

Understanding Sigma Protocols via MPC-in-the-Head

Haiyang Xue

Singapore Management University
haiyangxue@smu.edu.sg haiyangxc@gmail.com

Abstract. In this paper, we propose an understanding of classical Sigma protocols (as denoted by Σ -protocols in the literature) through the lens of the MPC-in-the-head (MPCitH) paradigm. We seek not to construct new schemes, but to build a formal bridge connecting these two foundational areas. While many elegant yet foundational Sigma-protocols were introduced decades ago as standalone algebraic or combinatorial interactive proofs, reconsidering them via the modern aspect of (extended) MPCitH significantly enhances our cryptographic insight into their underlying structures.

We present a formal mapping that translates classical commitment-challenge-and-response into the internal views of virtual parties, communicating via private point-to-point or public broadcast channels, secret-sharing schemes, commitments, and MPC consistency/correctness checks. By deconstructing algebraic schemes as linear secret-sharing protocols over public channels and combinatorial schemes as discrete threshold MPC instances over private/public channels, we isolate the exact structural reasons why their soundness and efficiency characteristics diverge. The protocols include:

- Algebraic schemes: Schnorr, Okamoto, Chaum-Pedersen, Guillou-Quisquater, GMR for QR and discrete log over an unknown-order group.
- Combinatorial schemes: Stern, GMW for Graph 3-Coloring, and Graph Isomorphism, among others.

Although reinterpreting these already elegant, classic proofs using a heavy-weight machinery like MPCitH might seem redundant, it deepens our understanding of their underlying design principles.

Table of Contents

Understanding Sigma Protocols via MPC-in-the-Head	1
<i>Haiyang Xue</i>	
1 Introduction	3
1.1 Motivation and Our Contribution	3
1.2 Organization	5
1.3 Related Work	5
2 Zero Knowledge Proof and Sigma Protocols	5
3 MPCitH Framework in [IKOS09]	7
3.1 Multi-Party Computation	8
3.2 MPCitH over Private Point-to-Point Channels	10
3.3 MPCitH over Public Broadcast Channels	11
3.4 Soundness, Special Soundness and Proofs of Knowledge	12
4 Algebraic Sigma protocols	15
4.1 MPCitH with Implicit View Commitments	15
4.2 Schnorr [Sch91]	16
4.3 Batch Schnorr [GLSY04]	17
4.4 Okamoto [Oka92]	18
4.5 Chaum-Pedersen [CP92] and Lossy ID	19
4.6 Guillou-Quisquater [GQ88]	20
4.7 GMR for Quadratic Residuosity [GMR89]	21
4.8 DL in Unknown Order Groups [PS98, GPS06]	22
5 Combinatorial Sigma Protocols	23
5.1 Stern's Protocol [Ste93]	23
5.2 Graph 3-Coloring [GMW91, GMW86]	25
5.3 Graph Isomorphism [GMW91, GMW86]	27
6 Discussion and Further Instances	29
6.1 Isogeny-Based Identification	29
6.2 Blum's Hamiltonian-Cycle Protocol	29
6.3 Further Directions	30

1 Introduction

Zero-knowledge proofs [GMR89] are fundamental building blocks of modern cryptography. A zero-knowledge proof lets a prover convince a verifier that a statement is true without revealing anything else, in particular the witness. A proof of knowledge [GMW86, BG92] further guarantees that a successful prover knows a witness, in the sense that one can be extracted from it.

Among these primitives, Sigma protocols (Σ -protocols), three-move public-coin proofs of knowledge, play a central role because of their simplicity, efficiency and elegant structure. Since the pioneering works of Schnorr [Sch91], Guillou and Quisquater [GQ88], Goldreich, Micali and Wigderson [GMW86], and Stern [Ste93], many Sigma protocols have been proposed, and several are collected in the textbook [BS23]. They are traditionally divided into two families.

- *Algebraic protocols* work over groups of prime or unknown order and prove knowledge of a preimage under a group homomorphism. Examples include the protocols of Schnorr [Sch91], Okamoto [Oka92], Chaum and Pedersen [CP92], and Guillou and Quisquater [GQ88]. Maurer [Mau15] unified many of them as proofs of knowledge of preimages of group homomorphisms.
- *Combinatorial protocols* prove knowledge of solutions to coding-theoretic or graph-theoretic problems, such as syndrome decoding [Ste93], graph 3-coloring and graph isomorphism [GMW86, GMW91], and Hamiltonian cycles [Blu86]. They rely on explicit commitments, random permutations and cut-and-choose.

Separately, Ishai, Kushilevitz, Ostrovsky and Sahai [IKOS09] introduced the MPC-in-the-head (MPCitH) paradigm, which compiles a multi-party computation (MPC) protocol, secure against a small number of semi-honest parties, into a zero-knowledge proof. The prover runs an N -party protocol in its head on a secret sharing of the witness, commits to the views of the virtual parties, and opens some of them in response to the verifier’s challenge; the verifier checks that the opened views are consistent. MPCitH and its descendant VOLE-in-the-head have become the main engines of post-quantum signatures such as Picnic, SDitH, MQOM and FAEST. The paradigm is nevertheless usually viewed as a general-purpose compiler for circuit-based relations, distinct from the classical Sigma protocols.

1.1 Motivation and Our Contribution

Our primary objective is **not to** introduce new constructions, **but rather to** establish a formal bridge between classical Sigma protocols and MPCitH, thereby providing understanding on both sides. While these connections may have been informally understood or implicitly assumed in folklore, our goal is to formalize and systematize them. As a secondary benefit, we aim to make these concepts accessible to beginners.

At first sight, explaining lightweight classical protocols with the heavier machinery of MPCitH may seem unnecessary. The two topics, however, are usually described in disjoint vocabularies: algebraic Sigma protocols through group homomorphisms [Mau15], combinatorial ones through ad hoc cut-and-choose, and modern MPCitH through secret sharings, broadcast transcripts, seed trees and all-but-one commitments. We show that the classical protocols are early, special-purpose instances of MPCitH. Each is determined by three choices: the channel model of the virtual MPC protocol, the secret sharing of the witness, and the sets of parties the verifier may open. Together, these choices determine its soundness error.

- **The IKOS framework.** We recall MPC-in-the-Head as in [IKOS09], with generalized secret sharing, private point-to-point channels, a broadcast channel and ideal two-party functionalities (Section 3.1). We describe two compilers, one over private channels and one over a broadcast channel. Each is parameterized by an *opening family*, the sets of parties the verifier may open, and we prove their zero-knowledge (Sections 3.2 and 3.3).

We then prove special soundness for all compilers by a single argument (Section 3.4), making precise the proof-of-knowledge property that [IKOS09] state without proof. The special-soundness parameter is a covering number of the opening family and depends on whether a fault is attributed to one party (vertex checks) or only to a pair (edge checks): opening ℓ of N parties gives knowledge error $1 - \ell/N$ in the first case and $1 - \ell(\ell - 1)/(N(N - 1))$ in the second, recovering the bounds of [IKOS09, KKW18, BN20, GMO16].

- **Algebraic protocols.** Section 4 introduces MPCitH with *implicit view commitments* (Section 4.1). The witness is shared by a polynomial of degree d and one party is opened. Through a homomorphism, the first message of the prover, together with the statement, determines a commitment to the view of each of exponentially many parties. Schnorr’s protocol [Sch91], batch Schnorr [GLSY04], Okamoto’s protocol [Oka92] and Chaum–Pedersen [CP92] are instances. Okamoto’s protocol shows the role of a non-trivial kernel (many witnesses, witness indistinguishability), and Chaum–Pedersen that of a non-trivial cokernel (soundness for a language and lossy identification [KW03, AFLT12]). Over groups of unknown order the sharing becomes multiplicative, as in Guillou–Quisquater [GQ88] and the quadratic-residuosity protocol of [GMR89], or integral, as in GPS [GPS06], and the number of parties is limited by reconstruction.
- **Combinatorial protocols.** Section 5 reads Stern’s protocol [Ste93] as a three-party protocol over private channels whose parties interact through two-party functionalities. It reads the GMW 3-coloring protocol [GMW91] as a protocol whose network is the graph itself, and the GMW graph-isomorphism protocol [GMW91] as a two-party protocol over a broadcast channel.

How the protocols differ. Table 1 summarizes the protocols of Sections 4 and 5 along four dimensions.

- *Channel.* Over a public channel, each opened party is checked against public data, either a broadcast transcript or a homomorphic commitment; these are vertex checks. Over private point-to-point channels, possibly carrying two-party functionalities, a check involves two opened views; these are edge checks. All protocols of Section 4 and graph isomorphism use public channels and implicit commitments, while Stern’s and the 3-coloring protocol use private channels and explicit commitments.
- *Secret sharing.* The algebraic protocols share the witness linearly: by a polynomial of degree one (or d) over $\mathbb{Z}_q$, multiplicatively in a group of unknown order, or over the integers. The combinatorial protocols mask the witness with a random symmetry of the relation, a permutation of coordinates, vertices or colors, so that an opened set reveals only what that symmetry leaves invariant.
- *Opening.* The privacy of the sharing dictates which parties may be opened: a single party in Section 4 and in graph isomorphism, two of three parties in Stern’s protocol, and the two endpoints of an edge in the 3-coloring protocol.
- *Error.* The soundness or knowledge error is $(k-1)/|\mathcal{C}|$, where k is the special-soundness parameter and $\mathcal{C}$ the challenge space (Proposition 1).

Why the errors diverge. The errors in Table 1 are governed by three structural parameters, none of which is specific to algebra or to combinatorics.

- *Fault attribution.* A prover who commits to views for a false statement must make them inconsistent somewhere. With vertex checks, a fault belongs to one party. With edge checks, it may belong only to an edge, and hiding either endpoint hides it. Under all-but-one opening, this is the difference between $1/N$ and $2/N$ (Corollary 1).
- *Affordable parties.* Explicit commitments cost one commitment per party, which keeps N small, or tied to the size of the instance as in the 3-coloring protocol, and at most a few hundred in modern MPCitH signatures. Implicit commitments cost nothing per party: in Schnorr’s protocol, the single element $a = g^r$ fixes the commitment $C_c = a \cdot y^c$ of each of q parties (Theorem 4).
- *Reconstruction.* When the public data fix the whole sharing, two opened parties may already reconstruct the witness, so a single opening suffices. The number of parties is then the size of the largest set of evaluation points from which any two parties reconstruct: q over $\mathbb{Z}_q$, e for Guillou–Quisquater, and only 2 for quadratic residuosity and GPS.

These parameters also show that the traditional split does not follow the structure. Graph isomorphism, usually classed as combinatorial, is a group-action protocol with the structure of Section 4, shared by isogeny-based identification (Sections 5.3 and 6.1). Conversely, the quadratic-residuosity protocol is algebraic yet limited to two parties. Finally, they describe how modern MPCitH improves on the classical combinatorial protocols: it keeps explicit commitments but switches to vertex checks over a broadcast channel, and uses seed trees to make hundreds of parties affordable (Section 6.3).

Table 1. The protocols of Sections 4 and 5 as MPC-in-the-head. *Sec.*: the subsection treating the protocol. *Secret sharing*: for a polynomial $f(X)$, party P_c holds $f(c)$; otherwise the tuple of shares is listed; the second entry is the ring or group of the shares. GQ: Guillou–Quisquater; QR: quadratic residuosity; GPS: Girault–Poupard–Stern. *N*: number of virtual parties. *Open*: number of opened parties, or the two endpoints of an edge. *Channel*: public channels give vertex checks and here come with implicit commitments; private point-to-point channels, which carry two-party functionalities, give edge checks and come with explicit commitments. *k*: special-soundness parameter. *Error*: soundness or knowledge error, $(k - 1)/|C|$. In prime-order groups every Schnorr-type statement is true, so the error is a knowledge error; Chaum–Pedersen also covers lossy identification, and GPS is listed with binary challenges.

Protocol	Sec.	Secret sharing	N	Open	Channel	k	Error
<i>Algebraic protocols</i>							
Schnorr [Sch91]	4.2	$r + wX, \mathbb{Z}_q$	q	1	public	2	$1/q$
Batch Schnorr [GLSY04]	4.3	$r + \sum_k w_k X^k, \mathbb{Z}_q$	q	1	public	$d + 1$	d/q
Okamoto [Oka92]	4.4	$r + wX, \mathbb{Z}_q^2$	q	1	public	2	$1/q$
Chaum–Pedersen [CP92]	4.5	$r + wX, \mathbb{Z}_q$	q	1	public	2	$1/q$
GQ [GQ88]	4.6	$x_t x^X, \mathbb{Z}_n^*$	e	1	public	2	$1/e$
QR [GMR89]	4.7	$(x_t, x_t x), \mathbb{Z}_n^*$	2	1	public	2	$1/2$
GPS [GPS06]	4.8	$r + xX, \mathbb{Z}$	2	1	public	2	$1/2$
<i>Combinatorial protocols</i>							
Stern [Ste93]	5.1	$(\pi, \pi(r), \pi(x + r))$	3	2	private	3	$2/3$
3-coloring [GMW91]	5.2	$(\sigma(\chi(v)))_{v \in V}$	$ V $	edge	private	$ E $	$1 - 1/ E $
Graph iso. [GMW91]	5.3	$(\sigma, \sigma\pi^{-1}), S_n$	2	1	public	2	$1/2$

1.2 Organization

Section 2 recalls zero-knowledge proofs and Sigma protocols. Section 3 presents the framework: the MPC model (Section 3.1), the compilers over private and broadcast channels with their zero-knowledge (Sections 3.2 and 3.3), and soundness, special soundness and proofs of knowledge (Section 3.4). Section 4 treats the algebraic protocols, starting from the template with implicit view commitments, first over groups of prime order and then over groups of unknown order. Section 5 treats Stern’s protocol, graph 3-coloring and graph isomorphism. Section 6 discusses isogeny-based identification, Blum’s Hamiltonian-cycle protocol and further directions.

1.3 Related Work

Two works are especially close to our perspective. Maurer [Mau15] unified algebraic Sigma protocols as proofs of knowledge of preimages of group homomorphisms, and Zhang et al. [ZCYW23] derived Sigma protocols from verifiable secret sharing (VSS). Both focus on algebraic protocols. Zhang et al. [ZCYW23] already note that setting the privacy threshold of their VSS-based construction to 1 recovers Schnorr’s protocol, and they also cover Guillou–Quisquater, Okamoto and batch Schnorr; their aim is to simplify the composition of algebraic proofs with circuit-based proofs. Our reading of the algebraic protocols is closest to theirs, with Feldman-type implicit commitments, and Maurer’s conditions for groups of unknown order reappear in it as conditions on reconstruction. Our goal, however, is a single framework that covers algebraic and combinatorial protocols alike and explains their soundness errors through the channel model, the sharing and the opening family.

