operations in each of $\mathbb{G}$ and $\mathbb{Z}_p$ as part of the Σ-protocol. However, now both parties have upload cost (at most) logarithmic in n: V sends $t \in \mathbb{Z}_p$, plus a single challenge from $\mathbb{Z}_p$ as part of the Σ-protocol; meanwhile, P sends just $5\lceil \lg n \rceil - 3$ elements from $\mathbb{G}$ and $4\lceil \lg n \rceil - 2$ elements from $\mathbb{Z}_p$ as part of the Σ-protocol.

Table 1 summarizes the asymptotic costs of the three protocols just described in Sections 2.1–2.3. Notice that the new protocol is (i) at least as efficient (asymptotically) as the other two protocols in every column, and (ii) more efficient than than at least one other protocol in all but two columns. An empirical, head-to-head evaluation of the concrete performance of all three protocols appears later on, in Section 6. That evaluation confirms that the superior asymptotics of our new protocol, as summarized in Table 1, equate to (rather dramatic) across-the-board speedups relative to the other two protocols.

3 ROOTS OF POLYNOMIALS

This section proves some elementary results about the (maximum possible number of) roots of certain collections of polynomials arising in the analysis of our protocol. The results established in this section are neither surprising nor are their precise statements (or proofs of those statements) especially crucial for readers who seek an intuitive understanding of our new protocol; nonetheless, these facts *do* play a sufficiently central role in the construction as to warrant prominent inclusion in the main body of the text. With that said, readers who are willing to accept our claims of soundness at face value may safely skip to Section 4.

Theorem 3.1 (*"Tight-range" bound*). *Fix a positive integer n. Let p an odd prime and let $V \in \mathbb{Z}_p[x]$ such that (i) $\deg V < n$ and (ii) V is not a (monic) monomial. Then the n polynomials in $P = \{V - x^0, V - x^1, \ldots, V - x^{n-1}\}$ can have at most $n \cdot (n-1)$ distinct roots in total between them.*

Proof. Let $h \in P$, say, $h(x) := V(x) - x^{i-1}$ for some $i \in [1..n]$. Then, since $\deg V < n$ and V is not a monic monomial, we have that $h \neq 0$ so that $\deg h = \max(\deg V, i-1) < n$ and, consequently, that h has at most $n-1$ distinct roots in $\mathbb{Z}_p$. Moreover, as h was selected arbitrarily from a set of n possibilities, it follows by the union bound that the total number of distinct roots of polynomials in P is at most $n \cdot (n-1)$, as desired. □

Theorem 3.2 (*"Loose-range" bound*). *Fix positive integers k and n such that $n \leq 2^k$. Let p be an odd prime and let $V \in \mathbb{Z}_p[x]$ such that (i) $\deg V < n$ and (ii) V is not a (monic) monomial. Then the 2^k polynomials in $P = \{V - x^0, V - x^1, \ldots, V - x^{n-1}, V - x^n, \ldots, V - x^{2^k-1}\}$ can have at most $2^{2k-1} - 2^{k-1} + \frac{n \cdot (n-1)}{2}$ distinct roots in total between them.*

Proof. Write $P = P_0 \cup P_1$ where $P_0 = \{V - x^0, V - x^1, \ldots, V - x^{n-1}\}$ and $P_1 = \{V - x^n, V - x^{n+1}, \ldots, V - x^{2^k-1}\}$ are disjoint sets. By Theorem 3.1, the polynomials in P_0 contribute at most $n \cdot (n-1)$ distinct roots. As for the polynomials in P_1, it suffices to note that each $h_i := V(x) - x^{i-1} \in P_1$ has degree $\deg h_i = \max(\deg V, i-1) = i-1$ and, hence, at most $i-1$ distinct roots. It then follows by the union bound that the total number of distinct roots for polynomials in P_1 is at most

$$\left(\sum_{l=1}^{2k-1} l\right) - \left(\sum_{i=1}^{n} i\right) = \frac{(2^k - 1) \cdot (2^k - 1 + 1)}{2} - \frac{n \cdot (n-1)}{2} \tag{1}$$

$$= 2^{2k-1} - 2^{k-1} - \frac{n \cdot (n-1)}{2}. \tag{2}$$

Adding to Equation (2) the at most $n \cdot (n-1)$ roots contributed by P_0 yields a total of at most

$$2^{2k-1} - 2^{k-1} + \frac{n \cdot (n-1)}{2} \tag{3}$$

distinct roots for polynomials in $P = P_0 \cup P_1$, as desired. □

Corollary 3.3. *Fix positive integers k and n such that $n \leq 2^k$. Let p be an odd prime and let $V \in \mathbb{Z}_p[x]$ such that (i) $\deg V < n$ and (ii) V is not a (monic) monomial. Then the 2^k polynomials in $P = \{V - x^0, V - x^1, \ldots, V - x^{n-1}, V - x^n, \ldots, V - x^{2^k-1}\}$ have fewer than 2^{2k} distinct roots in total between them.*

Proof. Since $n \leq 2^k$, we have $\frac{n\cdot(n-1)}{2} \leq \frac{2^k\cdot(2^k-1)}{2} = 2^{2k-1} - 2^{k-1}$. Substituting into Equation (3) yields a total of at most

$$\left(2^{2k-1} - 2^{k-1}\right) + \left(2^{2k-1} - 2^{k-1}\right) = 2^{2k} - 2^k < 2^{2k},$$

distinct roots for polynomial in P, as desired. □

We remark (without proof) that it is relatively easy to produce examples of polynomials $V \in \mathbb{Z}_p[x]$ to demonstrate that the bounds in Theorems 3.1 and 3.2 are tight.

Of course, when $V \in \mathbb{Z}_p[x]$ *is* a monic monomial—i.e., when $V(x) = x^{i-1}$ for some $i \in [1..n]$—then the set of polynomials P contains as an element the zero polynomial, for which *every* element of $\mathbb{Z}_p$ is a root. This observation together with the next corollary forms the basis for the completeness and soundness of our new zero-knowledge protocol.

Corollary 3.4. *Fix positive integers k and n such that $n \leq 2^k$. Let p be an odd prime, let $\vec{v} = \langle v_0, \ldots, v_{n-1}\rangle \in \mathbb{Z}_p^n$, and define the polynomial $V \in \mathbb{Z}_p[x]$ as $V(x) \coloneqq \sum_{i=0}^{n-1}(v_i \cdot x^i)$. If $\vec{v}$ is not a standard basis vector, then*

$$\Pr\left[\exists i \in [1..2^k], V(t) \equiv t^{i-1} \bmod p \;\middle|\; t \in_{\mathrm{R}} \mathbb{Z}_p\right] < 2^{2k}/p.$$

Proof. Notice that $V(t) = t^{i-1}$ if and only if t is a root of $h_i \coloneqq V(x) - x^{i-1}$. The latter polynomial is an element of the set P from Corollary 3.3, the elements of which have strictly fewer than 2^{2k} distinct roots in total; hence, a uniform choice for $t \in \mathbb{Z}_p$ will yield one of these strictly fewer than 2^{2k} roots with probability strictly less than $2^{2k}/p$, as desired. □

4 ZERO-KNOWLEDGE BUILDING BLOCKS

We now introduce four Σ-protocols that serve as building blocks for our new protocol. The first (§4.1) is (barely) new, while the second (§4.2) and third (§4.3) are entirely standard; ultimately, we present these first three protocols purely as stepping stones toward the fourth (§4.4) building block, a Σ-protocol for proving knowledge of a double DL with a publicly known base and a publicly known upper bound on the bitlength of the secret exponent. To the best of our knowledge, the latter double DL protocol *is* new, though perhaps unsurprising; it performs most of the heavy lifting for our main protocol, which follows in Section 5.

The reader should bare in mind that we have made some intentional presentation choices that make Protocols 1 and 2—and, to a lesser extent, Protocol 3—appear somewhat "messier" than is strictly necessary. The rationale for these choices will become apparent when we present Protocol 4, as they greatly increase the extent to which one can immediately "see" how we have amalgamated the first three protocols into our (significantly more complex) double DL protocol.

4.1 ZKPoK for a commitment to X or Y

Our first building block is a Σ-protocol allowing prover P to convince verifier V of its ability to open a commitment C to a value $a \in \mathbb{Z}_p$ such that either $a \equiv X \bmod p$ or $a \equiv Y \bmod p$. Here the values $X, Y \in \mathbb{Z}_p$ can be arbitrary (publicly known) constants, subject only to $X \not\equiv Y \bmod p$. In Camenisch-Stadler notation [10], the protocol implements

$$\mathrm{PK}\{\, r : C/g^X = h^r \;\vee\; C/g^Y = h^r \,\}.$$

The protocol is a modest adaptation of an existing protocol [16, Figure 1] allowing P to demonstrate its ability to open a commitment to some $a \in \mathbb{Z}_p$ such that either $a \equiv 0 \bmod p$ or $a \equiv 1 \bmod p$. The idea is quite simple: Given $C = g^a h^r$ as common input, V computes $\widehat{C} \coloneqq \left(C/g^X\right)^d = g^{(a-X)\cdot d} h^{r\cdot d}$ with $d \coloneqq (Y - X)^{-1} \bmod p$, and then it engages with P in an otherwise-standard proof of knowledge of the value $\hat{a} \equiv (a - X) \cdot d \bmod p$ committed to by $\widehat{C}$ such that either $\hat{a} \equiv 0 \bmod p$ or $\hat{a} \equiv 1 \bmod p$. This works because $\widehat{C}$ commits either to (i) $(X - X) \cdot d \equiv 0 \bmod p$ when $a \equiv X \bmod p$, or to (ii) $(Y - X) \cdot d \equiv 1 \bmod p$ when $a \equiv Y \bmod p$. For concreteness, the full protocol follows in Protocol 1 below.