On the MPCitH side, Feneuil and Rivain [FR23] use Shamir sharings with few opened parties. Schnorr-type protocols are the special case of a linear relation, where implicit commitments remove the need to commit to the views explicitly.

2 Zero Knowledge Proof and Sigma Protocols

Throughout this work, since our focus is Sigma protocols, we consider zero-knowledge proofs of knowledge (ZKPoK) for an NP-relation $\mathcal{R}$. That is, beyond convincing the verifier of statement validity, the protocol guarantees that an efficient prover must explicitly possess a valid witness w , formalized via a knowledge extractor.

Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ be an NP-relation consisting of statement-witness pairs (x, w) . The language associated with $\mathcal{R}$ is $\mathcal{L}_{\mathcal{R}} = \{x \mid \exists w \text{ s.t. } (x, w) \in \mathcal{R}\}$.

Definition 1 (Interactive Proof System [GMR89]). An interactive proof system for a language $\mathcal{L}_{\mathcal{R}}$ is a pair of interactive Turing machines $(\mathcal{P}, \mathcal{V})$, where $\mathcal{V}$ is probabilistic polynomial-time (PPT) and $\mathcal{P}$ has unbounded computational power (or holds a secret witness w). The protocol proceeds in rounds of message exchanges, ending with $\mathcal{V}$ outputting 1 (accept) or 0 (reject). $(\mathcal{P}, \mathcal{V})$ is an interactive proof for $\mathcal{L}_{\mathcal{R}}$ with soundness error ϵ if:

1. **Completeness:** For every $x \in \mathcal{L}_{\mathcal{R}}$, $\Pr[\langle \mathcal{P}(w), \mathcal{V} \rangle(x) = 1] = 1$. (We may relax the perfect completeness requirement to allow a negligible error probability)
2. **Soundness:** For every $x \notin \mathcal{L}_{\mathcal{R}}$ and every (possibly malicious) prover strategy $\mathcal{P}^*$,

$$\Pr[\langle \mathcal{P}^*, \mathcal{V} \rangle(x) = 1] \leq \epsilon(|x|).$$

If $\mathcal{V}$'s messages consist solely of uniform random bits drawn independently of prior messages, $(\mathcal{P}, \mathcal{V})$ is called a public-coin¹ interactive proof system [Bab85]. In this paper, all the protocols are public-coin.

Remark 1 (Soundness Error). The soundness error of an interactive proof can be negligible (e.g., 2^{-128}) or constant (e.g., $1/2$ or $1/3$). For a protocol with perfect completeness, the soundness error can generally be reduced from ϵ to ϵ^k by executing the protocol k times sequentially and rejecting if any single execution fails. Furthermore, looking ahead, for public-coin Σ -protocols, this error reduction to ϵ^k holds even under parallel repetition.

Remark 2 (Interactive Argument vs. Proof System). When the soundness condition is restricted to hold only against computationally bounded (PPT) prover strategies $\mathcal{P}^*$, the protocol is called an *interactive argument system* (or computationally sound proof system) introduced by Brassard, Chaum, and Crépeau in 1986 [BCC88]. Unlike interactive proofs, argument systems can utilize cryptographic primitives under computational assumptions. Correspondingly:

- In *interactive proofs* (such as GMW graph proofs [GMW86], Schnorr [Sch91] or Stern [Ste93] with a statistically binding commitment), soundness is unconditional/statistical and holds even against an infinitely powerful malicious prover.
- In *interactive arguments*, soundness relies on the computational hardness of an underlying assumption and thus restricts $\mathcal{P}^*$ to PPT algorithms.

Definition 2 (Zero-Knowledge Proof (or Argument) of Knowledge [GMR89, BG92]). An interactive proof (resp. argument) system $(\mathcal{P}, \mathcal{V})$ is a Zero-Knowledge Proof (resp. Argument) of Knowledge for an NP-relation $\mathcal{R}$ if it satisfies:

1. **Knowledge Soundness:** There exists a probabilistic polynomial-time (PPT) knowledge extractor $\mathcal{E}$ and a polynomial $p(\cdot)$ such that for any prover strategy $\mathcal{P}^*$ (restricted to PPT for argument systems) that convinces $\mathcal{V}(x)$ with probability $\epsilon(x) > \kappa(x)$, the extractor $\mathcal{E}^{\mathcal{P}^*}(x)$ outputs a valid witness w such that $(x, w) \in \mathcal{R}$ in expected time bounded by $\frac{p(|x|)}{\epsilon(x) - \kappa(x)}$.
2. **Zero-Knowledge:** For every PPT verifier $\mathcal{V}^*$, there exists a PPT simulator $\mathcal{S}$ such that the execution view $\text{View}_{\mathcal{V}^*}[\langle \mathcal{P}(x, w), \mathcal{V}^*(x) \rangle]$ is computationally (statistically or perfectly) indistinguishable from $\mathcal{S}^{\mathcal{V}^*}(x)$.

The knowledge soundness error $\kappa(x)$ may be negligible, or a constant such as $1/2$ or $2/3$, as for most protocols in this paper before repetition.

We refer the reader to Goldreich's Foundations of Cryptography: Volume 1 [Gol01, Chapter 4] for further details.

Sigma (or Σ -) protocol was first defined in [CDS94] and named in [Cra97]. A Sigma protocol is a three-move public-coin zero-knowledge proof of knowledge between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for an NP-relation $\mathcal{R}$. For a shared statement x and secret witness w known to $\mathcal{P}$, the execution flow is:

1. **Commitment (a):** $\mathcal{P}$ computes a commitment $a \leftarrow \mathcal{P}(x, w; r)$ using randomness r and sends a to $\mathcal{V}$.
2. **Challenge (c):** $\mathcal{V}$ samples a challenge $c \xleftarrow{\$} \mathcal{C}$ uniformly at random from a predefined challenge space $\mathcal{C}$ and sends c to $\mathcal{P}$.

¹ The public-coin means any coin-toss is visible to the prover when it is tossed. In the case of private randomness, the verifier's randomness is not visible to the prover.

3. *Response (z):* $\mathcal{P}$ computes a response $z \leftarrow \mathcal{P}(x, w, r, c)$ and sends z to $\mathcal{V}$.
4. *Verification:* $\mathcal{V}$ evaluates a deterministic public predicate $\mathcal{V}(x, a, c, z) \in \{0, 1\}$.

Definition 3 (Σ -Protocol). An interactive proof system $(\mathcal{P}, \mathcal{V})$ for an NP-relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ is a Σ -protocol if it is a three-move, public-coin protocol, comprising commitment a , random challenge $c \xleftarrow{\$} \mathcal{C}$, and response z , that satisfies the following properties:

1. **Completeness:** If $(x, w) \in \mathcal{R}$, an honest verifier $\mathcal{V}(x)$ always accepts the transcript (a, c, z) produced by $\mathcal{P}(x, w)$.
2. **k -Special Soundness:** A Sigma protocol satisfies k -special soundness ($k \geq 2$) if there exists a deterministic polynomial-time extractor $\mathcal{E}$ that, given statement x and k valid transcripts $(a, c_1, z_1), \dots, (a, c_k, z_k)$ with identical commitment a and distinct challenges $c_i \neq c_j$, outputs a witness w such that $(x, w) \in \mathcal{R}$.
3. **Special Honest-Verifier Zero-Knowledge (Special HVZK):** There exists a PPT simulator $\mathcal{S}$ that, given statement x and any challenge $c \in \mathcal{C}$, outputs a transcript (a, c, z) distributed (computationally or perfectly) identically to an interactive transcript between honest $\mathcal{P}(x, w)$ and $\mathcal{V}(x)$.

Remark 3. In this work, while we use the traditional term “ Σ -protocol” following standard terminology, we explicitly account for both statistically sound Σ -proofs and computationally sound Σ -arguments within our generalized framework.

3 MPCitH Framework in [IKOS09]

Introduced by Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS09], the *MPC-in-the-Head* (MPCitH) paradigm is a general compiler that turns a multi-party computation (MPC) protocol, secure against a small number of semi-honest parties, into a zero-knowledge proof (or argument).

Core idea. To prove knowledge of a witness w with $\mathcal{R}(x, w) = 1$, the prover runs an N -party MPC protocol Π locally, “in its head”. The witness is secret-shared as $(w_1, \dots, w_N)$ among N virtual parties $P_1, \dots, P_N$, who jointly compute

$$f(x, w_1, \dots, w_N) = \mathcal{R}(x, w_1 \oplus \dots \oplus w_N).$$

Ishai et al. use this XOR sharing, but the construction does not depend on it. The analysis only needs every N -tuple of shares to determine a witness candidate, so additive sharing over any finite field or group works equally well. (Threshold variants based on Shamir sharing are studied in [FR23].) The proof then proceeds in three steps:

1. **Commitment:** The prover simulates Π to completion and sends individual commitments to the local view of each party P_i ($i \in [N]$) to the verifier.
2. **Challenge:** Upon receiving the commitments, the verifier selects and sends a random challenge specifying a subset (of size t) of parties to be inspected.
3. **Verification:** The prover opens the requested subset of views. The verifier accepts if and only if all revealed views are consistent with a valid execution of Π .

Roughly speaking, the completeness is simply implied by the correctness of the underlying MPC protocol. Soundness stems from the fact that if a cheating prover attempts to prove a false statement ($x \notin \mathcal{L}_{\mathcal{R}}$), at least one pair of views or state transitions (binding in the commitment) must be inconsistent due to the correctness of the underlying MPC. The inconsistency can be detected with some probability.

The zero-knowledge property of the proof follows directly from the hiding property of the underlying commitment scheme and the t -privacy of the MPC protocol, which guarantees the existence of a simulator capable of generating indistinguishable views for any coalition of at most t corrupted parties.

Ishai et al. [IKOS09] remark that their construction is also a proof of knowledge, but omit the proof for simplicity. In fact, the construction is special sound. Since Com is binding, the first message fixes the views of all N parties, including their input shares w_i . By rewinding the prover with the same

commitments but different challenges, an extractor can obtain the opening of enough views. Once every consistency check has passed, the views form an honest execution (Lemma 1), so w , which can be recovered from the secret sharing encoded in the views, is a valid witness. Section 3.4 makes this precise for each channel model and opening pattern. The number of transcripts the extractor needs ranges from two for broadcast with all-but-one opening, to $\binom{N}{2}$ for the pair opening of [IKOS09].

3.1 Multi-Party Computation

We recall the model of [IKOS09], in which a protocol is specified by its next-message function, and extend it with a broadcast channel.

Witness sharing. Let $\mathcal{R}$ be an NP-relation with witness space $\mathcal{W}$, and let $\mathcal{X}$ be a share space.

Definition 4 (Witness sharing). *A witness sharing for N parties consists of two algorithms. The probabilistic algorithm Share maps a witness $w \in \mathcal{W}$ to N shares $w_1, \dots, w_N \in \mathcal{X}$, and the deterministic function $\text{Rec} : \mathcal{X}^N \rightarrow \mathcal{W}$ is defined on every N -tuple and satisfies $\text{Rec}(\text{Share}(w)) = w$ for every w . It is t -private if, for every $w \in \mathcal{W}$ and every $T \subseteq [N]$ with $|T| \leq t$, the shares $(w_i)_{i \in T}$ output by $\text{Share}(w)$ are uniform over $\mathcal{X}^{|T|}$.*

XOR sharing over $\{0, 1\}^m$, used in [IKOS09], and additive sharing over a finite abelian group, with $\text{Rec}(w_1, \dots, w_N) = \sum_i w_i$, are both $(N - 1)$ -private. Because Rec is total, every N -tuple of shares determines a witness candidate; the soundness analysis of Section 3.4 relies on this. The functionality evaluated in the head is

$$f_{\mathcal{R}}(x, w_1, \dots, w_N) = \mathcal{R}(x, \text{Rec}(w_1, \dots, w_N)),$$

where the statement x is a public input, w_i is the private input of P_i , and the output bit is delivered to all parties.

Protocols and next-message functions. An N -party protocol Π runs in R synchronous rounds among parties $P_1, \dots, P_N$ that are connected by pairwise private channels and, possibly, a broadcast channel. Each P_i holds the public input x , a private input $w_i \in \mathcal{X}$ and a random tape $r_i \in \{0, 1\}^\ell$. In round $\rho \in [R]$, party P_i sends a message $m_{i \rightarrow j}^{(\rho)}$ to each P_j with $j \neq i$ and may broadcast a message $b_i^{(\rho)}$.

Following [IKOS09, Sect. 3], each pair of parties P_i, P_j with $i < j$ may in addition have access to an ideal deterministic two-party functionality $g_{i,j}$, such as oblivious transfer (OT). In round ρ , P_i and P_j then call $g_{i,j}$ once, on inputs $\alpha_{i,j}^{(\rho)}$ from P_i and $\alpha_{j,i}^{(\rho)}$ from P_j , and receive

$$(\beta_{i,j}^{(\rho)}, \beta_{j,i}^{(\rho)}) = g_{i,j}(\alpha_{i,j}^{(\rho)}, \alpha_{j,i}^{(\rho)}).$$

For OT with P_i as sender, $g_{i,j}((s_0, s_1), c) = (\perp, s_c)$. We denote by

$$M_i^{(\rho)} = \left((m_{j \rightarrow i}^{(\rho)})_{j \neq i}, (\beta_{i,j}^{(\rho)})_{j \neq i}, (b_j^{(\rho)})_{j \neq i} \right)$$

everything that P_i receives in round ρ .

Definition 5 (Next-message function). *A protocol Π is specified by a deterministic polynomial-time function, also denoted Π . For $0 \leq \rho < R$,*

$$\Pi(i, x, w_i, r_i, M_i^{(1)}, \dots, M_i^{(\rho)}) = \left((m_{i \rightarrow j}^{(\rho+1)})_{j \neq i}, (\alpha_{i,j}^{(\rho+1)})_{j \neq i}, b_i^{(\rho+1)} \right) \quad (1)$$

returns the messages that P_i sends and its inputs to the two-party functionalities in round $\rho + 1$, and for $\rho = R$ it returns the local output y_i of P_i .

In the *private-channel model* no party broadcasts, and in the *broadcast model* all communication is by broadcast; likewise, a protocol without two-party functionalities makes no calls. Accordingly, any component in (1) may be empty.

Definition 6 (View). *The view of P_i in an execution of Π is*

$$\text{View}_i = (w_i, r_i, M_i^{(1)}, \dots, M_i^{(R)}),$$

consisting of its private input, its random tape, and all messages and functionality outputs it receives.

Since Π is deterministic, repeated application of (1) shows that x and View_i determine every message sent by P_i , every input it gives to a two-party functionality, and its output. We write $m_{i \rightarrow j}^{(\rho)}(\text{View}_i)$, $\alpha_{i,j}^{(\rho)}(\text{View}_i)$, $b_i^{(\rho)}(\text{View}_i)$ and $\text{out}(\text{View}_i)$ for these values; the round- ρ values depend only on the part of View_i received before round ρ . This is what allows a verifier to check an opened view without re-running the other parties.

Definition 7 (Consistency). *Let $\text{View}_i = (w_i, r_i, M_i^{(1)}, \dots, M_i^{(R)})$ be a string parsed as a view of P_i .*

1. View_i is well formed if $w_i \in \mathcal{X}$, $r_i \in \{0, 1\}^\ell$, and each $M_i^{(\rho)}$ lies in the message space of round ρ .
2. Two well-formed views View_i and View_j , indexed so that $i < j$, are consistent, written $\text{View}_i \sim \text{View}_j$ or $\text{View}_j \sim \text{View}_i$, if the following hold for every $\rho \in [R]$ [IKOS09, Def. 2.2].
 - The message $m_{i \rightarrow j}^{(\rho)}(\text{View}_i)$ equals the message from P_i recorded in $M_j^{(\rho)}$, and $m_{j \rightarrow i}^{(\rho)}(\text{View}_j)$ equals the message from P_j recorded in $M_i^{(\rho)}$.
 - If the parties use a two-party functionality, their input–output pairs are consistent with it: the outputs $\beta_{i,j}^{(\rho)}$ and $\beta_{j,i}^{(\rho)}$ recorded in $M_i^{(\rho)}$ and $M_j^{(\rho)}$ equal $g_{i,j}(\alpha_{i,j}^{(\rho)}(\text{View}_i), \alpha_{j,i}^{(\rho)}(\text{View}_j))$.
3. Let $T_{\text{bc}} = (b_k^{(\rho)})_{k \in [N], \rho \in [R]}$ be a broadcast transcript. A well-formed view View_i is consistent with T_{bc} if the broadcasts recorded in View_i are those of T_{bc} , and $b_i^{(\rho)}(\text{View}_i) = b_i^{(\rho)}$ for every $\rho \in [R]$.

OT channels make it possible to use 2-private protocols such as the 3-party GMW protocol [IKOS09, Sect. 3]. They are compatible with the compilers that check pairs of opened views, patterns (a) and (c) of Section 3.4, but not with pattern (d), which opens a single view together with its committed channels [IKOS09, Sect. 3]: an opened channel carrying an OT call would expose the other party's input to it.

Lemma 1 (Local versus global consistency [IKOS09, Lemma 2.3]). *Consider well-formed views View_i , $i \in [N]$, and, in the broadcast model, a broadcast transcript T_{bc} . Suppose that $\text{View}_i \sim \text{View}_j$ for all $i \neq j$ and, in the broadcast model, that every View_i is consistent with T_{bc} . Then there is an honest execution of Π , on x and on the inputs and random tapes recorded in the views, in which the view of each P_i is View_i and, in the broadcast model, the broadcast transcript is T_{bc} . The converse holds by definition.*

Proof. Run Π honestly on x and $(w_i, r_i)_{i \in [N]}$. We show by induction on ρ that the messages received in round ρ of this run agree with those recorded in the views. If this holds for all rounds before ρ , then by (1) the round- ρ messages sent in the run are $m_{i \rightarrow j}^{(\rho)}(\text{View}_i)$ and $b_i^{(\rho)}(\text{View}_i)$, and the inputs to $g_{i,j}$ are $\alpha_{i,j}^{(\rho)}(\text{View}_i)$ and $\alpha_{j,i}^{(\rho)}(\text{View}_j)$. Consistency states that these messages, and the outputs of $g_{i,j}$ on these inputs, are exactly the round- ρ values recorded in the views of their receivers.

Definition 8 (Correctness and privacy [IKOS09, Defs. 2.4 and 2.5]). *Let Π be an N -party protocol for a functionality f .*

1. Π realizes f with perfect correctness if, for all inputs $x, w_1, \dots, w_N$ and all random tapes $r_1, \dots, r_N$, the output of every party equals $f(x, w_1, \dots, w_N)$.
2. For $1 \leq t < N$, Π realizes f with perfect t -privacy if there is a PPT simulator Sim such that, for all inputs $x, w_1, \dots, w_N$ and every $T \subseteq [N]$ with $|T| \leq t$,

$$(\text{View}_i)_{i \in T} \equiv \text{Sim}(T, x, (w_i)_{i \in T}, f(x, w_1, \dots, w_N)),$$

where the left-hand side is taken over uniformly random tapes $r_1, \dots, r_N$. In the broadcast model, the left-hand side also includes T_{bc} , which Sim must output as well. Statistical and computational t -privacy are obtained by replacing $\equiv$ with statistical or computational indistinguishability.

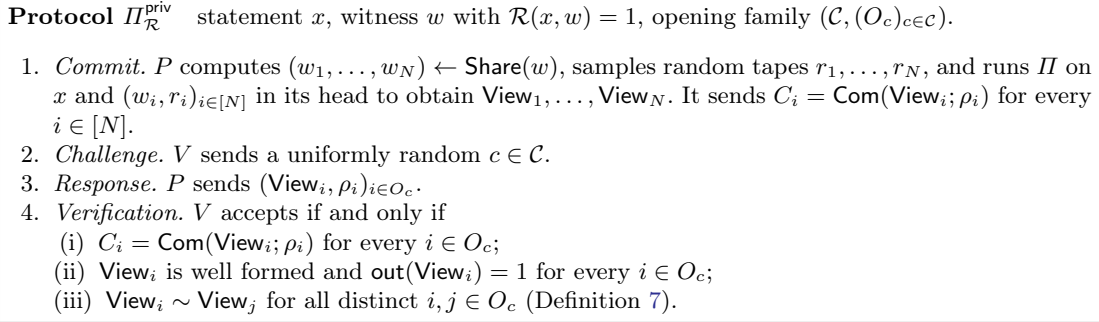


Fig. 1. MPCitH over private channels. With pair opening this is the basic construction of [IKOS09, Fig. 1]; with all-but-one opening it extends the three-party structure of ZKBoo [GMO16] to N parties.

Combining the two: if Π realizes $f_{\mathcal{R}}$ with perfect correctness and the views satisfy the hypotheses of Lemma 1, then every party outputs $\mathcal{R}(x, w)$ for the witness candidate $w = \text{Rec}(w_1, \dots, w_N)$. Every soundness argument in this paper rests on this fact.

Remark 4. The privacy threshold a compiler needs equals the number of views it opens: $t = 2$ for the pair opening of [IKOS09], $t = N - 1$ for all-but-one opening, and $t = 1$ for opening a single view with its incident channels (Section 3.4). The zero-knowledge simulator samples the opened input shares uniformly, which is possible when $(\text{Share}, \text{Rec})$ is t -private, and then runs Sim with output 1. Robustness against malicious parties is not needed for these compilers; it is needed only when a single execution must reach negligible soundness error by opening many views [IKOS09, Sect. 4].

Remark 5 (Public randomness). Several compilers let the verifier supply public random coins between rounds of Π , as in [IKOS09, Sect. 4.2] and in the five-round protocols of [BN20]. This is captured by giving (1) an additional public input γ that becomes available from a fixed round on. Perfect correctness is then relaxed to correctness except with some probability p over γ , and this false-positive probability enters the soundness error.

3.2 MPCitH over Private Point-to-Point Channels

We first describe the compiler of [IKOS09] for protocols in the private-channel model, possibly with two-party functionalities. It is parameterized by an *opening family*: a challenge space $\mathcal{C}$ together with a set $O_c \subseteq [N]$ of parties to be opened for each $c \in \mathcal{C}$. The basic compiler of [IKOS09] uses pair opening, in which $\mathcal{C}$ is the set of pairs $\{i, j\} \subseteq [N]$ with $i \neq j$, and $O_{\{i, j\}} = \{i, j\}$.

Let Com be a non-interactive commitment scheme. Statistical binding makes the result a proof rather than an argument (Section 3.4). The statistically binding commitment of Naor used in [IKOS09] needs a first message from the verifier; this adds one move and changes nothing below. The protocol is given in Figure 1. Note that the verifier does not check the opened shares against the witness: Rec is defined on every tuple, and the output check $\text{out}(\text{View}_i) = 1$ plays this role.

Committed channels. For 1-private protocols without two-party functionalities, [IKOS09, Sect. 3] give a variant in which the prover also commits, separately, to the transcript $(m_{i \rightarrow j}^{(\rho)}, m_{j \rightarrow i}^{(\rho)})_{\rho \in [R]}$ of each of the $\binom{N}{2}$ channels. The verifier samples $i \leftarrow [N]$; the prover opens View_i and the $N - 1$ channels incident to P_i ; and the verifier checks that View_i is well formed, outputs 1, and agrees with every opened channel, both on the messages it records as received and on the messages it determines as sent. We call this *single opening with committed channels*.

Theorem 1 (Zero-knowledge over private channels). *Let $t = \max_c |O_c|$. Suppose that $(\text{Share}, \text{Rec})$ is t -private, that Π realizes $f_{\mathcal{R}}$ with perfect correctness and (perfect, statistical or computational) t -privacy, and that Com is computationally hiding. Then $\Pi_{\mathcal{R}}^{\text{priv}}$ is perfectly complete and computationally special honest-verifier zero-knowledge. The same holds for single opening with committed channels, with $t = 1$.*

Proof. Completeness. If $\mathcal{R}(x, w) = 1$, the views come from an honest execution on a sharing of w . They are therefore well formed and pairwise consistent, and perfect correctness gives $\text{out}(\text{View}_i) = f_{\mathcal{R}}(x, w_1, \dots, w_N) = \mathcal{R}(x, w) = 1$.

Special HVZK. Given x and $c \in \mathcal{C}$, the simulator samples $(w_i)_{i \in O_c}$ uniformly from $\mathcal{X}^{|O_c|}$ and runs $(\text{View}_i)_{i \in O_c} \leftarrow \text{Sim}(O_c, x, (w_i)_{i \in O_c}, 1)$. It sets $C_i = \text{Com}(\text{View}_i; \rho_i)$ for $i \in O_c$ and $C_i = \text{Com}(\mathbf{0}; \rho_i)$, a commitment to an all-zero string of the right length, for $i \notin O_c$. It outputs $((C_i)_{i \in [N]}, c, (\text{View}_i, \rho_i)_{i \in O_c})$.

Consider the hybrid H that is a real transcript on challenge c , except that $C_i = \text{Com}(\mathbf{0}; \rho_i)$ for every $i \notin O_c$. These commitments are never opened, so the real transcript and H are computationally indistinguishable by the hiding of Com . In H , the only information about w lies in the opened views. By t -privacy of the sharing, the opened shares $(w_i)_{i \in O_c}$ are uniform whatever w is. By t -privacy of Π , for every full input $(w_1, \dots, w_N)$ with output 1, the opened views are distributed as $\text{Sim}(O_c, x, (w_i)_{i \in O_c}, 1)$, which depends only on the opened shares. Hence H and the simulation are identically, statistically or computationally indistinguishable, according to the privacy of Π .

For committed channels, the simulator computes the opened channels from the simulated View_i , which determines them, and commits to zero strings for the others; 1-privacy then suffices.

The theorem shows where each ingredient enters. Hiding covers the views that are never opened, the privacy of the sharing covers the opened input shares, and the privacy of Π covers everything else in the opened views. The privacy threshold is dictated by the opening family, not by the relation.

Remark 6 (Malicious verifiers and repetition). Because $\mathcal{C}$ has polynomial size, a single execution is zero-knowledge also against malicious verifiers. The simulator guesses c , prepares the first message as above, runs V^* , and rewinds whenever V^* asks for a different challenge. By hiding, the guess is correct with probability close to $1/|\mathcal{C}|$, so $O(k|\mathcal{C}|)$ attempts suffice except with probability $2^{-\Omega(k)}$. This is the argument of [IKOS09, Thm. 3.2], which follows the GMW analysis for 3-coloring [GMW91]. Zero-knowledge is preserved under the sequential repetition used in [IKOS09] to reduce the soundness error; under parallel repetition only special HVZK is guaranteed, as for Sigma protocols in general, and this is all that the Fiat–Shamir transform requires.

3.3 MPCitH over Public Broadcast Channels

In the broadcast model, every message is received by all parties. The compiler enforces this in the head by publishing a single transcript T_{bc} , so a cheating prover cannot record different versions of one broadcast in different views. Since everything a party receives is contained in T_{bc} , its view is determined by its *state* $\text{st}_i = (w_i, r_i)$ together with T_{bc} . The prover therefore commits only to the states and sends T_{bc} in the clear. We assume, as is standard, that the output is a public function $\text{out}(T_{bc})$ of the transcript, for instance because the parties broadcast shares of the output in the last round.

As in Section 3.2, the compiler is parameterized by an opening family, that is, a challenge space $\mathcal{C}$ with a set $O_c \subseteq [N]$ of opened parties for each $c \in \mathcal{C}$, and $t = \max_c |O_c|$. The families used in this paper are:

- *all-but-one opening*: $\mathcal{C} = [N]$ and $O_{i^*} = [N] \setminus \{i^*\}$, as in [KKW18, BN20];
- *single opening*: $\mathcal{C} = [N]$ and $O_c = \{c\}$;
- more generally, *ℓ -subset opening*: $\mathcal{C}$ is the set of ℓ -element subsets $S \subseteq [N]$, and $O_S = S$.

These families apply to both compilers. The protocol is given in Figure 2. Which opening family is best depends on the witness sharing. With additive sharing, all-but-one opening minimizes the knowledge error (Section 3.4); with a degree-one sharing, as in Schnorr’s protocol (Section 4.2), a single opening already gives knowledge error $1/|\mathcal{C}|$.

Theorem 2 (Zero-knowledge over broadcast). *Let $t = \max_c |O_c|$. Suppose that $(\text{Share}, \text{Rec})$ is t -private, that Π realizes $f_{\mathcal{R}}$ in the broadcast model with perfect correctness and (perfect, statistical or computational) t -privacy, and that Com is computationally hiding. Then $\Pi_{\mathcal{R}}^{bc}$ is perfectly complete and computationally special honest-verifier zero-knowledge.*

Proof. Completeness is as in Theorem 1, with consistency with T_{bc} in place of pairwise consistency. For special HVZK, given $c \in \mathcal{C}$, the simulator samples $(w_i)_{i \in O_c}$ uniformly and runs $\text{Sim}(O_c, x, (w_i)_{i \in O_c}, 1)$, which in the broadcast model returns the views of the parties in O_c together with T_{bc} (Definition 8).