Protocol 1: (HV)ZKPoK for a commitment to X or Y

Common input: $C\left[=g^{a_1}h^{r_1}\right] \in \mathbb{G}$ and $(X_1, Y_1) \in \mathbb{Z}_p \times \mathbb{Z}_p$

Prover knows: $(a_1, r_1) \in \mathbb{Z}_p \times \mathbb{Z}_p$

0: Define $d_1 \coloneqq (Y_1 - X_1)^{-1} \bmod p$ and $\widehat{C} \coloneqq \left(C/g^{X_1}\right)^{d_1}$.

1: P chooses $(s_1, t_1) \in_{\mathrm{R}} (\mathbb{Z}_p \times \mathbb{Z}_p) \times \mathbb{Z}_p$ and $\varepsilon_1 \in_{\mathrm{R}} \mathbb{Z}_p$ sends

1a: $A_1 \leftarrow g^{s_1} h^{t_1}$; and

1b: $\bar{A}_1 \leftarrow g^{s_1\cdot(a-X_1)\cdot d_1^2} h^{\varepsilon_1}$

to V.

2: V chooses $c \in_{\mathrm{R}} \mathbb{Z}_p$ and sends it to P.

3: P sends

3a: $(v_1, u_1) \leftarrow (s_1 + a_1 \cdot c \bmod p,\ t_1 + r_1 \cdot c \bmod p)$; and

3b: $w_1 \leftarrow \varepsilon_1 + r_1 \cdot d_1 \cdot \left(c - (v_1 - X_1 \cdot c) \cdot d_1\right) \bmod p$

to V.

4: V accepts if and only if (i) $A_1, \bar{A}_1 \in \mathbb{G}$, $(v_1, u_1) \in \mathbb{Z}_p \times \mathbb{Z}_p$, and $w_1 \in \mathbb{Z}_p$; and (ii) the following verification equations both hold:

4a: $A_1 C^c \stackrel{?}{=} g^{v_1} h^{u_1}$; and

4b: $\bar{A}_1 \widehat{C}^{c-(v_1-X_1\cdot c)\cdot d_1} \stackrel{?}{=} h^{w_1}$.

Theorem 4.1. *The Σ-protocol depicted in Protocol 1 is a system for 2-extractable special honest-verifier zero-knowledge proofs of knowledge of $r \in \mathbb{Z}_p$ such that either (i) $C = g^{X_1} h^r$ or (ii) $C = g^{Y_1} h^r$.*

Proof. We prove the completeness, the 2-extractability (including soundness), and the c-simulatability of the protocol in turn. In each of the following arguments, let $\bar{v}_1 \coloneqq (v_1 - X \cdot c) \cdot d_1 \bmod p$.

Completeness: Completeness follows easily by inspection. Indeed,

$$\begin{aligned} A_1 C^c &= \left(g^{s_1} h^{t_1}\right)\left(g^{a_1} h^{r_1}\right)^c \\ &= g^{s_1 + a_1 \cdot c} h^{t_1 + r_1 \cdot c} \\ &= g^{v_1} h^{u_1} \end{aligned}$$

and

$$\begin{aligned}\bar{A}_1\widehat{C}^{c-\bar{v}_1} &= \left(g^{s_1\cdot(a_1-X_1)\cdot d_1^2}h^{\varepsilon_1}\right)\left(g^{(a_1-X_1)\cdot d_1}h^{r_1\cdot d_1}\right)^{c-\bar{v}_1}\\ &= g^{s_1\cdot(a_1-X_1)\cdot d_1^2+(a_1-X_1)\cdot d_1\cdot(c-\bar{v}_1)}h^{\varepsilon_1+r_1\cdot d_1\cdot(c-\bar{v}_1)}\\ &= h^{w_1},\end{aligned}$$

where the last equality holds because either

(1) $a_1 - X_1 \equiv 0 \bmod p$ so that $s_1 \cdot (a_1 - X_1) \cdot d_1^2 \equiv (a_1 - X_1) \cdot d_1 \cdot (c - \bar{v}_1) \equiv 0 \bmod p$ when $a_1 \equiv X_1 \bmod p$; or

(2) $(a_1-X_1)\cdot d_1 \equiv 1 \bmod p$ so that $s_1\cdot(a_1-X_1)\cdot d_1^2 \equiv s_1\cdot d_1 \bmod p$ and

$$\begin{aligned}(a_1 - X_1) \cdot d_1 \cdot (c - \bar{v}_1) &\equiv \left(c - (v_1 - X_1 \cdot c) \cdot d_1\right)\\ &\equiv \left(c - (s_1 + a_1 \cdot c - X_1 \cdot c) \cdot d_1\right)\\ &\equiv \left(c - s_1 \cdot d_1 - (a_1 - X_1) \cdot d_1 \cdot c\right)\\ &\equiv c - s_1 \cdot d_1 - c\\ &\equiv -s_1 \cdot d_1 \bmod p\end{aligned}$$

when $a_1 \equiv Y_1 \bmod p$.

2-Extractability: Let

$$(C, X_1, Y_1;\, A_1, \bar{A}_1;\, c;\, (v_1, u_1), w_1)$$

and

$$(C, X_1, Y_1;\, A_1, \bar{A}_1;\, c';\, (v_1', u_1'), w_1')$$

be a pair of accepting transcripts that share common inputs (C, X_1, Y_1) and initial commitments $(A_1, \bar{A}_1)$, but that use distinct challenges $c \not\equiv c' \bmod p$. From the first verification equation, we have that $C^{c-c'} = g^{v_1-v_1'}h^{u_1-u_1'}$; hence, the extractor can compute

$$a_1 \equiv (v_1 - v_1') \cdot (c - c')^{-1} \bmod p$$

and

$$r_1 \equiv (u_1 - u_1') \cdot (c - c')^{-1} \bmod p,$$

which establishes that the protocol is 2-extractable. To see that it is also sound (i.e., that either $a_1 \equiv X_1 \bmod p$ or $a_1 \equiv Y_1 \bmod p$), note that

$$\begin{aligned}\bar{v}_1 - \bar{v}_1' &\equiv d_1 \cdot (v_1 - X_1 \cdot c) - d_1 \cdot (v_1' - X_1 \cdot c')\\ &\equiv d_1 \cdot \left((a_1 - X_1) \cdot c - (a_1 - X_1) \cdot c'\right)\\ &\equiv d_1 \cdot (a_1 - X_1) \cdot (c - c') \bmod p;\end{aligned}$$

hence,

$$\begin{aligned}\widehat{C}^{(c-\bar{v}_1)-(c'-\bar{v}_1')} &= \left(g^{a_1-X_1}h^{r_1}\right)^{d_1\cdot\left((c-\bar{v}_1)-(c'-\bar{v}_1')\right)}\\ &= \left(g^{a_1-X_1}h^{r_1}\right)^{d_1\cdot\left((c-c')-(\bar{v}_1-\bar{v}_1')\right)}\\ &= g^{(a_1-X_1)\cdot d_1\cdot\left((c-c')-d_1\cdot(c-c')\cdot(a_1-X_1)\right)}h^{w_1-w_1'}\\ &= g^{(a_1-X_1)\cdot d_1\cdot(c-c')\cdot(1-(a_1-X_1)\cdot d_1)}h^{w_1-w_1'}.\end{aligned}$$

Moreover, from the second verification equation, we also have that $\widehat{C}^{(c-\bar{v}_1)-(c'-\bar{v}_1')} = h^{w_1-w_1'}$, which implies

$$(a_1 - X_1) \cdot d_1 \cdot (c - c') \cdot (1 - (a_1 - X_1) \cdot d_1) \equiv 0 \bmod p.$$

As $c \not\equiv c' \bmod p$ and $d_1 \not\equiv 0 \bmod p$, this latter congruence holds if and only if $a_1 - X_1 \equiv 0 \bmod p$ or $(a_1 - X_1) \cdot d_1 \equiv 1 \bmod p$; i.e., if and only if $a_1 \equiv X_1 \bmod p$ or $a_1 \equiv Y_1 \bmod p$, as desired.

***c*-Simulatability:** Given the common input (C, X_1, Y_1) and a challenge $c \in \mathbb{Z}_p$, a simulator for the honest verifier samples uniform responses $((v_1, u_1), w_1) \in_{\mathrm{R}} (\mathbb{Z}_p \times \mathbb{Z}_p) \times \mathbb{Z}_p$, and then it computes

$$A_1 \leftarrow C^{-c} g^{v_1} h^{u_1}$$

and

$$\bar{A}_1 \leftarrow \left(C/g^{X_1}\right)^{-d_1\cdot(c-\bar{v}_1)} h^{w_1},$$

and outputs the simulated transcript

$$(C, X_1, Y_1;\, A_1, \bar{A}_1;\, c;\, (v_1, u_1), w_1).$$

Notice that, by virtue of P uniformly selecting $(s_1, t_1), \varepsilon_1 \in_{\mathrm{R}} \mathbb{Z}_p$ in Step 2, $((v_1, u_1), w_1)$ is also uniformly distributed in (accepting) real transcripts. Also notice that $(A_1, \bar{A}_1)$ is uniquely determined by $(c, (v_1, u_1), w_1)$ and the common input. Hence, the simulated transcripts are drawn from the same probability distribution as those of accepting proofs with an honest verifier who just so happens to choose challenge c. □

4.2 ZKPoK for 2-ary multiplication

Our second building block is a (completely standard) Σ-protocol allowing prover P to convince verifier V of its ability to open a triplet of commitments (C_1, C_2, C_3') to values $(a_1, a_2, a_3') \in \mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p$ such that $a_3' \equiv a_1 \cdot a_2 \bmod p$. In Camenisch-Stadler notation [10], the protocol implements

$$\mathrm{PK}\left\{(a_1, a_2, r_1, r_2, \gamma_3) : C_1 = g^{a_1}h^{r_1} \wedge C_2 = g^{a_2}h^{r_2} \wedge C_3' = g^{a_1\cdot a_2}h^{\gamma_3}\right\}.$$

Again, the idea is quite simple: Given $C_1 = g^{a_1}h^{r_1}$, $C_2 = g^{a_2}h^{r_2}$, and $C_3' = g^{a_3}h^{\gamma_3}$ as common input, P demonstrates knowledge of $(a_1, r_1), (a_2, r_2)$, and γ_3 such that $\log_{(g^{a_2})}\left(C_3'/h^{\gamma_3}\right) \equiv a_1 \bmod p$, from which it follows that $\left(C_3'/h^{\gamma_3}\right)^{a_1} = g^{a_1\cdot a_2}$ or, equivalently, that $a_3' \equiv a_1 \cdot a_2 \bmod p$. The full protocol follows in Protocol 2 below.

Theorem 4.2. *The Σ-protocol depicted in Protocol 2 is a system for 2-extractable, special honest-verifier zero-knowledge proofs of knowledge of $(a_1, r_1), (a_2, r_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and $\gamma_3 \in \mathbb{Z}_p$ such that (i) $C_1 = g^{a_1}h^{r_1}$, (ii) $C_2 = g^{a_2}h^{r_2}$, and (iii) $C_3' = g^{a_1\cdot a_2}h^{\gamma_3}$.*

Proof. We prove the completeness, the 2-extractability (including soundness), and the *c*-simulatability of the protocol in turn.

Completeness: Completeness follows easily by inspection; indeed,

$$\begin{aligned}A_1C_1^c &= \left(g^{s_1}h^{t_1}\right)\left(g^{a_1}h^{r_1}\right)^c\\ &= g^{s_1+a_1\cdot c}h^{t_1+r_1\cdot c}\\ &= g^{v_1}h^{u_1};\\ A_2C_2^c &= \left(g^{s_2}h^{t_2}\right)\left(g^{a_2}h^{r_2}\right)^c\\ &= g^{s_2+a_2\cdot c}h^{t_2+r_2\cdot c}\\ &= g^{v_2}h^{u_2};\end{aligned}$$

Protocol 2: (HV)ZKPoK for product of (two) committed values

Common input: $C_1\left[=g^{a_1}h^{r_1}\right], C_2\left[=g^{a_2}h^{r_2}\right], C_3'\left[=g^{a_1\cdot a_2}h^{\gamma_3}\right] \in \mathbb{G}$

Prover knows: $(a_1, r_1), (a_2, r_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and $\gamma_3 \in \mathbb{Z}_p$

1: P chooses $(s_1, t_1), (s_2, t_2) \in_{\mathrm{R}} \mathbb{Z}_p \times \mathbb{Z}_p$ and $\delta_3 \in_{\mathrm{R}} \mathbb{Z}_p$ and sends
  1a: $A_1 \leftarrow g^{s_1}h^{t_1}$;
  1b: $A_2 \leftarrow g^{s_2}h^{t_2}$; and
  1c: $A_3' \leftarrow g^{a_1\cdot s_2}h^{\delta_3}$
  to V.
2: V chooses $c \in_{\mathrm{R}} \mathbb{Z}_p$ and sends it to P.
3: P sends
  3a: $(\nu_1, u_1) \leftarrow (s_1 + a_1 \cdot c \bmod p,\ t_1 + r_1 \cdot c \bmod p)$;
  3b: $(\nu_2, u_2) \leftarrow (s_2 + a_2 \cdot c \bmod p,\ t_2 + r_2 \cdot c \bmod p)$; and
  3c. $z_3 \leftarrow (-r_1 \cdot \nu_2) + \delta_3 + \gamma_3 \cdot c \bmod p$
  to V.
4: V accepts if and only if (i) $A_1, A_2, A_3' \in \mathbb{G}$, both (ν_1, u_1), $(\nu_2, u_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$, and $z_3 \in \mathbb{Z}_p$; and (ii) the following verification equations all hold:
  4a: $A_1C_1^c \stackrel{?}{=} g^{\nu_1}h^{u_1}$;
  4b: $A_2C_2^c \stackrel{?}{=} g^{\nu_2}h^{u_2}$; and
  4c: $A_3'(C_3')^c \stackrel{?}{=} C_1^{\nu_2}h^{z_3}$.

and

$$\begin{aligned} A_3'(C_3')^c &= \left(g^{a_1\cdot s_2}h^{\delta_3}\right)\left(g^{a_1\cdot a_2}h^{\delta_3}\right)^c \\ &= g^{a_1\cdot s_2 + a_1\cdot a_2\cdot c}h^{\delta_3+\gamma_3\cdot c} \\ &= g^{a_1\cdot(s_2+a_2\cdot c)}h^{\delta_3+\gamma_3\cdot c} \\ &= g^{a_1\cdot\nu_2}h^{\delta_3+\gamma_3\cdot c} \\ &= g^{a_1\cdot\nu_2}h^{r_1\cdot\nu_2}h^{-r_1\cdot\nu_2}h^{\delta_3+\gamma_3\cdot c} \\ &= C_1^{\nu_2}h^{(-r_1\cdot\nu_2)+\delta_3+\gamma_3\cdot c} \\ &= C_1^{\nu_2}h^{z_3}. \end{aligned}$$

2-Extractability: Let

$$(C_1, C_2, C_3'; A_1, A_2, A_3'; c; (\nu_1, u_1), (\nu_2, u_2), z_3)$$

and

$$(C_1, C_2, C_3'; A_1, A_2, A_3'; c'; (\nu_1', u_1'), (\nu_2', u_2'), z_3')$$

be a pair of accepting transcripts that share common inputs (C_1, C_2, C_3') and initial commitments (A_1, A_2, A_3'), but that use distinct challenges $c \not\equiv c' \bmod p$. From the first two verification equations, we have $C_1^{c-c'} = g^{\nu_1-\nu_1'}h^{u_1-u_1'}$ and $C_2^{c-c'} = g^{\nu_2-\nu_2'}h^{u_2-u_2'}$; hence, the extractor can compute

$$\begin{aligned} a_1 &\equiv (\nu_1 - \nu_1')\cdot(c-c')^{-1} \bmod p, \\ a_2 &\equiv (\nu_2 - \nu_2')\cdot(c-c')^{-1} \bmod p, \\ r_1 &\equiv (u_1 - u_1')\cdot(c-c')^{-1} \bmod p, \end{aligned}$$

and

$$r_2 \equiv (u_2 - u_2')\cdot(c-c')^{-1} \bmod p,$$

which establishes that the protocol is 2-extractable with respect to (a_1, r_1) and (a_2, r_2). To see that it is also 2-extractable with respect to γ_3 as well as sound (i.e., that the prover can indeed open C_3' to $a_3' \equiv a_1 \cdot a_2 \bmod p$), note that, from the third verification equation, we also have that $(C_3')^{c-c'} = C_1^{\nu_2-\nu_2'}h^{z_3-z_3'}$ so that

$$\begin{aligned} C_3' &= \left(C_1^{(\nu_2-\nu_2')}h^{(z_3-z_3')}\right)^{(c-c')^{-1}} \\ &= C_1^{(\nu_2-\nu_2')\cdot(c-c')^{-1}}h^{(z_3-z_3')\cdot(c-c')^{-1}} \\ &= C_1^{a_2}h^{(z_3-z_3')\cdot(c-c')^{-1}} \\ &= g^{a_1\cdot a_2}h^{r_1\cdot a_2}h^{(z_3-z_3')\cdot(c-c')^{-1}} \\ &= g^{a_1\cdot a_2}h^{r_1\cdot a_2+(z_3-z_3')\cdot(c-c')^{-1}}; \end{aligned}$$

hence, the extractor can compute $\gamma_3 \equiv r_1 \cdot a_2 + (z_3 - z_3') \cdot (c - c')^{-1} \bmod p$ using the r_1, a_2 extracted above.