Protocol $\Pi_{\mathcal{R}}^{\text{bc}}$ statement x , witness w with $\mathcal{R}(x, w) = 1$, opening family $(\mathcal{C}, (O_c)_{c \in \mathcal{C}})$.

1. *Commit.* P computes $(w_1, \dots, w_N) \leftarrow \text{Share}(w)$, samples random tapes $r_1, \dots, r_N$, and runs Π in the broadcast model in its head to obtain T_{bc} . It sends T_{bc} and $C_i = \text{Com}(\text{st}_i; \rho_i)$ with $\text{st}_i = (w_i, r_i)$ for every $i \in [N]$.
2. *Challenge.* V sends a uniformly random $c \in \mathcal{C}$.
3. *Response.* P sends $(\text{st}_i, \rho_i)_{i \in O_c}$.
4. *Verification.* V accepts if and only if $\text{out}(T_{\text{bc}}) = 1$ and, for every $i \in O_c$,
 - (i) $C_i = \text{Com}(\text{st}_i; \rho_i)$;
 - (ii) the view of P_i determined by st_i and T_{bc} is well formed and consistent with T_{bc} (Definition 7); that is, recomputing the broadcasts of P_i from st_i and T_{bc} via (1) reproduces the entries of P_i in T_{bc} .

Fig. 2. MPCitH over a broadcast channel. With all-but-one opening this is the structure of [KKW18, BN20].

It reads off $\text{st}_i = (w_i, r_i)$ for $i \in O_c$, commits to these and to all-zero states for the other parties, and outputs T_{bc} , the commitments, c and the openings. The analysis is the one of Theorem 1: hiding covers the unopened commitments, and the privacy of the sharing and of Π cover the rest.

What is specific to broadcast is that T_{bc} also contains the broadcasts of the unopened parties. This is why the privacy simulator must output the whole transcript, and why t -privacy must hold with respect to the views *and* T_{bc} .

Remark 7 (What broadcast buys). Compared with Figure 1, the broadcast compiler improves three things. First, the verifier checks each opened party once against T_{bc} , in time linear in $|T_{\text{bc}}|$, instead of checking all pairs of opened views. Second, the committed state $\text{st}_i = (w_i, r_i)$ is small: all shares but one, and all random tapes, can be derived from short seeds, with one explicit correction share, and with all-but-one opening the $N - 1$ opened seeds cost $O(\lambda \log N)$ bits with a GGM seed tree [KKW18]. This is what makes N in the hundreds practical. Third, and most importantly for this paper, every check becomes a vertex check.

Remark 8 (Committed broadcast). Instead of sending T_{bc} in the clear, the prover may commit to it and open, for each challenge, only the entries that the checks of the opened parties involve. The commitment still fixes a single transcript, so every check remains a vertex check. This is the structure of Blum’s Hamiltonian-cycle protocol (Section 6.2): the committed adjacency matrix plays the role of T_{bc} , and a challenge opens either all of it or only the entries on the cycle.

The transcript as a commitment. The commitments C_i in Figure 2 only bind the prover to the states before the challenge. Sometimes the transcript already does this.

Lemma 2 (Transcript as commitment). Suppose that, for every $i \in [N]$ and every broadcast transcript T_{bc} , at most one well-formed state st_i is consistent with T_{bc} . Let $\Pi_{\mathcal{R}}^{\text{bc}, -}$ be the protocol of Figure 2 without the commitments C_i , for any opening family. Then $\Pi_{\mathcal{R}}^{\text{bc}, -}$ is perfectly complete, has the same special-soundness parameter as $\Pi_{\mathcal{R}}^{\text{bc}}$ (Theorem 3), and is special HVZK with the same quality as the privacy of Π (perfect, statistical or computational), without any hiding assumption.

Proof. The proof of Theorem 3 uses the commitments only to ensure that accepting transcripts with the same first move open the same state for each party. Here the first move contains T_{bc} , which determines every consistent state. For zero-knowledge, a transcript on challenge c now consists of T_{bc} , c and $(\text{st}_i)_{i \in O_c}$. After sampling the opened shares uniformly, this is exactly what $\text{Sim}(O_c, x, (w_i)_{i \in O_c}, 1)$ outputs.

3.4 Soundness, Special Soundness and Proofs of Knowledge

Ishai et al. analyze the soundness of their compilers directly, and remark that the constructions are also proofs of knowledge but leave this property out for simplicity [IKOS09, Sect. 2.1]. We treat both at once. We first recall that special soundness yields the soundness error and the knowledge error together. We then show that every compiler of Sections 3.2 and 3.3 is special sound, with a parameter determined by two things: how an inconsistency among the committed views can be *attributed*, to a single party or only to a pair of parties, and how quickly the opened sets cover all parties.

From special soundness to soundness and knowledge. The following standard fact reduces both properties to special soundness. We include its short proof because it also covers small challenge spaces, which is the case for most compilers below.

Proposition 1 (Special soundness implies soundness and knowledge soundness). *Let Π be a k -special sound, public-coin three-move protocol with challenge space $\mathcal{C}$, and let $\kappa = (k - 1)/|\mathcal{C}|$. Then Π has soundness error κ . If, moreover, $|\mathcal{C}|$ is polynomial in $|x|$, then Π is a proof of knowledge with knowledge error κ in the sense of Definition 2.*

Proof. Without loss of generality, P^* is deterministic once its random tape ρ is fixed, so its first move is a function of ρ . Let $A_\rho \subseteq \mathcal{C}$ be the set of challenges on which P^* with tape ρ makes V accept. If $|A_\rho| \geq k$, the special-soundness extractor applied to k of these transcripts outputs a witness; otherwise $|A_\rho|/|\mathcal{C}| \leq \kappa$. Hence

$$\epsilon = \mathbb{E}_\rho[|A_\rho|/|\mathcal{C}|] \leq \Pr_\rho[|A_\rho| \geq k] + \kappa.$$

Soundness. If $x \notin L_{\mathcal{R}}$, no witness exists, so $|A_\rho| \leq k - 1$ for every ρ and $\epsilon \leq \kappa$.

Knowledge soundness. The extractor samples a fresh ρ , runs P^* to obtain the first move, and rewinds it once for every $c \in \mathcal{C}$. If $|A_\rho| \geq k$, it applies the special-soundness extractor; otherwise it repeats. Each attempt succeeds with probability at least $\epsilon - \kappa$ and costs $|\mathcal{C}| + 1$ invocations of P^* , so the expected running time is at most $\text{poly}(|x|) \cdot |\mathcal{C}|/(\epsilon - \kappa)$.

Remark 9. When $\mathcal{C}$ is exponentially large, as for Schnorr-type protocols, trying every challenge is too expensive. The same knowledge error $(k - 1)/|\mathcal{C}|$ is then achieved by an extractor that samples challenges at random [ACK21]; for $k = 2$ this is the classical heavy-row argument. For the t -fold parallel repetition of a k -special-sound protocol, the knowledge error drops to κ^t [AF22], which supports the parallel-repetition claim of Remark 1 for knowledge soundness and not only for soundness.

It therefore suffices to determine the special-soundness parameter of each compiler.

Setting. Fix a witness sharing (Share, Rec) (Definition 4) and an N -party protocol Π that realizes $f_{\mathcal{R}}$ with perfect correctness (Definition 8). The protocol is compiled with Figure 1, with Figure 2, or with committed channels (Section 3.2), for an arbitrary opening family $(\mathcal{C}, (O_c)_{c \in \mathcal{C}})$. The ℓ -subset opening applies equally to the private-channel compiler, where pair opening and all-but-one opening are its cases $\ell = 2$ and $\ell = N - 1$. Following [IKOS09], we work in the commitment-hybrid model, or equivalently with a perfectly binding Com, so that the first move fixes the N views; a statistically binding Com adds its binding error to the bounds below. Protocols Π that are only statistically correct require the additional coin-flipping step of [IKOS09, Sect. 3.1] and are not covered here.

Faults and checks. For fixed views, party P_i is *faulty* if View_i is not well formed (Definition 7) or $\text{out}(\text{View}_i) \neq 1$. In the broadcast compiler, P_i is also faulty if View_i is not consistent with T_{bc} ; with committed channels, if View_i disagrees with one of its incident committed channels. For the private-channel compiler we use the *inconsistency graph* of [IKOS09, proof of Thm. 4.1]: the graph $\mathcal{G}$ on $[N]$ with an edge $\{i, j\}$ whenever $\text{View}_i \not\sim \text{View}_j$. The broadcast compiler and single opening with committed channels use *vertex checks*, each involving one opened view together with public or committed data. The private-channel compiler uses *edge checks*, the pairwise consistency of two opened views.

By Lemma 1 and perfect correctness, if no party is faulty and $\mathcal{G}$ has no edge, then the committed views form an honest execution in which every party outputs $\mathcal{R}(x, \text{Rec}(w_1, \dots, w_N))$. With committed channels the lemma applies because two views that agree with the same committed channel are consistent with each other.

Definition 9 (Covering numbers). *Let $(\mathcal{C}, (O_c)_{c \in \mathcal{C}})$ be an opening family. Its vertex covering number κ_V is the least k such that, for any k distinct challenges $c_1, \dots, c_k$, the opened sets cover $[N]$, that is, $O_{c_1} \cup \dots \cup O_{c_k} = [N]$. Its edge covering number κ_E is the least k such that, for any k distinct challenges, every pair $\{i, j\} \subseteq [N]$ lies in some O_{c_m} . If no such $k \leq |\mathcal{C}|$ exists, the number is undefined.*

For instance, single opening has $\kappa_V = N$ and no edge covering number, since it never opens two parties together; this is why it can be used only with vertex checks.

Theorem 3 (Special soundness by fault attribution). *A compiler with vertex checks is κ_V -special sound, and a compiler with edge checks is κ_E -special sound. Consequently, by Proposition 1, for the respective k its soundness error is $(k-1)/|\mathcal{C}|$, and it is a proof of knowledge with the same knowledge error. These values of k cannot be lowered in general.*

Proof. Let k accepting transcripts be given, with a common first move and distinct challenges $c_1, \dots, c_k$. The first move fixes the views, so every transcript that opens index j opens the same View_j . With vertex checks and $k = \kappa_V$, every party is opened in some transcript and checked there against public or committed data, so no party is faulty. With edge checks and $k = \kappa_E$, every pair of parties is opened together in some transcript, so $\mathcal{G}$ has no edge; since every party then belongs to an opened pair, no party is faulty either. In both cases the views form an honest execution in which every party outputs $\mathcal{R}(x, w)$ for $w = \text{Rec}(w_1, \dots, w_N)$. Since the opened outputs are 1, the extractor outputs this w .

For the last claim, minimality of k gives $k-1$ distinct challenges that leave some party P_i unopened (vertex checks) or some pair $\{i, j\}$ never opened together (edge checks). A prover whose only fault is at P_i , or on the edge $\{i, j\}$, answers all of them, and its views need not determine a witness.

Corollary 1 (Opening ℓ parties). *For ℓ -subset opening with $1 \leq \ell \leq N-1$,*

$$\kappa_V = \binom{N-1}{\ell} + 1, \quad \frac{\kappa_V - 1}{\binom{N}{\ell}} = 1 - \frac{\ell}{N},$$

and, for $\ell \geq 2$,

$$\kappa_E = \binom{N}{\ell} - \binom{N-2}{\ell-2} + 1, \quad \frac{\kappa_E - 1}{\binom{N}{\ell}} = 1 - \frac{\ell(\ell-1)}{N(N-1)}.$$

In particular:

Compiler	ℓ	checks	k	knowledge error
(a) private, all-but-one	$N-1$	edge	3	$2/N$
(b) broadcast, all-but-one	$N-1$	vertex	2	$1/N$
(c) private, pair opening	2	edge	$\binom{N}{2}$	$1 - 1/\binom{N}{2}$
(d) committed channels, single	1	vertex	N	$1 - 1/N$
(e) broadcast, single	1	vertex	N	$1 - 1/N$

Proof. A collection of ℓ -subsets fails to cover $[N]$ if and only if all of them avoid some party i , and exactly $\binom{N-1}{\ell}$ subsets avoid a given i . Likewise, a collection fails to cover all pairs if and only if none of its members contains some pair $\{i, j\}$, and exactly $\binom{N}{\ell} - \binom{N-2}{\ell-2}$ subsets do not contain a given pair. The error formulas follow from $\binom{N-1}{\ell}/\binom{N}{\ell} = (N-\ell)/N$ and $\binom{N-2}{\ell-2}/\binom{N}{\ell} = \ell(\ell-1)/(N(N-1))$.

Remark 10 (Reading the bounds). For a false statement, the theorem says that a cheating prover survives at most $k-1$ challenges. In (b) this is a single hidden index, that of the one faulty party. In (a) it is at most two hidden indices, which happens when the only inconsistency is one edge and either endpoint may be hidden.

Two consequences follow. First, for a fixed number of parties, opening more parties lowers the knowledge error; with additive sharing, whose privacy threshold is $N-1$, all-but-one opening is therefore optimal. Second, for every ℓ , vertex checks beat edge checks, since $\ell/N \geq \ell(\ell-1)/(N(N-1))$. At $\ell = N-1$ the gap is the factor 2 between $1/N$ and $2/N$: an inconsistency on a private channel belongs to an edge and escapes detection whenever either endpoint is hidden, whereas an inconsistency with a broadcast transcript belongs to its sender alone. This is the all-but-one counterpart of an observation in [IKOS09, Sect. 4]: without robustness, a single misbehaving party can form a vertex cover of $\mathcal{G}$ by itself, and is then caught only if it is opened.

The corollary recovers known bounds. Case (c) is [IKOS09, Thm. 3.1], case (d) is the $1 - 1/N$ of their 1-private variant, and case (b) is the $1/N$ of [KKW18, BN20]. For $N = 3$, cases (a) and (c) coincide and give the soundness error $2/3$ and the 3-special soundness of ZKBoo [GMO16], and, as shown in Section 5.1, of Stern's protocol. Note finally that the extractor uses only binding and perfect correctness; the privacy of Π plays no role in knowledge soundness.