***c*-Simulatability:** Given the common input (C_1, C_2, C_3') and a challenge $c \in \mathbb{Z}_p$, a simulator for the honest verifier samples uniform $\left((\nu_1, u_1), (\nu_2, u_2), z_3\right) \in_{\mathrm{R}} (\mathbb{Z}_p \times \mathbb{Z}_p) \times (\mathbb{Z}_p \times \mathbb{Z}_p) \times \mathbb{Z}_p$, and then it computes

$$\begin{aligned} A_1 &\leftarrow C_1^{-c}g^{\nu_1}h^{u_1}; \\ A_2 &\leftarrow C_2^{-c}g^{\nu_2}h^{u_2}; \end{aligned}$$

and

$$A_3' \leftarrow C_3^{-c}C_1^{\nu_2}h^{z_3},$$

and outputs the simulated transcript

$$(C_1, C_2, C_3'; A_1, A_2, A_3'; c; (u_1, \nu_1), (u_2, \nu_2), z_3).$$

Notice that, by virtue of P uniformly selecting $(s_1, t_1), (s_2, t_2) \in_{\mathrm{R}} \mathbb{Z}_p \times \mathbb{Z}_p$ and $\delta_3 \in_{\mathrm{R}} \mathbb{Z}_p$ in Step 2, $\left((\nu_1, u_1), (\nu_2, u_2), z_3\right)$ is also uniformly distributed in (accepting) real transcripts. Also notice that (A_1, A_2, A_3') is uniquely determined by $\left(c, (\nu_1, u_1), (\nu_2, u_2), z_3\right)$ and the common input; hence, the simulated transcripts are drawn from the same probability distribution as those of accepting proofs with an honest verifier who just so happens to choose challenge c. □

4.3 ZKPoK for *k*-ary multiplication

Our third building block is a relatively straightforward generalization of Protocol 2 into a Σ-protocol allowing prover P to convince verifier V of its ability to open a sequence of commitments $(C_1, \ldots, C_k, C_{k+1}')$ to values $(a_1, \ldots, a_k, a_{k+1}') \in \mathbb{Z}_p^{k+1}$ such that $a_{k+1}' \equiv \prod_{j=1}^k a_j \bmod p$. In Camenisch-Stadler notation [10], the resulting generalization implements

$$\mathrm{PK}\left\{((a_j, r_j)_{j=1}^k, \gamma_{k+1}) : \left(\textstyle\bigwedge_{j=1}^k C_j = g^{a_j}h^{r_j}\right) \wedge C_{k+1}' = g^{(\prod_{j=1}^k a_j)}h^{\gamma_{k+1}}\right\}.$$

The protocol is constructed via a k-fold (parallel) composition of Protocol 2, emitting any requisite commitments to partial products and omitting any "redundant" commitments, responses, and verification equations along the way. For completeness, the full protocol follows in Protocol 3.

A full proof of the completeness, 2-extractability (including soundness), and c-simulatability for Protocol 3 closely mirrors that

Protocol 3: (HV)ZKPoK for product of k committed values

Common input: $\left(C_j\,[=g^{a_j}h^{r_j}]\right)_{j=1}^{k}, C'_{k+1}\,[=g^{(\prod_{j=1}^{k}a_j)}h^{\gamma_{k+1}}] \in \mathbb{G}$

Prover knows: $(a_1,r_1),\ldots,(a_k,r_k) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and $\gamma_{k+1} \in \mathbb{Z}_p$

0: For convenience, define C'_2 and γ_2 as synonyms for C_1 and r_1, respectively.

1: **for each** ($j \in [1\,..\,k]$) **do**

1a: P chooses $(s_j,t_j) \in_{\mathrm{R}} \mathbb{Z}_p \times \mathbb{Z}_p$ and sends $A_j \leftarrow g^{s_j}h^{t_j}$ to V.

end for

for each ($j \in [3\,..\,k+1]$) **do**

1b: P chooses $\delta_j \in_{\mathrm{R}} \mathbb{Z}_p$ and sends $A'_j \leftarrow g^{(s_{j-1}\prod_{l=1}^{j-2}a_l)}h^{\delta_j}$ to V.

end for

for each ($j \in [3\,..\,k]$) **do**

1c: P chooses $\gamma_j \in_{\mathrm{R}} \mathbb{Z}_p$ and sends $C'_j \leftarrow g^{(\prod_{l=1}^{j-1}a_l)}h^{\gamma_j}$ to V.

end for

2: V chooses $c \in_{\mathrm{R}} \mathbb{Z}_p$ and sends it to P.

3: **for each** ($j \in [1\,..\,k]$) **do**

3a: P sends $(\nu_j,u_j) \leftarrow (s_j + a_j\cdot c \bmod p,\ t_j + r_j\cdot c \bmod p)$ to V.

end for

for each ($j \in [3\,..\,k+1]$) **do**

3b: P sends $z_j \leftarrow (-\gamma_{j-1}\cdot\nu_{j-1}) + \delta_j + \gamma_j\cdot c \bmod p$ to verifier.

end for

4: V accepts if and only if (i) $A_j \in \mathbb{G}$ and $(u_j,\nu_j) \in \mathbb{Z}_p \times \mathbb{Z}_p$ for each $j \in [1\,..\,k]$; (ii) $(C'_j,A'_j) \in \mathbb{G}\times\mathbb{G}$ and $z_j \in \mathbb{Z}_p$ for each $j \in [3\,..\,k+1]$; and (iii) the following verification equations all hold:

4a: $A_jC_j^c \stackrel{?}{=} g^{\nu_j}h^{u_j}$ for each $j \in [1\,..\,k]$; and

4b: $A'_j(C'_j)^c \stackrel{?}{=} C'^{\nu_{j-1}}_{j-1}h^{z_j}$ for each $j \in [3\,..\,k+1]$.

for Protocol 2. Indeed, one can easily verify that, given the transcript of an accepting protocol run between P and honest V, the sub-transcript comprising

$$(C_1,C_2,C'_3;\ A_1,A_2,A'_3;\ c;\ (\nu_1,u_1),(\nu_2,u_2),z_3)$$

is itself an accepting transcript of Protocol 2 with common inputs (C_1,C_2,C'_3), thus establishing that P knows $(a_1,r_1) \in \mathbb{Z}_p\times\mathbb{Z}_p$, $(a_2,r_2) \in \mathbb{Z}_p\times\mathbb{Z}_p$, and $\gamma_3 \in \mathbb{Z}_p$ such that $C_1 = g^{a_1}h^{r_1}$, $C_2 = g^{a_2}h^{r_2}$, and $C'_3 = g^{a_1\cdot a_2}h^{\gamma_3}$. In a similar vein, the sub-transcript comprising

$$(C_3,C'_3,C'_4;\ A_3,A'_3,A'_4;\ c;\ (\nu_3,u_3),z_4)$$

establishes that P also knows $(a_3,r_3) \in \mathbb{Z}_p\times\mathbb{Z}_p$ such that $C_3 = g^{a_3}h^{r_3}$ and, moreover, that *if* P knows some $(a'_3,\gamma_3) \in \mathbb{Z}_p\times\mathbb{Z}_p$ satisfying $C'_3 = g^{a'_3}h^{\gamma_3}$ (which we just proved it must; namely, $a'_3 \equiv a_1\cdot a_2 \bmod p$), *then* it also knows $\gamma_4 \in \mathbb{Z}_p$ such that $C'_4 = g^{a_3\cdot a_3}h^{\gamma_4}$. More generally, the sub-transcript comprising

$$(C_j,C'_j,C'_{j+1};\ A_j,A'_j,A'_{j+1};\ c;\ (\nu_j,u_j),z_{j+1})$$

"extends" the proof to establish that P knows $(a_j,r_j) \in \mathbb{Z}_p\times\mathbb{Z}_p$ and $\gamma_{j+1} \in \mathbb{Z}_p$ such that $C_j = g^{a_j}h^{r_j}$ and $C'_{j+1} = g^{(a_j\prod_{l=1}^{j-1}a_l)}h^{\gamma_{j+1}}$; when $j = k$, this establishes precisely what P set out to prove. Thus, the key difference between Protocol 3 and a naïve k-fold parallelization of Protocol 2 is that the former eschews "direct" proofs of knowledge for the partial products committed to by the C'_j in favour of "implicit" proofs of knowledge following from the demonstrated relationship that each C'_j has with C_j and C'_{j-1}. In light of the preceding correspondence with Protocol 2, we forgo a self-contained proof of the following theorem.

Theorem 4.3. *The Σ-protocol depicted in Protocol 3 is a system for 2-extractable, special honest-verifier zero-knowledge proofs of knowledge of* $((a_1,r_1),\ldots(a_k,r_k),\gamma_{k+1})$ *such that (i)* $C_j = g^{a_j}h^{r_j}$ *for each* $j \in [1\,..\,k]$*, and (ii)* $C_{k+1} = g^{(\prod_{j=1}^{k}a_j)}h^{\gamma_{k+1}}$.

Protocol 4: (HV)ZKPoK for double discrete logarithm

Common input: $C'_{k+1}\left[=g^{t^\ell}h^{\gamma_{k+1}}\right] \in \mathbb{G}$, $t \in \mathbb{Z}_p$, and $k \in \mathbb{N}$

Prover knows: $\ell \in [1\,..\,2^k]$ and $\gamma_{k+1} \in \mathbb{Z}_p$

0a: For each $j \in [1\,..\,k]$, define $d_j := (t^{2^{j-1}}-1)^{-1} \bmod p$.

0b: Define C'_2 and γ_2 as synonyms for C_1 and r_1, respectively.

1: Express $\ell-1 = \ell_k\ell_{k-1}\cdots\ell_1$ in binary and define $L(j) := \ell_j\cdot 2^{j-1}$ and $a_j := t^{L(j)} \bmod p$ for each $j \in [1\,..\,k]$.

for each ($j \in [1\,..\,k]$) **do**

1a: P chooses $r_j \in_{\mathrm{R}} \mathbb{Z}_p$ and $(s_j,t_j) \in_{\mathrm{R}} \mathbb{Z}_p\times\mathbb{Z}_p$, and then it sends $C_j \leftarrow g^{a_j}h^{r_j}$ and $A_j \leftarrow g^{s_j}h^{t_j}$ to V; and

1b: P chooses $\varepsilon_j \in_{\mathrm{R}} \mathbb{Z}_p$ and sends $\bar{A}_j \leftarrow g^{s_j\cdot(a_j-1)\cdot d_j^2}h^{\varepsilon_j}$ to V.

end for

for each ($j \in [3\,..\,k]$) **do**

1c: P chooses $\gamma_j \in_{\mathrm{R}} \mathbb{Z}_p$ and sends $C'_j \leftarrow g^{(\prod_{l=1}^{j-1}a_l)}h^{\gamma_j}$ to V.

end for

for each ($j \in [3\,..\,k+1]$) **do**

1d: P chooses $\delta_j \in_{\mathrm{R}} \mathbb{Z}_p$ and sends $A'_j \leftarrow g^{(s_{j-1}\prod_{l=1}^{j-2}a_l)}h^{\delta_j}$ to V.

end for

2: V chooses $c \in_{\mathrm{R}} \mathbb{Z}_p$ and sends it to P.

3: **for each** ($j \in [1\,..\,k]$) **do**

3a: P sends $(\nu_j,u_j) \leftarrow (s_j + a_j\cdot c \bmod p,\ t_j + r_j\cdot c \bmod p)$ to V; and

3b: P sends $w_j \leftarrow \varepsilon_j + r_j\cdot d_j\cdot(c-(\nu_j-c)\cdot d_j)$ to V.

end for

for each ($j \in [3\,..\,k+1]$) **do**

3c: P sends $z_j \leftarrow (-\gamma_{j-1}\cdot\nu_{j-1}) + \delta_j + \gamma_j\cdot c \bmod p$ to verifier.

end for

4: V accepts if and only if (i) $(C_j,A_j,\bar{A}_j) \in \mathbb{G}\times\mathbb{G}\times\mathbb{G}$ and $(u_j,\nu_j,w_j) \in \mathbb{Z}_p\times\mathbb{Z}_p\times\mathbb{Z}_p$ for each $j \in [1\,..\,k]$; (ii) $(C'_j,A'_j) \in \mathbb{G}\times\mathbb{G}$ and $z_j \in \mathbb{Z}_p$ for each $j \in [3\,..\,k+1]$; and (iii) the following verification equations all hold:

4a. $A_jC_j^c \stackrel{?}{=} g^{\nu_j}h^{u_j}$ for each $j \in [1\,..\,k]$;

4b: $\bar{A}_j\left(C_j/g\right)^{d_j\cdot(c-(\nu_j-c)\cdot d_j)} \stackrel{?}{=} h^{w_j}$ for each $j \in [1\,..\,k]$; and

4c. $A'_j(C'_j)^c \stackrel{?}{=} (C'_{j-1})^{\nu_{j-1}}h^{z_j}$ for each $j \in [3\,..\,k+1]$.

4.4 ZKPoK for (public-base) double DL

Our fourth and final building block is a Σ-protocol allowing prover P to convince verifier V that it knows a double DL with a fixed

publicly known base $t \in \mathbb{Z}_p$ and a publicly known upper bound $k \in \mathbb{N}$ on the bitlength of the secret exponent; that is, for proving knowledge of $\ell \in [1..2^k]$ such that P can open C to $t^{\ell-1} \in \mathbb{Z}_p$. In Camenisch-Stadler notation [10], the protocol implements

$$\mathrm{PK}\{(\ell,\gamma) : C = g^{t^{\ell-1}} h^{\gamma} \wedge \ell \in [1..2^k]\}.$$

The most common way to prove knowledge of such double DLs is via addition chains; e.g., using an oblivious form of the ubiquitous "left-to-right double-and-add" method to homomorphically reconstruct $\ell - 1$ in the exponent of t (all in the exponent of g). Here we propose a slightly different approach that uses a straightforward composition of Algorithms 1 and 3 to reconstruct $\ell - 1$ in the exponent of t "directly" from $(\ell - 1)$'s binary decomposition. This approach is quite efficient when the bitlength of $\ell - 1$ is *a priori* known to be small.

Specifically, we express $\ell - 1 = \ell_k \ell_{k-1} \cdots \ell_1$ in binary, with each ℓ_j denoting the jth-least-significant bit of $\ell - 1$ (padding to the left with zeros as necessary) and, for each $j \in [1..k]$, we define $L(j) \coloneqq \ell_j \cdot 2^{j-1}$ so that $\ell - 1 = \sum_{j=1}^{k} L(j)$. Notice that $L(j) \in \{0, 2^{j-1}\}$ for each $j \in [1..k]$. From here, P simply outputs the sequence of commitments

$$(C_1, \ldots, C_k) \leftarrow (g^{t^{L(1)}} h^{r_1}, \ldots, g^{t^{L(k)}} h^{r_k}),$$

and then it uses a k-fold parallel composition of Protocol 1 to prove for each $j \in [1..k]$ that it indeed knows $(a_j, r_j) \in \mathbb{Z}_p \times \mathbb{Z}_p$ such that (i) $C_j = g^{a_j} h^{r_j}$, and (ii) either $a_j \equiv t^{2^{j-1}} \bmod p$ or $a_j \equiv 1 \bmod p$. Finally, P uses Protocol 3 to prove that it also knows $\gamma \in \mathbb{Z}_p$ such that $C = g^{(\prod_{j=1}^{k} a_j)} h^{\gamma}$. To see that this approach is *complete*, it suffices to observe that

$$\begin{aligned} \textstyle\prod_{j=1}^{k} a_j &\equiv \textstyle\prod_{j=1}^{k} t^{L(j)} \\ &\equiv t^{\sum_{j=1}^{k} L(j)} \\ &\equiv t^{\ell-1} \bmod p \end{aligned}$$

by construction. Likewise, to see that this approach is *sound* (i.e., that $\ell \in [1..2^k]$), it suffices to note that the guarantee that $a_j \in \{1, t^{2^{j-1}}\}$ implies $\log_t a_j \in \{0, 2^{j-1}\}$ for each $j \in [1..k]$, which in turn implies that

$$\begin{aligned} \log_t(\textstyle\prod_{j=1}^{k} a_j) &\leq \log_t(\textstyle\prod_{j=1}^{k} t^{2^{j-1}}) \\ &= \textstyle\sum_{j=1}^{k} 2^{j-1} \\ &= 2^k - 1. \end{aligned}$$

Theorem 4.4. *The Σ-protocol depicted in Protocol 4 is a system for 2-extractable, special honest-verifier zero-knowledge proofs of knowledge of $\ell \in [1..2^k]$ and $\gamma \in \mathbb{Z}_p$ such that $C = g^{t^{\ell-1}} h^{\gamma}$.*

Proof (Sketch). By inspection, the "sub-protocol" consisting of Steps 1a,b; 2; 3a,b; and 4a,b comprises k parallel instances of Protocol 1 with the jth instance using $X_j = 1$ and $Y_j = t^{2^{j-1}}$. Theorem 4.1 establishes that the subprotocol is complete, 2-extractable (allowing to extract the binary decomposition of ℓ) and sound, and special honest-verifier zero-knowledge.

Meanwhile, the "sub-protocol" consisting of Steps 1a,c,d; 2; 3a,c; and 4a,c is an instance of Protocol 3 establishing that the prover can open the common input to the product of values whose correctness was established by the first subprotocol. Theorem 4.3 establishes that the latter subprotocol is complete, 2-extractable (again, allowing to extract the binary decomposition of ℓ as well as γ), and special honest-verifier zero-knowledge. □

5 ZKPoK FOR ONE-HOTNESS

We now present our main result, a 4-move special honest-verifier zero-knowledge proof of knowledge of $\ell \in [1..n]$ such that the vector of commitments $\vec{E} \coloneqq \langle E_1, \ldots, E_n \rangle$ opens component-wise to the ℓth standard basis vector for $\mathbb{Z}_p^n$.

Protocol 5: (HV)ZKPoK for opening of a standard basis vector

Common input: $E_1 \left[= g^{\vec{e}_\ell[1]} h^{\beta_1}\right], \ldots, E_n \left[= g^{\vec{e}_\ell[n]} h^{\beta_n}\right] \in \mathbb{G}$
Prover knows: $\ell \in [1..n]$ and $\beta_1, \ldots, \beta_n \in \mathbb{Z}_p$

1: V chooses $t \in_{\mathrm{R}} \mathbb{Z}_p$, and then it computes $\widetilde{E}(t) \leftarrow \prod_{i=1}^{n} (E_i)^{t^{i-1}}$ and sends t to P.
2: P computes $\gamma \leftarrow \sum_{i=1}^{n} (\beta_i \cdot t^{i-1}) \bmod p$, and then it engages V in the Σ-protocol denoted by

$$\mathrm{PK}\{(\ell,\gamma) : \widetilde{E}(t) = g^{t^{\ell-1}} h^{\gamma} \wedge \ell \in [1..2^k]\}$$

and realized by Protocol 4.
3: V accepts if and only if it accepts in the Σ-protocol in Step 2.