Remark 11 (Reconstruction instead of covering). Theorem 3 extracts by opening every party, which is why it needs Rec to be total and, for single opening, N accepting transcripts. When a public transcript fixes the whole sharing and the sharing has a low reconstruction threshold, fewer transcripts suffice. In Schnorr’s protocol (Section 4.2), single opening of a degree-one sharing is 2-special sound, whereas single opening of an additive sharing, case (e), is only N -special sound. Section 4.1 develops this route for linear relations.

Together with Theorems 1 and 2, this shows that each compiler is a Sigma protocol in the sense of Definition 3. Since soundness is statistical whenever Com is statistically binding, the compiled protocol is a zero-knowledge *proof*; it is an argument only when binding is computational, e.g. for hash-based Merkle commitments.

4 Algebraic Sigma protocols

Algebraic Sigma protocols are instances of the broadcast compiler of Section 3.3, whose algebraic structure removes the explicit commitments and lowers the soundness error.

4.1 MPCitH with Implicit View Commitments

With explicit view commitments, the first move costs at least one commitment per party, so in practice N stays in the hundreds even with seed trees. Algebraic Sigma protocols escape this constraint because their view commitments are *implicit*: a constant number of public group elements determines, through a homomorphism, a commitment to each of exponentially many virtual views.

Let $\mathbb{G}$ be a cyclic group of prime order q and let $\varphi : \mathbb{Z}_q^n \rightarrow \mathbb{G}^m$ be a group homomorphism, equivalently a $\mathbb{Z}_q$ -linear map, with $\mathbb{G}^m$ written multiplicatively. For $d \geq 1$ consider the relation

$$R_{\varphi,d} = \left\{ ((y_1, \dots, y_d), (w_1, \dots, w_d)) : \varphi(w_k) = y_k \text{ for all } k \in [d] \right\}.$$

The case $d = 1$ is the setting of Maurer’s framework [Mau15]; the case $d > 1$ reappears as batching in Section 4.3.

Sharing in the head. To share $(w_1, \dots, w_d)$, the prover samples $r \leftarrow \mathbb{Z}_q^n$ and forms

$$f(X) = r + w_1 X + \dots + w_d X^d \in \mathbb{Z}_q^n[X].$$

Fix an evaluation set $E \subseteq \mathbb{Z}_q$ with $|E| \geq d + 1$. There is one virtual party P_c for each $c \in E$, holding the share $f(c)$. A single share is uniform over $\mathbb{Z}_q^n$ because of the constant term r (privacy threshold 1), and any $d + 1$ shares determine f , hence the witness, by interpolation (reconstruction threshold $d + 1$). The parties have no random tapes and exchange no messages, so $\text{View}_c = f(c)$.

Implicit view commitments. Applying φ coefficient-wise gives the public polynomial

$$F(X) = \varphi(r) \cdot y_1^X \cdots y_d^{X^d}, \quad \text{with} \quad \varphi(f(c)) = F(c) \text{ for every } c \in E.$$

Once the prover broadcasts the single element $a = \varphi(r)$, the verifier can compute $C_c := F(c)$ for every $c \in E$, and C_c serves as the commitment to View_c . The tuple $(a, y_1, \dots, y_d)$ is a Feldman-type commitment to the coefficients of f , and the check $\varphi(f(c)) = F(c)$ is Feldman’s share-verification equation.

Theorem 4. $\Pi_{\varphi,E,d}$ is perfectly complete, $(d + 1)$ -special sound and perfectly special honest-verifier zero-knowledge. Its soundness error is at most $d/|E|$.

Proof. Completeness is the identity $\varphi(f(c)) = F(c)$.

Special soundness is reconstruction. Let (a, c_j, z_j) for $j = 0, \dots, d$ be accepting transcripts with distinct c_j . Let $f'(X) = r' + \sum_k w'_k X^k$ be the unique polynomial of degree at most d with $f'(c_j) = z_j$, obtained by Lagrange interpolation over $\mathbb{Z}_q$. Since φ is $\mathbb{Z}_q$ -linear, applying φ coefficient-wise to f' yields a polynomial of degree at most d whose value at each c_j is $\varphi(z_j) = F(c_j)$. A polynomial of degree at most d over the $\mathbb{Z}_q$ -module $\mathbb{G}^m$ is determined by $d + 1$ values at distinct points, so this

Protocol $\Pi_{\varphi, E, d}$ statement $(y_k)_{k \in [d]}$, witness $(w_k)_{k \in [d]}$.	
1. P samples $r \leftarrow \mathbb{Z}_q^n$ and sends $a = \varphi(r)$.	<i>broadcast; fixes every C_c</i>
2. V samples $c \leftarrow E$ and sends it.	<i>chooses the party to open</i>
3. P sends $z = f(c) = r + \sum_{k=1}^d c^k w_k$.	<i>the view of P_c</i>
4. V accepts iff $\varphi(z) = a \cdot \prod_{k=1}^d y_k^{c^k}$.	<i>consistency with C_c</i>

Fig. 3. MPCitH with implicit view commitments. With $d = n = m = 1$ and $\varphi(w) = g^w$ this is Schnorr's protocol.

polynomial is F , and comparing coefficients gives $\varphi(w'_k) = y_k$ for all k . For $d = 1$ this is the familiar extractor $w' = (z_0 - z_1)/(c_0 - c_1)$.

Zero-knowledge is 1-privacy. Given $c \in E$, the simulator samples $z \leftarrow \mathbb{Z}_q^n$ and sets $a = \varphi(z) \cdot \prod_k y_k^{-c^k}$. In a real execution, $z = f(c)$ is uniform because r is, and a is the unique first message for which (a, c, z) is accepting, so the two distributions are identical.

Soundness. If $(y_k)_k$ has no witness, special soundness implies that for every first message a at most d challenges in E admit an accepting response.

Remark 12 (Implicit commitments need not hide). The simulator never reasons about the unopened views. Once $\text{View}_c = z$ and the public statement are fixed, every other commitment $C_{c'} = \varphi(z) \cdot \prod_k y_k^{c'^k - c^k}$ is a deterministic function of them. In MPC language, the commitments are broadcast messages of a protocol whose output is public, and such messages are simulatable from a corrupted view and the output. This is why Schnorr-type protocols are *perfect HVZK*: the hardness of inverting φ is needed for the security of the derived identification or signature scheme, not for zero-knowledge.

Remark 13 (Where $1/q$ comes from). The soundness argument of Theorem 4 has the same shape as case (b) in Corollary 1: every check involves one opened view and public data, so few challenges can pass. What the algebraic setting changes is the number of parties that can be afforded. With explicit commitments the first move costs $\Omega(N)$; here it costs one evaluation of φ regardless of $|E|$, so one may take $E = \mathbb{Z}_q$.

Remark 14 (Beyond prime order). Reconstruction from shares at $c \neq c'$ requires inverting $c - c'$. When the order of the codomain is unknown it suffices, as in [Mau15], to know an integer ℓ and a preimage u with $\varphi(u) = y^\ell$, and to choose E such that $\gcd(c - c', \ell) = 1$ for all distinct $c, c' \in E$. Writing $\alpha(c - c') + \beta\ell = 1$, the element $w' = \alpha(z - z') + \beta u$ (written additively in the domain) satisfies $\varphi(w') = y$. The largest such E has size equal to the smallest prime factor of ℓ . For Guillou–Quisquater, $\varphi(u) = u^e$, $u = y$ and $\ell = e$. For the quadratic-residuosity protocol of [GMR89], $\ell = 2$ forces $E = \{0, 1\}$, which is why its soundness error is $1/2$.

4.2 Schnorr [Sch91]

Schnorr's protocol with binary challenge. With $d = n = m = 1$, $\varphi(w) = g^w$, and $E = \{0, 1\}$, the protocol in Fig. 3 is Schnorr's protocol [Sch91] with binary challenge.

Let $\mathbb{G} = \langle g \rangle$ be a group of prime order q and $\mathcal{R} = \{(y, w) : y = g^w\}$. Take $N = 2$ parties P_0, P_1 and the witness sharing $\text{Share}(w) = (s_0, s_1) = (r, r + w)$ with $r \leftarrow \mathbb{Z}_q$ and $\text{Rec}(s_0, s_1) = s_1 - s_0$, which is total and 1-private. Let Π_{Sch} be the one-round broadcast protocol in which each P_c broadcasts $b_c = g^{s_c}$, and the output is 1 if and only if $b_1 = b_0 \cdot y$. It is perfectly correct, since $b_1/b_0 = g^{\text{Rec}(s_0, s_1)}$. It is perfectly 1-private, since given s_c and output 1 the transcript is $b_c = g^{s_c}$ and $b_{c'} = b_c \cdot y^{c' - c}$ for the other party c' .

Compile Π_{Sch} with Fig. 2 and single opening; for $N = 2$ this is also all-but-one opening. Each state s_c is determined by b_c , because exponentiation is injective on $\mathbb{Z}_q$, so by Lemma 2 the commitments can be dropped. Moreover, the output condition fixes $b_1 = b_0 \cdot y$, so the prover need only send $a = b_0$, and the verifier computes b_1 itself. The protocol becomes: P sends $a = g^r$, V sends $c \in \{0, 1\}$, P sends $z = s_c = r + cw$, and V checks $g^z = a \cdot y^c$. This is Schnorr's protocol with binary challenges. By Corollary 1(b) and Lemma 2, it is 2-special sound with knowledge error $1/2$ and perfectly special HVZK.

Table 2. Three ways to generalize Schnorr’s protocol with binary challenge. All use vertex checks against a public transcript.

	Path A	single opening, additive	Path B: Schnorr
Parties	$P_1, \dots, P_N$	$P_1, \dots, P_N$	$P_c, c \in \mathbb{Z}_q$
Sharing	$w = \sum_i w_i$	$w = \sum_i w_i$	$f(c) = r + cw$
Reconstruction threshold	N	N	2
Opened parties	$N - 1$	1	1
First message	$b_1, \dots, b_{N-1}$	$b_1, \dots, b_{N-1}$	$a = g^{f(0)}$
Special soundness	2 (covering)	N (covering)	2 (reconstruction)
Knowledge error	$1/N$	$1 - 1/N$	$1/q$
Cost	linear in N	linear in N	independent of q

From two parties to q parties. The shares $s_0 = r$ and $s_1 = r + w$ are the evaluations at 0 and 1 of $f(X) = r + wX$. At $N = 2$, single opening and all-but-one opening coincide, and so do 1-privacy and $(N - 1)$ -privacy. Enlarging the challenge space forces a choice between keeping the additive sharing and keeping the single opening (Table 2).

Additive sharing with N parties. Share $w = w_1 + \dots + w_N$, let each P_i broadcast $b_i = g^{w_i}$, and output 1 if and only if $\prod_i b_i = y$. This protocol is $(N - 1)$ -private, and by Lemma 2 it needs no commitments: the prover sends $b_1, \dots, b_{N-1}$. With all-but-one opening (Path A) the knowledge error is $1/N$. With single opening it is only $1 - 1/N$, because every party must be opened before a witness can be extracted (Corollary 1(e)). In both cases the cost grows linearly with N , because the b_i are independent group elements.

Path B: single opening of a degree-one sharing. Let there be one party P_c for each $c \in \mathbb{Z}_q$, holding $f(c)$. Each P_c broadcasts $b_c = g^{f(c)}$, and the output is 1 if and only if $b_c = b_0 \cdot y^c$ for every c . Any single share is uniform, and given $f(c)$ and output 1 the transcript is $b_{c'} = g^{f(c)} \cdot y^{c'-c}$; so the protocol is 1-private, and one party can be opened. Whenever the output is 1, the whole transcript is determined by b_0 and y . The prover therefore sends only $a = b_0$, and the verifier computes the broadcast $a \cdot y^c$ of the opened party itself. Opening P_c then gives exactly Schnorr’s protocol: $z = f(c) = r + cw$ and the check $g^z = a \cdot y^c$. This protocol has q parties, but the prover never runs it: it evaluates only the opened share, and the transcript is described by two group elements.

For Path B, the covering argument of Section 3.4 is useless: it would open all q parties. Moreover, the sharing is not of the form of Definition 4, since not every q -tuple lies on a line. The analysis is simpler. Every check still involves one opened party and public data. If two challenges $c \neq c'$ are answered for the same a , then $g^{z-z'} = y^{c-c'}$, so $w = (z - z')/(c - c')$: two opened parties reconstruct the witness, because the degree-one sharing has reconstruction threshold 2. Hence the protocol is 2-special sound with knowledge error $1/q$ (Proposition 1 and Remark 9).

The middle column of Table 2 isolates what makes this work. Single opening needs N accepting transcripts with an additive sharing, but only two with a degree-one sharing. The opening family is the same; what changes is the reconstruction threshold, from N to 2. Because the public transcript fixes the whole sharing, the extractor can reconstruct from any two consistent opened parties without ever opening the others. Section 4.1 extends Path B to arbitrary linear maps φ and to higher-degree sharings. Threshold MPCitH [FR23] is its counterpart for non-linear relations, where the views must again be committed explicitly.

Remark 15 (Knowledge error versus soundness). In a group of prime order every y has a discrete logarithm, so the statement proved by Schnorr’s protocol is always true, and its soundness error is vacuous; the meaningful guarantee is the knowledge error $1/q$. For Chaum–Pedersen the distinction disappears: $(g^w, h^{w'})$ with $w \neq w'$ is a false statement, which is what lossy identification [AFLT12] exploits.

4.3 Batch Schnorr [GLSY04]

Gennaro et al. [GLSY04] prove knowledge of d discrete logarithms at essentially the cost of a single Schnorr execution. Let $\mathbb{G} = \langle g \rangle$ have prime order q . The statement is $(y_1, \dots, y_d) \in \mathbb{G}^d$, and the witness is $(w_1, \dots, w_d) \in \mathbb{Z}_q^d$ with $y_k = g^{w_k}$ for every $k \in [d]$. The protocol runs as follows.

1. P samples $r \leftarrow \mathbb{Z}_q$ and sends $a = g^r$.
2. V sends a random $c \in \mathbb{Z}_q$.
3. P sends $z = r + \sum_{k=1}^d c^k w_k$.
4. V accepts if and only if $g^z = a \cdot \prod_{k=1}^d y_k^{c^k}$.

A single challenge serves all d witnesses, through its powers $c, c^2, \dots, c^d$. This is the protocol $\Pi_{\varphi, \mathbb{Z}_q, d}$ of Figure 3 with $\varphi(w) = g^w$, so Theorem 4 applies directly.