Theorem 5.1. *The 4-move protocol depicted in Protocol 5 is a system for honest-verifier zero-knowledge proofs of knowledge of $(\ell; \beta_1, \ldots, \beta_n)$ such that $E_i = g^{\vec{e}_\ell[i]} h^{\beta_i}$ for each $i \in [1..n]$. It is special honest-verifier zero-knowledge; the witness $\ell \in [1..n]$ is 2-extractable; and the witness tuple $(\beta_1, \ldots, \beta_n) \in \mathbb{Z}_p^n$ is $(2n + 2)$-extractable.*

Proof. We prove the completeness, the 2-extractablitity (including soundness), and the special honest-verifier simulatablity of the protocol in turn.

Completeness: V accepts if and only if it accepts in the Σ-protocol of Step 2; i.e., if P indeed knows $\ell \in [1..2^k]$ and $\gamma \in \mathbb{Z}_p$ such that $\widetilde{E}(t) = g^{t^{\ell-1}} h^{\gamma}$. Thus, it suffices to note that, when indeed each $E_i = g^{\vec{e}_\ell[i]} h^{\beta_i}$, we have that

$$\begin{aligned} \widetilde{E}(t) &= \textstyle\prod_{i=1}^{n} (E_i)^{t^{i-1}} \\ &= \textstyle\prod_{i=1}^{n} \left(g^{\vec{e}_\ell[i]} h^{\beta_i}\right)^{t^{i-1}} \\ &= \textstyle\prod_{i=1}^{n} g^{\vec{e}_\ell[i] \cdot t^{i-1}} h^{\beta_i \cdot t^{i-1}} \\ &= g^{(\sum_{i=1}^{n} \vec{e}_\ell[i] \cdot t^{i-1})} h^{(\sum_{i=1}^{n} \beta_i \cdot t^{i-1})} \\ &= g^{t^\ell} h^{\gamma}, \end{aligned}$$

where $\gamma \leftarrow \sum_{i=1}^{n} (\beta_i \cdot t^{i-1}) \bmod p$.

Extractability: From Theorem 4.4, there exists an extractor *Ext* for Protocol 4 that, by rewinding P just once, can extract $\ell \in [1..2^k]$ and $\gamma \in \mathbb{Z}_p$ such that $\widetilde{E}(t) = g^{t^{\ell-1}} h^{\gamma}$ in Step 2 (assuming V sends $t \in \mathbb{Z}_p$ in Step 1). We now show how to construct an extractor for the full Protocol 5 that, by using *Ext* as a subroutine, extracts a basis vector $\vec{e}_\ell \in \mathbb{Z}_p^n$ and $\langle \beta_1, \ldots, \beta_n \rangle \in \mathbb{Z}_p^n$ such that $E_i = g^{\vec{e}_\ell[i]} h^{\beta_i}$ for each $i \in [1..n]$.

The extractor works as follows: First, it repeatedly samples challenges $t_i \in_{\mathrm{R}} \mathbb{Z}_p$ and invokes *Ext* to obtain the corresponding

ordered pairs $(\ell_i, \gamma_i) \in [1..2^k] \times \mathbb{Z}_p$ satisfying $\widetilde{E}(t_i) = g^{t_i^{\ell_i - 1}} h^{\gamma_i}$.[4] Upon successfully extracting such pairs for $n+1$ pairwise distinct choices of $t_i \in \mathbb{Z}_p$, the extractor uses the pairs to construct two length-n sequences of points

$$(t_1, t_1^{\ell_1 - 1}), \ldots, (t_n, t_n^{\ell_n - 1}) \in \mathbb{Z}_p \times \mathbb{Z}_p$$

and

$$(t_1, \gamma_1), \ldots, (t_n, \gamma_n) \in \mathbb{Z}_p \times \mathbb{Z}_p,$$

plus two "extra" distinguished points $(t_0, t_0^{\ell_0 - 1}) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and $(t_0, \gamma_0) \in \mathbb{Z}_p \times \mathbb{Z}_p$ to be used later on.

Next, the extractor interpolates through the n points in the first and second sequences to respectively obtain the degree-$(n-1)$ polynomials $A \in \mathbb{Z}_p[x]$ and $B \in \mathbb{Z}_p[x]$ satisfying

$$\widetilde{E}(t_i) = g^{A(t_i)} h^{B(t_i)}$$

for all $i \in [1..n]$; whence, it follows that

$$\begin{aligned} g^{A(t_i)} h^{B(t_i)} &= \left(\prod_{i=1}^n E_i\right)^{t_i^{i-1}} \\ &= \left(\prod_{i=1}^n g^{a_i} h^{\beta_i}\right)^{t_i^{i-1}} \\ &= \prod_{i=1}^n g^{a_i \cdot t_i^{i-1}} h^{\beta_i \cdot t_i^{i-1}} \\ &= g^{\left(\sum_{i=1}^n a_i \cdot t_i^{i-1}\right)} h^{\left(\sum_{i=1}^n \beta_i \cdot t_i^{i-1}\right)}, \end{aligned}$$

and we can write $E_i = g^{a_i} h^{\beta_i}$ for each $i \in [1..n]$ using the coefficients of $A(x) = \sum_{i=1}^n (a_i \cdot x^{i-1})$ and $B(x) = (\sum_{i=1}^n \beta_i \cdot x^{i-1})$.

We have thus far shown how our extractor can open $\vec{E}$ component-wise to *some* vector $\langle a_1, \ldots, a_n \rangle \in \mathbb{Z}_p^n$; it now remains to show that it can open $\vec{E}$ component-wise to some *standard basis vector* from $\mathbb{Z}_p^n$. For this, we employ the "extra" distinguished points $(t_0, t_0^{\ell_0 - 1})$ and (t_0, γ_0). In particular, we consider the following two cases:

Case 1 ($A(t_0) \equiv t_0^{\ell_0 - 1} \bmod p$ and $B(t_0) \equiv \gamma_0 \bmod p$): By Corollary 3.4, this event occurs with probability strictly less than $2^{2k}/p$, unless $A(x) = x^{\ell_0 - 1}$ so that the vector $\langle a_1, \ldots, a_n \rangle$ already extracted is indeed the ℓ_0th standard basis vector, as desired.

Case 2 ($A(t_0) \not\equiv t_0^{\ell_0 - 1} \bmod p$ and $B(t) \not\equiv \gamma_0 \bmod p$): Here, the extractor holds (distinct) ordered pairs $(t_0^{\ell_0 - 1}, \gamma_0) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and $(A(t_0), B(t_0)) \in \mathbb{Z}_p \times \mathbb{Z}_p$ such that both

$$\widetilde{E}(t_0) = g^{t_0^{\ell_0 - 1}} h^{\gamma_0}$$

and

$$\widetilde{E}(t_0) = g^{A(t_0)} h^{B(t_0)};$$

hence, $g^{t_0^{\ell_0 - 1}} h^{\gamma_0} = g^{A(t_0)} h^{B(t_0)}$ and the extractor can solve for

$$\log_g h \equiv (t_0^{\ell_0 - 1} - A(t_0)) \cdot (B(t_0) - \gamma_0)^{-1} \bmod p.$$

From here, the extractor is free to choose any $\ell \in [1..n]$, define $B_\ell := B(x) - (\log_g h)(A(x) - x^\ell)$, and then output $\vec{e}_\ell \in \mathbb{Z}_p^n$ and $\langle B_\ell(1), \ldots, B_\ell(n) \rangle \in \mathbb{Z}_p^n$ as an opening of $\vec{E}$ to $\vec{e}_\ell$.

[4] Note that we cannot simply presume the ℓ_i so extracted by *Ext* in this step are all equal; indeed, they need not be. We shall address this issue in the sequel.

Simulatable: From Theorem 4.4, there exists a special simulator *Sim* for the honest-verifier for Protocol 4 that, on input $c \in \mathbb{Z}_p$, outputs simulated transcripts from the same distribution as real transcripts in Step 2. Given *Sim*, a special simulator for the full protocol is trivial: On input $(t, c) \in \mathbb{Z}_p \times \mathbb{Z}_p$, compute $\widetilde{E}(t) \leftarrow \prod_{i=1}^n (E_i)^{t^{i-1}}$ and $k \leftarrow \lceil \lg n \rceil$, and then invoke *Sim* with common inputs $\left(\widetilde{E}(t), t, k\right)$ and challenge c to obtain a simulated transcript for Step 2. □

5.1 Cost analysis

We now analyze the computation and communication cost of Protocol 5 on input a vector of length n. By inspection, we see that P sends just $5\lceil \lg n \rceil - 3$ commitments from $\mathbb{G}$ and a further $4\lceil \lg n \rceil - 2$ elements of $\mathbb{Z}_p$ as part of the sub-protocol in Step 2; hence, P uploads $\Theta(\lg n)$ group and $\Theta(\lg n)$ field elements. Meanwhile, V sends just two $\mathbb{Z}_p$ elements to P; hence, V uploads just $\Theta(1)$ field elements.

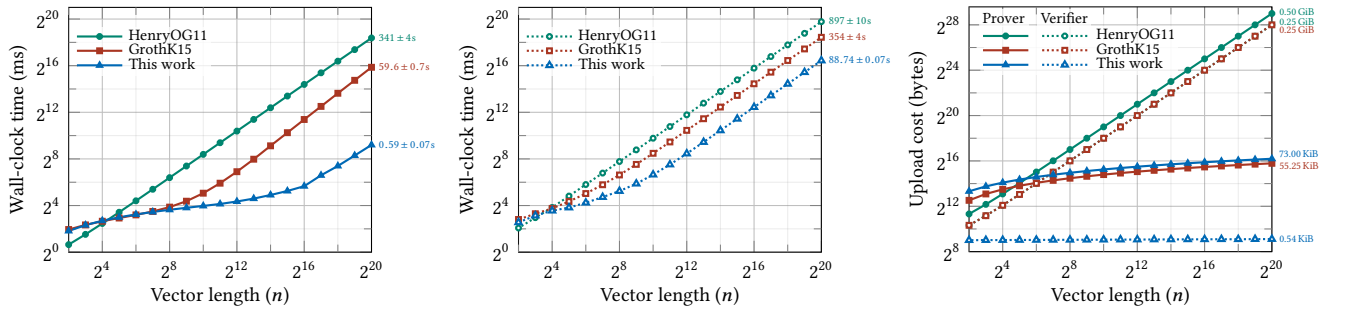
In terms of computational effort, the dominant (asymptotic) cost for P is the computation of γ in Step 2, a polynomial evaluation in $\mathbb{Z}_p[x]$ requiring $2n - 1$ operations in $\mathbb{Z}_p$ using Horner's method. Within the sub-protocol, P evaluates an additional $\Theta(\lg n)$ operations in each of $\mathbb{G}$ and $\mathbb{Z}_p$; thus, P evaluates $\Theta(n)$ field operations and $\Theta(\lg n)$ group operations. The dominant computation cost for V is the computation of $\widetilde{E}(t)$ in Step 1, which can be realized as a sequence of $n - 2$ multiplications in $\mathbb{Z}_p$ followed by an n-base multiexponentiation in $\mathbb{G}$.[5] Within the sub-protocol, V evaluates an additional $\Theta(\lg n)$ operations in each of $\mathbb{G}$ and $\mathbb{Z}_p$; thus, V evaluates $\Theta(n)$ field operations and $\Theta(n)$ group operations. A detailed summary of these costs—as well as analogous breakdowns for Henry et al.'s protocol (§2.1) and its variant with Groth and Kohlweiss' improvement (§2.2)—was located back in Table 1.

5.2 Trading speed for soundness

Referring back to Section 2.1 (and Section 2.2), we note that the verifier V in Henry et al.'s protocol (and in its variant with Groth and Kohlweiss' improvement) need not select the vector $\vec{R}$ and challenge c uniformly from *all* of $\mathbb{Z}_p^n$ and $\mathbb{Z}_p$. Rather, V may select $\vec{R} \in_{\mathrm{R}} [0..2^\lambda]^n$ and $c \in_{\mathrm{R}} [0..2^\lambda]$ for some $\lambda < \lg p$ as a means to trade somewhat-increased soundness error (going from $2^{-\lg p}$ up to $2^{-\lambda}$) in exchange for lower upload cost (from $\Theta(n)$ field elements down to $\Theta(n \cdot \lambda)$ bits), as well as a cheaper n-base multiexponentiation to compute $\widetilde{E}(\vec{R})$ and cheaper exponentiations to compute the sequence of $\widetilde{E}(\vec{R})/g^{\vec{R}[i]}$ (both going from $\Theta(n \cdot \lg p)$ down to $\Theta(n \cdot \lambda)$ group operations). This technique, sometimes referred to as *small-exponent batching* [3], can yield substantial speedups while still maintaining adequate soundness in practice.

We can likewise reduce the size of V's challenges t and c in the new protocol; however, the potential speedups from doing so are decidedly less profound than in the case of Henry et al.'s protocol. Specifically, the n-base multiexponentiation in the new protocol uses exponents from $\vec{t} := \langle 1, t, t^2, \ldots, t^{n-1} \rangle$, a vector whose entries quickly grow to $\lg p$ bits, even when t is small. The new protocol

[5] One might be tempted to compute $\widetilde{E}(t)$ using "Horner's method in the exponent"; however, it is not difficult to verify that this approach incurs strictly greater computation cost than the multiexponentiation we count here.



(a) Prover computation **(b) Verifier computation** **(c) Communication**

Figure 1: Empirically measured computation times for the prover (Fig. 1a) and the verifier (Fig. 1b) and analytically computed communication costs (Fig. 1c) as the vector length ranges from $n = 2^1$ through $n = 2^{20}$. In all experiments, we chose challenges uniformly from $\mathbb{Z}_p$ so as to minimize the soundness error. The verifier upload costs in "HenryOG11" and "GrothK15" are identical, with the plot of the former (·····o·····) totally eclipsed by that of the latter (·····□·····) in Figure 1c. The value to the right of each trend line is a 99% confidence interval for the final (e.g., $n = 2^{20}$) data point.

must also contend with the $2^{2k} \approx n^2$ soundness loss (see Corollary 3.4), which implies that, to get soundness error less than $2^{-\lambda}$, V must choose $t \in_R [0\,..\,2^{\lambda+2k} - 1]$. The latter point serves to further stifle any nominal speedups V might hope to gain in the computation of $\vec{t}$ itself. Nevertheless, as we will soon see in Section 6.2, unless the soundness error is allowed to exceed about 2^{-30}, the new protocol still comfortably outperforms its competitors.

5.3 5-move with better soundness

It is possible to slightly improve on the soundness of the protocol with the introduction of one additional move. (In applications where the prover produces $\vec{E}$ in an implicit 0th move with the intention of following up with a proof that it, in fact, commits to a standard basis vector—such as the applications mentioned in Section 1—this extra move can piggyback onto that implicit initial move.) Essentially, the prover commits to the binary decomposition of ℓ prior to learning the initial challenge t, and those commitments are then transformed into appropriate powers of t by V as part of Step 1. Having P commit to ℓ prior to its learning t effectively commits it to a *specific* polynomial $V(x) - x^\ell$, which must have a root at t is V is to accept. This reduces the number of "bad" challenges t from up to about 2^{2k} to at most 2^k, allowing to reduce the soundness error by a factor n or, alternatively, to reduce the size of t (for a fixed soundness error) by $\lg n$ bits.

6 IMPLEMENTATION & EVALUATION

In order to empirically validate the efficiency of Protocol 5, we have implemented it in C++ together with the algorithms of Henry et al. (§2.1) and its variant with Groth and Kohlweiss' improvement (§2.2).[6] We herein refer to our implementations of Protocol 5 as "This work", to our implementation of Henry et al.'s protocol as "HenryOG11", and to our implementation of Henry et al.'s protocol with Groth and Kohlweiss' improvement as "GrothK15", where the underline colors reference the corresponding plot colors in Figures 1 and 2. Our implementations leverage version 3.11.2 of the NTL library [22] for multiprecision arithmetic and the latest (at time of writing—see citation for the precise commit digest) developer build of the RELIC toolkit [1] for elliptic curve arithmetic. All experiments use `Curve P-256`, a NIST-standardized elliptic curve of 256-bit prime order $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ for the commitment group $\mathbb{G}$.

We conducted all of our experiments on a workstation running Ubuntu 18.04.2 LTS atop an AMD Ryzen 7 2700x eight-core CPU @ 4.30 GHz with 16 GiB of RAM. Our code was compiled using version `7.4.0-1ubuntu~18.04.1` of `g++` with both the `-O3` and `-march=native` compiler flags enabled.

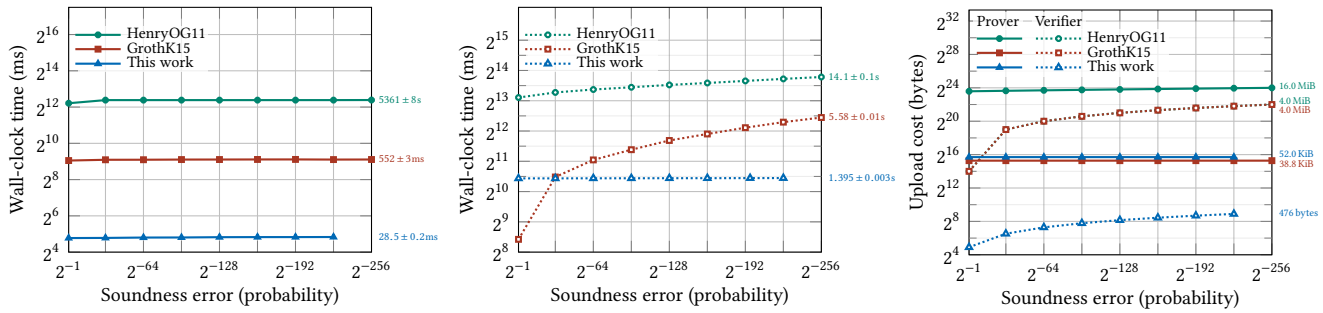
Each of our experiments measures the (wall-clock) running times for a prover and verifier each running in isolation, single-threaded on a single CPU core. In lieu of a "warm up period", we repeated all experiments for 110 trials and then discarded the 10 trials with running times furthest from the sample mean (note that we discarded both unusually fast and unusually slow trials).[7] Our plots report the sample mean running timings across the 100 remaining trials. We note that all experiments exhibited sample standard deviations that were minuscule relative to the sample means, resulting in error bars that were, in all instances, far too small to be visible on the plots. When discussing selected statistics, we write $\mu \pm w$ to indicate a 99% confidence interval with sample mean μ and interval width w (i.e., $w \coloneqq 2.58\,s$ for sample standard deviation s). We report all such figures to one digit of precision in the interval width.