In MPCitH terms, batching changes only the degree of the sharing. The d witnesses become the non-constant coefficients of $f(X) = r + w_1 X + \dots + w_d X^d$. Party P_c still holds the single share $f(c)$, and its implicit commitment $C_c = a \cdot \prod_k y_k^{c^k}$ is still computed from public data. The privacy threshold stays 1, because the constant term r still masks every share, so one party is opened as before and the response is still one element of $\mathbb{Z}_q$. The reconstruction threshold, however, rises from 2 to $d+1$, since interpolating a polynomial of degree d takes $d+1$ shares. The protocol is therefore $(d+1)$ -special sound, with knowledge error d/q (Theorem 4 and Remark 9).

The sharing is thus a ramp scheme with privacy threshold 1 and reconstruction threshold $d+1$. The gap d between the two thresholds is exactly the number of witnesses packed into one sharing, and it reappears as the factor d in the knowledge error. Compared with d parallel executions of Schnorr's protocol, which cost d group elements and d responses for knowledge error $1/q$, batching sends one of each and loses only a factor d , which is negligible for $d \ll q$.

4.4 Okamoto [Oka92]

Let $\mathbb{G} = \langle g \rangle$ have prime order q , and let $h \in \mathbb{G}$ be a public system parameter. Okamoto's protocol [Oka92] proves knowledge of a *representation* of $u \in \mathbb{G}$, that is, of a witness $(\alpha, \beta) \in \mathbb{Z}_q^2$ with $g^\alpha h^\beta = u$. Following the presentation of [BS23, Sect. 19.5.1], with challenge space $\mathcal{C} \subseteq \mathbb{Z}_q$, it runs as follows.

1. P samples $\alpha_t, \beta_t \leftarrow \mathbb{Z}_q$ and sends $u_t = g^{\alpha_t} h^{\beta_t}$.
2. V sends a random $c \in \mathcal{C}$.
3. P sends $(\alpha_z, \beta_z) = (\alpha_t + \alpha c, \beta_t + \beta c)$.
4. V accepts if and only if $g^{\alpha_z} h^{\beta_z} = u_t \cdot u^c$.

This is the protocol $\Pi_{\varphi, \mathcal{C}, 1}$ of Figure 3 with $n = 2$, $m = 1$ and $\varphi(\alpha, \beta) = g^\alpha h^\beta$. By Theorem 4, it is perfectly complete, 2-special sound and perfectly special HVZK, with knowledge error $1/|\mathcal{C}|$.

The sharing. The witness is now a vector, and so is every share. The prover shares (α, β) with the degree-one polynomial

$$f(X) = (\alpha_t, \beta_t) + (\alpha, \beta) X \in \mathbb{Z}_q^2[X],$$

and party P_c holds $f(c) = (\alpha_z, \beta_z)$, with implicit commitment $C_c = u_t \cdot u^c = \varphi(f(c))$. Both coordinates are evaluated at the same point c and opened at the same party. The statement forces this: it contains only $\varphi(\alpha, \beta) = u$, not g^α and h^β separately, so a share of α and a share of β can be checked only together, through φ . Otherwise nothing changes from Schnorr's protocol. The uniform mask (α_t, β_t) makes each share uniform over $\mathbb{Z}_q^2$ (privacy threshold 1), and two shares reconstruct a witness,

$$(\alpha', \beta') = \left(\frac{\alpha_z - \alpha'_z}{c - c'}, \frac{\beta_z - \beta'_z}{c - c'} \right)$$

(reconstruction threshold 2).

A non-injective homomorphism. What is new is that φ is not injective. Its kernel $\{(\alpha, \beta) : g^\alpha h^\beta = 1\}$ has q elements, so every statement has exactly q witnesses, and the implicit commitment C_c determines the share $f(c)$ only up to $\ker \varphi$. This has two consequences.

First, special soundness does not need the views to be bound. Lemma 2 does not apply, since many states are consistent with the same broadcast, but Theorem 4 never used it. The first message fixes only F , the image of the sharing under φ , and interpolating any two consistent opened shares yields a preimage of u . The extractor thus recovers a witness, which need not be the one the prover used.

Second, the protocol is perfectly witness indistinguishable. The 1-privacy simulator of Theorem 4 uses only the statement and the output, not the witness, so the transcript has the same distribution for all q witnesses of u . This is what Okamoto's security proof exploits [Oka92]. A reduction that knows one representation of u can run the protocol honestly, and the representation it extracts from a successful impersonator differs from its own with probability $1 - 1/q$. Two distinct representations $(\alpha, \beta) \neq (\alpha', \beta')$ of u have $\beta \neq \beta'$ and yield $\log_g h = (\alpha' - \alpha)/(\beta - \beta')$, so security against active attacks follows from the discrete-logarithm assumption alone.

The same reading applies verbatim to representations with respect to n generators, with

$$\varphi(w_1, \dots, w_n) = \prod_i g_i^{w_i}$$

and shares in $\mathbb{Z}_q^n$, for generators $g_1, g_2, \dots, g_n$ in $\mathbb{G}$.

4.5 Chaum-Pedersen [CP92] and Lossy ID

Let $\mathbb{G} = \langle g \rangle$ have prime order q , and let $h \in \mathbb{G} \setminus \{1\}$. A triple (h, y_1, y_2) is a Diffie-Hellman (DDH) triple if $y_1 = g^w$ and $y_2 = h^w$ for some $w \in \mathbb{Z}_q$, that is, if $\log_g y_1 = \log_h y_2$. The Chaum-Pedersen protocol [CP92] proves this for the statement $y = (y_1, y_2)$ and the language

$$L_{\text{DH}} = \{(g^w, h^w) : w \in \mathbb{Z}_q\} \subseteq \mathbb{G}^2.$$

With challenge space $\mathcal{C} \subseteq \mathbb{Z}_q$, it runs as follows.

1. P samples $r \leftarrow \mathbb{Z}_q$ and sends $(a_1, a_2) = (g^r, h^r)$.
2. V sends a random $c \in \mathcal{C}$.
3. P sends $z = r + cw$.
4. V accepts if and only if $g^z = a_1 \cdot y_1^c$ and $h^z = a_2 \cdot y_2^c$.

This is the protocol $\Pi_{\varphi, \mathcal{C}, 1}$ of Figure 3 with $n = 1$, $m = 2$ and $\varphi(w) = (g^w, h^w)$. By Theorem 4, it is perfectly complete, 2-special sound and perfectly special HVZK, with knowledge error $1/|\mathcal{C}|$.

The sharing. The sharing is Schnorr's: party P_c holds the scalar share $f(c) = r + cw$ of $f(X) = r + wX$. What changes is the commitment, which now has two components, $C_c = (a_1 y_1^c, a_2 y_2^c) = \varphi(f(c))$, so the two verification equations are two checks on the same single view. The protocol is thus dual to Okamoto's (Section 4.4), which has a two-coordinate share and a one-component commitment. The duality extends to φ . Okamoto's φ is surjective with a kernel of size q , so every statement is true and has q witnesses. Here φ is injective but not surjective: its image L_{DH} has q elements in a group of q^2 , so almost every statement is false, and soundness for the language becomes meaningful (compare Remark 15).

Soundness for the DDH language. For a false statement, the implicit commitments leave room for at most one virtual party.

Proposition 2. *Let $y = (y_1, y_2) \notin L_{\text{DH}}$. For every first message $a = (a_1, a_2) \in \mathbb{G}^2$, there is at most one $c \in \mathbb{Z}_q$ for which party P_c can be opened, that is, for which some z passes the check. Hence the soundness error is at most $1/|\mathcal{C}|$, even against unbounded provers, and this bound is tight.*

Proof. Write $\mathbb{G}^2$ additively, as a two-dimensional vector space over $\mathbb{Z}_q$. Then $L_{\text{DH}} = \varphi(\mathbb{Z}_q)$ is a line through the origin, and the commitments $C_c = a + cy$ form an affine line with direction y . Party P_c can be opened if and only if $C_c \in L_{\text{DH}}$, and then $z = \varphi^{-1}(C_c)$ is its unique passing view. If C_c and $C_{c'}$ both lie in L_{DH} for some $c \neq c'$, then $(c - c')y = C_c - C_{c'} \in L_{\text{DH}}$, so $y \in L_{\text{DH}}$, a contradiction. For tightness, a prover who guesses c^* picks z and sends $a = \varphi(z) - c^*y$, which makes P_{c^*} pass.

The bound also follows from 2-special soundness and Proposition 1. The direct argument, however, shows what happens in the head: when the statement is false, the commitment line is not parallel to the image of φ and meets it in at most one point, so all but one of the virtual views are unopenable, whatever the first message. This is the single-opening counterpart of the all-but-one bound for broadcast, where a false statement forces at least one faulty party.

Lossy identification. Katz and Wang [KW03] observed that a tight security reduction for Fiat–Shamir signatures can rely on the underlying protocol being a proof of *membership* rather than a proof of knowledge, and used the Chaum–Pedersen protocol to build a signature scheme tightly related to DDH. Abdalla et al. [AFLT12] formalized this as lossy identification. A lossy identification scheme has, besides the normal key generation, a lossy key generation whose public keys are computationally indistinguishable from normal ones, and for which even an unbounded impersonator succeeds only with small probability. For Chaum–Pedersen, a normal key is a DDH triple (h, g^w, h^w) with secret w . A lossy key is a uniformly random triple, which lies outside L_{DH} except with probability $1/q$. The two are indistinguishable under DDH, and by Proposition 2 a lossy key limits any impersonator to probability $1/|\mathcal{C}|$, information-theoretically.

In MPCitH terms, switching to a lossy key moves the direction of the commitment line off the image of φ . After the switch, at most one virtual party can ever be opened, whatever the first message. Because this bound is statistical and needs no rewinding, the security reduction for the Fiat–Shamir signature avoids the loss of the forking lemma, which is the source of the tightness in [AFLT12, KW03].

4.6 Guillou–Quisquater [GQ88]

Let (n, e) be an RSA public key with e prime, viewed as a system parameter. The Guillou–Quisquater (GQ) protocol [GQ88] proves knowledge of an e -th root of a statement $y \in \mathbb{Z}_n^*$, that is, of a witness $x \in \mathbb{Z}_n^*$ with $x^e = y$. Following the presentation of [BS23, Sect. 19.5.5], with challenge space $\mathcal{C} \subseteq \{0, \dots, e-1\}$, it runs as follows.

1. P samples $x_t \leftarrow \mathbb{Z}_n^*$ and sends $y_t = x_t^e$.
2. V sends a random $c \in \mathcal{C}$.
3. P sends $x_z = x_t \cdot x^c$.
4. V accepts if and only if $x_z^e = y_t \cdot y^c$.

Formally, this is the protocol of Figure 3 for the homomorphism $\psi(x) = x^e$ on $\mathbb{Z}_n^*$. Theorem 4 does not apply as stated, however: $\mathbb{Z}_n^*$ is not a $\mathbb{Z}_q$ -module, and its order is unknown. Special soundness needs a separate argument, and the MPCitH reading shows exactly what changes.

Multiplicative sharing. The witness now lives in the multiplicative group $\mathbb{Z}_n^*$, on which the integers act by exponentiation. The degree-one sharing $r + wX$ of Section 4.1 therefore becomes multiplicative,

$$f(X) = x_t \cdot x^X,$$

and party P_c holds $f(c) = x_t x^c = x_z$. Its implicit commitment $C_c = \psi(f(c)) = y_t \cdot y^c$ is computed from the single element y_t and the statement. The mask x_t is uniform in $\mathbb{Z}_n^*$, and multiplication by x^c is a bijection, so each share is uniform (privacy threshold 1).

Reconstruction without division. Two shares give $f(c)/f(c') = x^{c-c'}$, a power of the witness. In a group of prime order one would now invert $c - c'$ in the exponent; in $\mathbb{Z}_n^*$ this amounts to taking a $(c - c')$ -th root, which is infeasible without knowing the order of the group. Reconstruction is rescued by the fact that ψ maps $\mathbb{Z}_n^*$ to itself, so the statement $y = x^e$ is also a power of the witness. It acts as a public third share. Since e is prime and $0 < |c - c'| < e$, the exponents $c - c'$ and e are coprime, and Bezout’s identity combines them. This is the case $u = y, \ell = e$ of Remark 14.

Proposition 3. *The GQ protocol is perfectly complete, 2-special sound and perfectly special HVZK. Its knowledge error is $1/|\mathcal{C}| \geq 1/e$.*

Proof. Completeness is the identity $(x_t x^c)^e = y_t y^c$. For special soundness, let (y_t, c, x_z) and (y_t, c', x'_z) be accepting with $c \neq c'$, and let $\Delta = c - c'$. Then $(x_z/x'_z)^e = y^\Delta$. Since $\gcd(\Delta, e) = 1$, the extended Euclidean algorithm gives integers α, β with $\alpha\Delta + \beta e = 1$, and

$$\hat{x} = (x_z/x'_z)^\alpha \cdot y^\beta \quad \text{satisfies} \quad \hat{x}^e = y^{\alpha\Delta} \cdot y^{\beta e} = y.$$

For special HVZK, given c , the simulator samples $x_z \leftarrow \mathbb{Z}_n^*$ and sets $y_t = x_z^e \cdot y^{-c}$. As for Schnorr, this is the 1-privacy simulator, and its output is distributed exactly as a real transcript. The knowledge error follows from Proposition 1 and Remark 9.

Why there are only e parties. The restriction $|\mathcal{C}| \leq e$ has a direct MPCitH explanation. Since $f(c+e) = f(c) \cdot x^e = f(c) \cdot y$, the share of party P_{c+e} is a public function of the share of P_c . Parties whose indices agree modulo e are therefore the same party, and the virtual parties are naturally indexed by $\mathbb{Z}_e$. A larger challenge space adds no new parties: a prover who can answer c with x_z can also answer $c+e$ with $x_z \cdot y$, so enlarging $\mathcal{C}$ beyond e does not lower the knowledge error. This is why e is taken to be a large prime when a large challenge space is wanted. For composite e , the evaluation points would moreover have to differ by units modulo e , which caps the number of parties at the smallest prime factor of e (Remark 14).

Knowledge versus membership, and lossy keys. Because $\gcd(e, \varphi(n)) = 1$, the map ψ is a bijection, so every statement has exactly one witness and is true. As for Schnorr (Remark 15), soundness is vacuous, and only the knowledge error is meaningful. The picture changes under a lossy key. If e divides $\varphi(n)$, say $e \mid p-1$ and $e \nmid q-1$ for $n = pq$, then ψ is e -to-one, and its image, the subgroup of e -th residues, has index e in $\mathbb{Z}_n^*$ with a cyclic quotient of order e . For a statement y that is not an e -th residue, the argument of Proposition 2 carries over to this quotient. The commitments $C_c = y_t y^c$ form a line whose direction is nonzero there, so at most one c modulo e makes C_c an e -th residue, and at most one party can be opened. Under the φ -hiding assumption, the two kinds of key are indistinguishable. This makes GQ a lossy identification scheme in the sense of [AFLT12], as shown in [ABP13], which yields a tight security reduction for GQ signatures under φ -hiding.