6.1 Running time versus vector length

Our first set of experiments measures the running time as the vector length increases from a minimum of $n = 2^1$ elements to a maximum of $n = 2^{20}$ elements. Plots of our findings are located in Figures 1a and 1b, while Figure 1c shows the (analytically computed) communications costs for the same. Our findings are in accordance with the theoretical analysis: "HenryOG11" incurs the lowest computation cost for the very shortest of vectors, but quickly overtakes "GrothK15" and "This work" by around $n = 2^4$, whereafter "This work" reigns supreme.

[6] Our implementation (of all three protocols) is available as free and open source software (under the MIT License) via Github [6].

[7] This outlier pruning had almost no impact on the running times reported herein.



(a) Prover computation **(b) Verifier computation** **(c) Communication**

Figure 2: Empirically measured computation times for the prover (Fig. 2a) and the verifier (Fig. 2b) and analytically computed communication costs (Fig. 2c) as the soundness error ranges from 2^{-1} down to 2^{-256} using a fixed vector length of $n = 2^{14}$. Plots for "This work" end with soundness error 2^{-224} because of the $2 \lg n = 28$ loss. The verifier upload costs in "HenryOG11" and "GrothK15" are identical, with the plot of the former (·o·) totally eclipsed by that of the latter (·□·) in Figure 2c. The value to the right of each trend line is a 99% confidence interval for the final (e.g., $n = 2^{20}$) data point.

By $n = 2^{20}$, the prover in "This work" runs about 100× faster than the "GrothK15" prover (590 ± 70 ms versus 59600 ± 700 ms), while the verifier in "This work" runs about 4× faster than the "GrothK15" verifier (88740 ± 70 ms versus 354000 ± 4000 ms). In terms of communication, the constant upload cost of the verifier in "This work" begins lower than the (linear) verifier upload costs in "GrothK15" and "HenryOG11", with the discrepancy growing in proportion to the vector length; meanwhile, the prover in "This work" uploads a small constant factor (about 1.3×) more than the "GrothK15" prover.

For vectors of length exceeding about $n = 2^{16}$, the (linear) cost of evaluating the degree-n polynomial $\beta(t)$ in Step 2 of Protocol 5 begins to dominate the overall running time for the prover in "This work" and the slope of its trendline begins to look (essentially) linear in n.

Finally, as is just barely perceptible from the two trendlines in Figure 2a, the "GrothK15" prover's computation cost grows asymptotically faster than that of the "HenryOG11" prover. Extrapolating (to an admittedly absurd degree), we estimate that the "GrothK15" prover would eventually overtake the "HenryOG11" prover for vector lengths in the vicinity of $n = 2^{80}$.

6.2 Running time versus soundness error

Our second set of experiments measured the running time for a fixed vector length $n = 2^{14}$ (an arbitrary choice) while varying the soundness error of the protocol from a minimum of (at most) 2^{-1} through to a maximum of (at most) 2^{-256}. The soundness error tuning was accomplished by varying the bitlengths of the elements of the random vector in the "HenryOG11" and "GrothK15" protocols and the point t in "This work", as well as the verifiers challenge c in all three protocols.[8] Plots of our findings are located in Figures 2a and 2b, while Figure 2c shows the (analytically computed) communications costs for the same. Notice that the plots for "This work" end with soundness error (at most) 2^{-224}. This is because the $2 \lg n = 28$-bit soundness loss limits that protocol to a soundness error of (at most) 2^{-242} in the best case.

We notice that the computation costs for "This work" are basically agnostic to the soundness parameter, while the "GrothK15" and "HenryOG11" verifiers both show nominal speedups when the soundness error is allowed to grow. For all three protocols, prover upload costs are independent of the soundness, while verifier upload costs scale linearly with it. For very small soundness errors, the "GrothK15" protocol is fastest; however, for all cryptographically suitable soundness errors (indeed, for any soundness error less than about 2^{-30}), "This work" is faster.

[8] For "HenryOG11", smaller c results in reduced prover upload, since the prover needs to send the challenges used in simulated proofs; for the other two protocols the impact of changing c is insignificant.

7 CONCLUSION

We proposed a new communication- and computation-efficient, 4-move special honest-verifier zero-knowledge proof of knowledge system with which a prover P can convince a verifier V of its ability to open a vector of Pedersen (or Pedersen-like) commitments to a so-called "one-hot" vector (i.e., to a vector from the standard orthonormal basis) over $\mathbb{Z}_p^n$. The new protocol offers superior asymptotics relative to existing protocols for proving one-hotness. We have implemented both our new protocol and its closest competitors from the literature; in accordance with our analytic results, experiments confirm that the new protocols handily outperform existing protocols for all but the shortest of vectors (roughly, for vectors with more than 16–32 elements).

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant Nos. 1718595 and 1565375 and the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN-2019-04821. The second author thanks Ian Goldberg, Nikita Borisov, and Thomas Yurek for thoughtful discussions out of which this work arose.

REFERENCES

[1] Diego F. Aranha and Conrado Porto Lopes Gouvêa. RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic [commit e56004] (June 2019).

[2] Stephanie Bayer and Jens Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. *IACR Cryptology ePrint Archive*, Report

2015/195 (March 2015).

[3] Mihir Bellare, Juan A. Garay, and Tal Rabin. Fast batch verification for modular exponentiation and digital signatures. In *Advances in Cryptology: Proceedings of EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 236–250, Espoo, Finland (June 1998).

[4] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *Proceedings of TCC 2016-B (Part II)*, volume 9986 of *LNCS*, pages 31–60, Beijing, China (October–November 2016).

[5] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *ITCS 2012*, pages 326–349, Cambridge, MA, USA (January 2012).

[6] William Black and Ryan Henry. ZKP-onehot. https://github.com/WillBlack403/ZKP-onehot (August 2019).

[7] Dan Boneh, Eyal Kushilevitz, Rafail Ostrovsky, and William E. Skeith III. Public key encryption that allows PIR queries. In *Advances in Cryptology: Proceedings of CRYPTO 2007*, volume 4622 of *LNCS*, pages 50–67, Santa Barbara, CA, USA (August 2007).

[8] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In *Proceedings of ESORICS 2015, (Part I)*, volume 9326 of *LNCS*, pages 243–265, Vienna, Austria (September 2015).

[9] Stefan Brands, Liesje Demuynck, and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In *Proceedings of ACISP 2007*, volume 4586 of *LNCS*, pages 400–415, Townsville, Australia (July 2007).

[10] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (Extended abstract). In *Advances in Cryptology: Proceedings of CRYPTO 1997*, volume 1294 of *LNCS*, pages 410–424, Santa Barbara, CA, USA (August 1997).

[11] Henry Corrigan-Gibbs, Dan Boneh, and David Mazières. Riposte: An anonymous messaging system handling millions of users. In *Proceedings of IEEE S&P 2015*, pages 321–338, San Jose, CA, USA (May 2015).

[12] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology: Proceedings of CRYPTO 1994*, volume 839 of *LNCS*, pages 174–187, Santa Barbara, CA, USA (August 1994).

[13] Ivan Damgård, Ji Luo, Sabine Oechsner, Peter Scholl, and Mark Simkin. Compact zero-knowledge proofs of small hamming weight. In *Proceedings of PKC 2018, (Part II)*, volume 10770 of *LNCS*, pages 530–560, Rio de Janeiro, Brazil (March 2018).

[14] Rosario Gennaro, Darren Leigh, Ravi Sundaram, and William S. Yerazunis. Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices. In *Advances in Cryptology: Proceedings of ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 276–292, Jeju Island, South Korea (December 2004).

[15] Matthew Green, Watson Ladd, and Ian Miers. A protocol for privately reporting ad impressions at scale. In *Proceedings of CCS 2016*, pages 1591–1601, Vienna, Austria (October 2016).

[16] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *Advances in Cryptology: Proceedings of EUROCRYPT 2015 (Part II)*, volume 9057 of *LNCS*, pages 253–280, Sofia, Bulgaria (April 2015).

[17] Ryan Henry and Ian Goldberg. Batch proofs of partial knowledge. In *Proceedings of ACNS 2013*, volume 7954 of *LNCS*, pages 502–517, Banff, AB, Canada (June 2013).

[18] Ryan Henry, Yizhou Huang, and Ian Goldberg. One (block) size fits all: PIR and SPIR with variable-length records via multi-block queries. In *Proceedings of NDSS 2013*, San Diego, CA, USA (February 2013).

[19] Ryan Henry, Femi Olumofin, and Ian Goldberg. Practical PIR for electronic commerce. In *Proceedings of CCS 2011*, pages 677–690, Chicago, IL, USA (October 2011).

[20] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology: Proceedings of CRYPTO 1991*, volume 576 of *LNCS*, pages 129–140, Santa Barbara, CA, USA (August 1991).

[21] Kun Peng, Colin Boyd, and Ed Dawson. Batch zero-knowledge proof and verification and its applications. *ACM Transactions on Information and System Security (TISSEC)*, 10(2):Article No. 6 (May 2007).

[22] Victor Shoup. NTL: A Library for doing Number Theory [version 11.3.2]. https://www.shoup.net/ntl/ (November 2018).