4.7 GMR for Quadratic Residuosity [GMR89]

Let $n = pq$ with p, q distinct odd primes, and let $\text{QR}_n \subseteq \mathbb{Z}_n^*$ be the subgroup of quadratic residues. The protocol of Goldwasser, Micali and Rackoff [GMR89] proves that a statement $y \in \mathbb{Z}_n^*$ lies in QR_n , with a square root x , $x^2 = y$, as witness. In the notation of Section 4.6, it runs as follows.

1. P samples $x_t \leftarrow \mathbb{Z}_n^*$ and sends $y_t = x_t^2$.
2. V sends a random $c \in \{0, 1\}$.
3. P sends $x_z = x_t \cdot x^c$.
4. V accepts if and only if $x_z^2 = y_t \cdot y^c$.

This is the GQ protocol with $e = 2$, but with an important difference: 2 always divides $\varphi(n)$. The map $\psi(u) = u^2$ is therefore neither injective nor surjective. It is four-to-one onto QR_n , a subgroup of index 4 in $\mathbb{Z}_n^*$.

A 2-out-of-2 multiplicative sharing. The two parties hold $s_0 = f(0) = x_t$ and $s_1 = f(1) = x_t x$. This is a 2-out-of-2 multiplicative sharing of the witness: $\text{Rec}(s_0, s_1) = s_1/s_0 = x$ is defined on every pair, and each share alone is uniform. As for Schnorr's protocol with binary challenge in Section 4.2, the virtual protocol has one broadcast round: each P_c broadcasts $b_c = s_c^2$, and the output is 1 if and only if $b_1 = b_0 \cdot y$. The prover sends only $y_t = b_0$, and the verifier computes $b_1 = y_t \cdot y$ itself.

There is one difference from Schnorr. Squaring is four-to-one, so a broadcast does not determine the state, and Lemma 2 does not apply. As for Okamoto's protocol (Section 4.4), it is not needed, because reconstruction uses only the two opened shares. If x_z and x'_z answer $c = 0$ and $c = 1$ for the same y_t , then $(x'_z/x_z)^2 = y$, so x'_z/x_z is a square root of y , though not necessarily the prover's. Since the two evaluation points differ by 1, the Bezout step of Section 4.6 disappears, and reconstruction is just the division Rec .

Proposition 4. *The protocol is perfectly complete, 2-special sound and perfectly special HVZK. Moreover, if $y \notin \text{QR}_n$, then for every first message y_t at most one party can be opened, so the soundness error is $1/2$ even against unbounded provers.*

Proof. We have shown completeness and special soundness above. Given a challenge c , the special HVZK simulator samples $x_z \leftarrow \mathbb{Z}_n^*$ and sets $y_t = x_z^2 \cdot y^{-c}$. For soundness, party P_c can be opened only if its implicit commitment, $C_c = y_t \cdot y^c$, is a square. If both $C_0 = y_t$ and $C_1 = y_t \cdot y$ are squares, then so is their quotient y , a contradiction.

Why there are only two parties. As for GQ, $f(c+2) = f(c) \cdot y$, so the parties are indexed by $\mathbb{Z}_2$. Moreover, the smallest prime factor of the exponent is 2, so in general the challenge space cannot be enlarged by using a larger exponent 2^k either (Remark 14). The soundness error is reduced by repetition instead. Since $|\mathcal{C}| = 2$, a single execution is zero-knowledge even against malicious verifiers, by the guess-and-rewind simulator of Remark 6, which is the original simulator of [GMR89]. Here no hiding assumption is needed: the simulated first message $y_t = x_z^2 \cdot y^{-c}$ is a uniformly random square whatever c is, so each guess succeeds with probability exactly $1/2$. Sequential repetition preserves zero-knowledge.

Membership, lossiness and factoring. The QR protocol brings together the phenomena of the three previous sections, each with its usual consequence.

- *Unknown order*, as for GQ: the integers act on $\mathbb{Z}_n^*$ by exponentiation, and the evaluation points must differ by units modulo the exponent, which here leaves only two parties.
- *A non-trivial cokernel*, as for Chaum–Pedersen: ψ is never surjective, so soundness for the language is meaningful without any change of key. The quotient $\mathbb{Z}_n^*/\text{QR}_n$ plays the role that the quotient by the e -th residues played for a lossy GQ key, and the proof of Proposition 4 is that of Proposition 2. Under the quadratic residuosity assumption, squares are indistinguishable from non-squares with Jacobi symbol 1. The protocol is therefore a lossy identification scheme by nature, with squares as normal keys and such non-squares as lossy keys [KW03, ABP13].
- *A non-trivial kernel*, as for Okamoto: every square has four square roots, and the protocol is perfectly witness indistinguishable. As in Okamoto’s argument, a reduction that picks x and publishes $y = x^2$ extracts from a successful impersonator a root x' that, by witness indistinguishability, equals $\pm x$ only with probability $1/2$. Otherwise $\gcd(x - x', n)$ is a non-trivial factor of n . This is the factoring-based security argument of Feige-Fiat-Shamir identification [FFS88].

4.8 DL in Unknown Order Groups [PS98, GPS06]

The GQ and QR protocols prove knowledge of roots. For discrete logarithms in a group of unknown order, the classical protocol is that of Girault, Poupard and Stern (GPS) [PS98, GPS06], which is Schnorr’s protocol carried out over the integers. Let $\mathbb{G} = \langle g \rangle$ be a group of unknown order, for instance a subgroup of $\mathbb{Z}_n^*$. The witness is an integer $x \in [0, S)$ and the statement is $y = g^x$. With a challenge limit B and a mask limit A much larger than $(B-1)(S-1)$, typically $2^\lambda(B-1)(S-1)$ for a statistical parameter λ , the protocol runs as follows.

1. P samples $r \leftarrow [0, A)$ and sends $a = g^r$.
2. V sends a random $c \in [0, B)$.
3. P sends $z = r + cx$, computed over the integers.
4. V accepts if and only if $0 \leq z < A + (B-1)(S-1)$ and $g^z = a \cdot y^c$.

Since the order of $\mathbb{G}$ is unknown, the response cannot be reduced modulo it. The original scheme publishes g^{-x} instead of g^x , which only changes a sign.

Sharing over the integers. This is the template of Figure 3 for $\varphi(u) = g^u$, with $\mathbb{Z}$ in place of $\mathbb{Z}_q$. The sharing is $f(X) = r + xX \in \mathbb{Z}[X]$, party P_c holds $f(c) = z$, and its implicit commitment is again $C_c = a \cdot y^c$. Two things change: the shares are no longer uniform, and they can no longer be divided.

Statistical privacy. Since r is uniform on $[0, A)$, the share $f(c)$ is uniform on $[cx, cx + A)$. Its statistical distance from the uniform distribution on $[0, A)$ is $cx/A \leq (B-1)(S-1)/A$. Privacy threshold 1 thus holds only statistically: the large mask smudges the witness. The simulator samples $z \leftarrow [0, A)$ and sets $a = g^z \cdot y^{-c}$, and taking $A \geq 2^\lambda(B-1)(S-1)$ makes the distance negligible. The verifier’s range check is the price of this. Shares are integers rather than group elements, and only bounded shares can be accepted.

Reconstruction over the integers. Two shares satisfy $g^{z-z'} = y^{c-c'}$, and reconstruction needs the exact quotient $(z-z')/(c-c')$, which exists in general only if $c-c' = \pm 1$. Unlike GQ, no public third share is available. In GQ, the statement $y = x^e$ lies in the same group as the witness and supplies the exponent e for Bezout’s identity. Here the witness lies in $\mathbb{Z}$ and the statement in $\mathbb{G}$. Remark 14 would need an integer ℓ and a known u with $g^u = y^\ell$; since $u \equiv \ell x$ modulo the order of g , finding such a pair without x amounts to knowing a multiple of the order. In MPCitH terms, the implicit

Table 3. Three protocols over groups of unknown order.

	GQ	QR	GPS
Witness	$x \in \mathbb{Z}_n^*$	$x \in \mathbb{Z}_n^*$	$x \in [0, S)$
Statement	$y = x^e$	$y = x^2$	$y = g^x$
Statement is a share	yes	yes	no
Reconstruction	Bezout with e division	division	integer division
Parties	e	2	2
Privacy	perfect	perfect	statistical
Knowledge error	$1/e$	$1/2$	$1/2$

commitments $a \cdot y^c$ exist for every integer c , but two opened parties reconstruct the witness only if their indices differ by a unit of $\mathbb{Z}$. The evaluation set is therefore limited by reconstruction, not by the commitments, and generically $E = \{0, 1\}$.

Proposition 5. *With binary challenges ($B = 2$), the GPS protocol is perfectly complete and statistically special HVZK, with statistical distance at most $(S - 1)/A$. It is 2-special sound for the relaxed relation $\{(y, x') : x' \in \mathbb{Z}, g^{x'} = y\}$: from accepting transcripts $(a, 0, z_0)$ and $(a, 1, z_1)$, the integer $x' = z_1 - z_0$ satisfies $g^{x'} = y$ and $|x'| < A + S$.*

Proof. Completeness holds because $0 \leq r + cx < A + (B - 1)(S - 1)$. Zero-knowledge was shown above. For special soundness, $g^{z_0} = a$ and $g^{z_1} = a \cdot y$ give $g^{z_1 - z_0} = y$, and both responses lie in $[0, A + S - 1)$.

The extracted witness is relaxed in two ways: it may be negative, and it may exceed S . Both are inherent to integer shares. The range check bounds the size of the shares, but not the size of their differences.

Large challenges. For $B > 2$, two transcripts do not in general yield a witness, since $c - c'$ need not divide $z - z'$. GPS is nevertheless used with large challenges, but its security as an identification and signature scheme is proved through a reduction to the discrete logarithm problem with short exponents, not through special soundness [PS98, GPS06]. Bangerter, Camenisch and Krenn show that this is not an artifact of the proof. In the generic group model, they prove lower bounds on the knowledge error of Σ -protocols for exponentiation homomorphisms in groups of hidden order [BCK10], so that repeating binary-challenge protocols is essentially unavoidable. This matches the MPCitH picture: the units of $\mathbb{Z}$ are ± 1 , and a set of integers whose pairwise differences are all units has at most two elements, so at most two parties can be used.

Summary of Section 4. Table 3 compares the three protocols of this kind. More broadly, every protocol in this section is an instance of a single template: one party is opened from a linear sharing with implicit commitments, every check is a vertex check, and special soundness is the reconstruction threshold of the sharing. The protocols differ along four independent directions:

- the degree of the sharing, which sets the reconstruction threshold (batch Schnorr, Section 4.3);
- a kernel of φ , which gives many witnesses and witness indistinguishability (Okamoto, Section 4.4);
- a cokernel of φ , which makes soundness for the language meaningful and gives lossiness (Chaum-Pedersen, Section 4.5);
- the ring of evaluation points, which sets the number of parties: q for $\mathbb{Z}_q$, e when the statement supplies the exponent e (GQ, with QR as the case $e = 2$), and 2 for the bare integers (GPS).

In each case, the size of the challenge space is the size of the largest set of evaluation points from which any two opened parties can reconstruct the witness.

5 Combinatorial Sigma Protocols

5.1 Stern's Protocol [Ste93]

Let $A \in \mathbb{F}_2^{n \times m}$, $y \in \mathbb{F}_2^n$ and a weight ω be public. Stern's protocol [Ste93] proves knowledge of a solution to the syndrome decoding problem, that is, a witness for

$$\mathcal{R}_{\text{SD}} = \{(A, y, \omega), x) : Ax = y \text{ and } \text{wt}(x) = \omega\}.$$

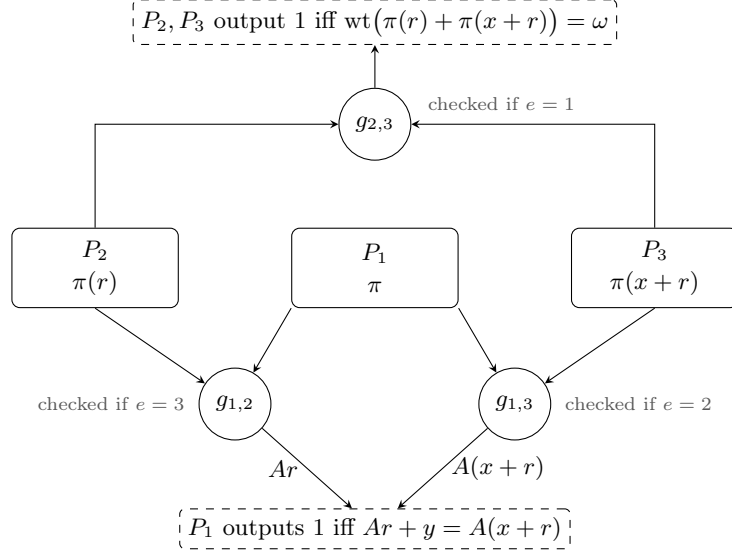


Fig. 4. The virtual three-party protocol Π_{Stern} . Each party holds one share of $\llbracket x \rrbracket = (\pi, \pi(r), \pi(x+r))$, and each pair calls a two-party functionality $g_{i,j}$. Hiding party P_e leaves exactly one call with both endpoints opened, and only that call can be checked.

We show that it is the private-channel compiler of Figure 1 with all-but-one opening and $N = 3$, applied to a three-party protocol in which each pair of parties calls a two-party functionality.

The sharing. The prover samples a permutation π of $[m]$ and a vector $r \leftarrow \mathbb{F}_2^m$, and shares x as

$$\llbracket x \rrbracket = (s_1, s_2, s_3) = (\pi, \pi(r), \pi(x+r)), \quad \text{Rec}(s_1, s_2, s_3) = s_1^{-1}(s_2 + s_3).$$

Reconstruction is defined on every triple and needs all three shares, so this is a 3-out-of-3 sharing. Any two shares hide x . The pairs $(\pi, \pi(r))$ and $(\pi, \pi(x+r))$ are uniform, and the pair $(\pi(r), \pi(x+r))$ reveals only $\pi(x) = \pi(r) + \pi(x+r)$, a uniformly random vector of weight ω , whose distribution is the same for every witness.

The virtual protocol. The three parties never exchange messages directly. Instead, each pair calls an ideal two-party functionality, as allowed in Section 3.1:

- $g_{1,2}(\pi, s_2) = (A\pi^{-1}(s_2), \perp)$ gives P_1 the value Ar ;
- $g_{1,3}(\pi, s_3) = (A\pi^{-1}(s_3), \perp)$ gives P_1 the value $A(x+r)$;
- $g_{2,3}(s_2, s_3) = (b, b)$ with $b = [\text{wt}(s_2 + s_3) = \omega]$ gives P_2 and P_3 the result of the weight test.

Party P_1 outputs 1 if and only if $Ar + y = A(x+r)$, that is, if and only if $Ax = y$; parties P_2 and P_3 output b . Figure 4 shows the resulting protocol Π_{Stern} . Each output checks only one part of the relation, but their conjunction is exactly $\mathcal{R}_{\text{SD}}((A, y, \omega), \text{Rec}(s_1, s_2, s_3))$. The protocol is perfectly 2-private: given the shares of two parties and output 1, their views are determined. For instance, the shares π and $\pi(r)$ determine Ar , and output 1 forces $A(x+r) = Ar + y$ and $b = 1$.

Compiling. Compiling Π_{Stern} with Figure 1 and all-but-one opening gives Figure 5. The commitments are to the three views, compressed by the output conditions. Since an accepted view must output 1, the bits b recorded by P_2 and P_3 need not be committed, and neither does the value $A(x+r)$ in the view of P_1 , which equals $Ar + y$. The commitment of P_1 therefore contains only (π, Ar) . A challenge e hides P_e , and the verifier checks the single functionality call whose two endpoints are opened.

Proposition 6. *The protocol of Figure 5 is perfectly complete, 3-special sound with soundness and knowledge error $2/3$, and computationally special HVZK.*

Proof. Completeness and zero-knowledge follow from Theorem 1 with $t = 2$, since Π_{Stern} is perfectly correct in the conjunctive sense above and perfectly 2-private, and the sharing is 2-private in the

Protocol $\Pi_{\text{Stern}}^{\text{priv}}$ statement (A, y, ω) , witness x with $Ax = y$ and $\text{wt}(x) = \omega$.

1. *Commit.* P samples a permutation π and $r \leftarrow \mathbb{F}_2^m$, and sends

$$C_1 = \text{Com}(\pi, Ar), \quad C_2 = \text{Com}(\pi(r)), \quad C_3 = \text{Com}(\pi(x+r)).$$
2. *Challenge.* V sends a random $e \in \{1, 2, 3\}$, the index of the hidden party.
3. *Response.* P opens the two commitments C_j with $j \neq e$.
4. *Verification.* V checks the openings and then the call between the two opened parties:
 - $e = 1$ (call $g_{2,3}$): $\text{wt}(\pi(r) + \pi(x+r)) = \omega$;
 - $e = 2$ (call $g_{1,3}$ and the output of P_1): $A\pi^{-1}(\pi(x+r)) = Ar + y$;
 - $e = 3$ (call $g_{1,2}$): $A\pi^{-1}(\pi(r)) = Ar$.

Fig. 5. Stern’s protocol [Ste93] as MPC-in-the-head over private channels with $N = 3$ and all-but-one opening.

sense just described. All checks are edge checks, so by Theorem 3 and case (a) of Corollary 1 the protocol is κ_E -special sound with $\kappa_E = 3$ and error $2/3$. Concretely, three accepting transcripts for the same commitments open all three views and check all three calls, so $x' = \pi^{-1}(\pi(r) + \pi(x+r))$ satisfies $Ax' = Ar + (Ar + y) = y$ and $\text{wt}(x') = \omega$.

With a statistically binding commitment the protocol is a proof; Stern’s original instantiation commits with a hash function, which gives an argument under collision resistance.

Why the error is $2/3$. Every check in Π_{Stern} is a two-party functionality call, which the verifier can check only if it opens both endpoints. An inconsistency in a single call therefore escapes detection whenever either endpoint is hidden, that is, for two of the three challenges. This is the edge-check bound of Section 3.4; vertex checks would give $1/3$. The structure also explains why Stern’s protocol does not improve by simply adding parties: with private channels, all-but-one opening still gives only $2/N$, and the weight test, a non-linear check, has to be distributed among the parties somehow. Lowering the error to $1/N$ requires vertex checks over a broadcast channel (Section 3.3), which is the route taken by later MPCitH protocols for syndrome decoding (Section 6).

Remark 16 (Two relaxations of Section 3.1). Stern’s protocol uses Section 3.1 with two harmless relaxations. First, the opened shares $(\pi(r), \pi(x+r))$ are not uniform, as Definition 4 asks; they reveal a random vector of weight ω . Their distribution is nevertheless the same for every witness and easy to sample, which is all that the simulator of Theorem 1 needs. Second, no single party computes $f_{\mathcal{R}}$: the outputs are partial checks whose conjunction is $f_{\mathcal{R}}$. This suffices for Theorem 3, because its extractor opens every party.

5.2 Graph 3-Coloring [GMW91, GMW86]

Let $G = (V, E)$ be a graph. A witness is a proper 3-coloring $\chi : V \rightarrow \{1, 2, 3\}$, that is, $\chi(u) \neq \chi(v)$ for every edge $\{u, v\} \in E$. The protocol of Goldreich, Micali and Wigderson [GMW91] runs as follows.

1. P samples a random permutation σ of $\{1, 2, 3\}$ and, for every $v \in V$, sends $C_v = \text{Com}(\sigma(\chi(v)))$.
2. V sends a random edge $\{u, v\} \in E$.
3. P opens C_u and C_v .
4. V accepts if and only if both openings are valid and the two opened colors are distinct elements of $\{1, 2, 3\}$.

We show that this is the private-channel compiler of Figure 1, with pair opening restricted to the edges of G , applied to a protocol whose network is the graph G itself.

The sharing. There is one virtual party P_v for each vertex, so $N = |V|$, and P_v holds the share $s_v = \sigma(\chi(v))$. Reconstruction is the identity, $\text{Rec}((s_v)_{v \in V}) = (s_v)_{v \in V}$: the shares themselves form a witness candidate, because relabeling the colors preserves properness, so $\sigma \circ \chi$ is again a proper coloring. The mask is therefore not additive. It is σ , a uniformly random symmetry of the relation, as in Stern’s protocol, where the permutation π preserves the weight. An opened set reveals exactly the relabeling-invariant information about χ on that set. For an edge $\{u, v\}$, this is only that the two

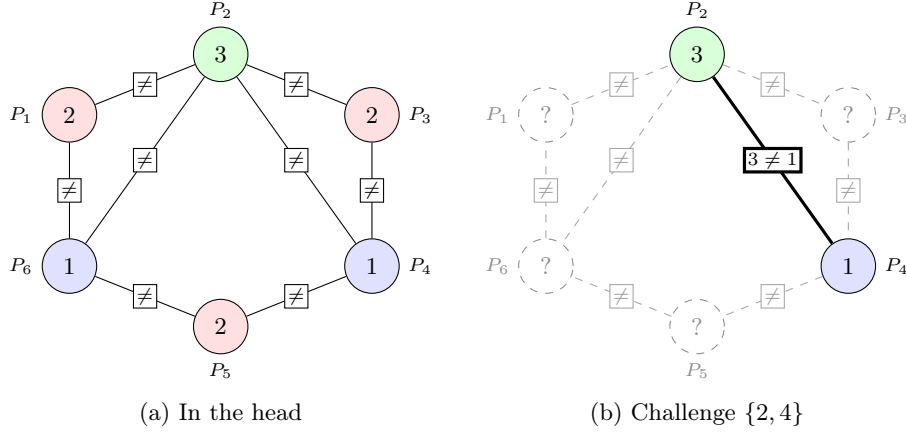


Fig. 6. The virtual protocol for 3-coloring on a small graph, with coloring $\chi = (1, 2, 1, 3, 1, 3)$ and relabeling $\sigma = (1\ 2\ 3)$. (a) Each vertex is a party holding its share $s_v = \sigma(\chi(v))$, and each edge is a private channel on which the two endpoints call $g_{u,v} = [s_u \neq s_v]$. (b) A challenge opens the two endpoints of one edge. Only their call can be checked, and the two opened shares are a uniformly random pair of distinct colors.

colors differ: the pair (s_u, s_v) is uniform over the six ordered pairs of distinct colors, the same for every proper coloring. For two non-adjacent vertices, the shares reveal whether $\chi(u) = \chi(v)$. The sharing is thus private on every edge, but not 2-private in the sense of Definition 4; Remark 17 explains why this suffices.

The virtual protocol. The network of the virtual protocol is G : parties P_u and P_v share a channel if and only if $\{u, v\} \in E$. As in Stern’s protocol, they do not exchange messages but call a two-party functionality,

$$g_{u,v}(s_u, s_v) = (b, b), \quad b = [s_u \neq s_v].$$

Each P_v outputs 1 if and only if $s_v \in \{1, 2, 3\}$ and all its bits are 1. No single output decides the relation, but their conjunction states that $\text{Rec}((s_v)_v)$ is a proper coloring. Exchanging the colors directly would break privacy: the view of P_u would then contain the colors of all its neighbors, and opening one edge would reveal which neighbors share a color. With the functionalities, opening an edge $\{u, v\}$ reveals two views that are determined by s_u, s_v and the output 1, since every recorded bit equals 1. The protocol therefore satisfies perfect privacy (Definition 8) for every set T that is an edge. Figure 6 illustrates it on a small graph.

Compiling. Compile the protocol with Figure 1 and the opening family $\mathcal{C} = E$, $O_{\{u,v\}} = \{u, v\}$. An accepted view outputs 1, so all its bits equal 1 and need not be committed. The commitment of P_v thus contains only s_v , which is exactly the commitment C_v of the GMW protocol. A challenge $\{u, v\}$ opens both endpoints and checks the single call $g_{u,v}$, that is, $s_u \neq s_v$; well-formedness gives $s_u, s_v \in \{1, 2, 3\}$.

Proposition 7. *The GMW protocol is perfectly complete, $|E|$ -special sound with soundness and knowledge error $1 - 1/|E|$, and computationally special HVZK. A single execution is zero-knowledge also against malicious verifiers.*

Proof. Zero-knowledge follows from the proof of Theorem 1, with privacy required only on edges (Remark 17). For a challenge $\{u, v\}$, the simulator samples two random distinct colors and commits to them and to dummy values elsewhere, which is the simulator of [GMW91]. Since $|E|$ is polynomial, the guess-and-rewind argument of Remark 6 handles malicious verifiers. For soundness, every check is an edge check on a channel of G , and a set of challenges covers every channel only if it contains every edge. Counting only pairs that share a channel (Remark 17), the edge covering number is therefore $|E|$, and Theorem 3 gives $|E|$ -special soundness with error $(|E| - 1)/|E|$. Concretely, $|E|$ accepting transcripts for the same commitments open every non-isolated vertex and check every edge, so the opened colors form a proper coloring; isolated vertices may be colored arbitrarily.

Why the opening must be an edge. The opening family is forced by privacy, not by efficiency. All-but-one opening would reveal $|V| - 1$ colors, essentially the whole coloring up to relabeling. A uniformly random pair of vertices might be a non-edge, whose shares reveal whether two vertices have the same color. Only edges are both private and checkable. The challenge space is therefore E , and the error $1 - 1/|E|$ is the pair-opening bound of case (c) of Corollary 1, with the pairs of $[N]$ replaced by the edges of G . Ishai et al. remark that the analysis of their pair-opening compiler follows the GMW proof for 3-coloring [IKOS09]. In the present reading, the GMW protocol is that compiler, run on a network shaped like the graph itself.

Remark 17 (Relaxations of Section 3.1). The 3-coloring protocol uses Section 3.1 with three relaxations, all harmless.

- *Privacy only for opened sets.* Definition 4 asks every pair of shares to be uniform, but the proof of Theorem 1 uses privacy only for the sets O_c that the verifier can open, and only to sample the opened shares from a distribution that is the same for every witness. Here the opened sets are the edges, and the opened shares are a uniform pair of distinct colors. Likewise, Definition 8 is needed only for sets T that are edges. This extends the first relaxation of Remark 16, where the opened shares reveal a random vector of weight ω .
- *Channels on a graph.* Only adjacent parties share a channel. Two parties without a channel exchange nothing and are consistent by definition, so the edge covering number of Definition 9 needs to range only over pairs that share a channel. With this reading, the proof of Theorem 3 applies verbatim and gives $\kappa_E = |E|$.
- *Conjunctive outputs.* As in Stern’s protocol, no single party decides the relation; the conjunction of all outputs does. This suffices because the extractor opens every non-isolated party.

5.3 Graph Isomorphism [GMW91, GMW86]

For a permutation τ of $[n]$ and a graph G on the vertex set $[n]$, let $\tau(G)$ be the graph obtained by relabeling each vertex v as $\tau(v)$. Two graphs G_0, G_1 on $[n]$ are isomorphic if $G_1 = \pi(G_0)$ for some permutation π , which is the witness. The protocol of Goldreich, Micali and Wigderson [GMW91] runs as follows.

1. P samples a random permutation σ of $[n]$ and sends $H = \sigma(G_0)$.
2. V sends a random $b \in \{0, 1\}$.
3. P sends $\tau = \sigma$ if $b = 0$, and $\tau = \sigma\pi^{-1}$ if $b = 1$.
4. V accepts if and only if $\tau(G_b) = H$.

We show that this is the broadcast compiler of Figure 2 with two parties, applied to a one-round broadcast protocol on a 2-out-of-2 sharing of the isomorphism. Unlike the protocols of Sections 5.1 and 5.2, it needs no commitment scheme at all.

The sharing. The prover shares π between two parties as

$$s_0 = \sigma, \quad s_1 = \sigma\pi^{-1}, \quad \text{Rec}(s_0, s_1) = s_1^{-1}s_0 = \pi.$$

Reconstruction is defined on every pair, and each share alone is a uniformly random permutation, so the sharing is 1-private in the sense of Definition 4, with the symmetric group S_n as share space. This is the multiplicative 2-out-of-2 sharing of the QR protocol (Section 4.7), with $s_0 = x_t$ and $s_1 = x_tx$ there. Here the group S_n is not abelian, however, so the order of composition matters.

The virtual protocol. The protocol Π_{GI} has a single broadcast round. Party P_b applies its share to the public graph G_b and broadcasts $h_b = s_b(G_b)$, and the output is 1 if and only if $h_0 = h_1$. It is perfectly correct, since $s_1(G_1) = s_0(G_0)$ holds if and only if $G_1 = s_1^{-1}s_0(G_0) = \text{Rec}(s_0, s_1)(G_0)$. It is perfectly 1-private, since given s_b and output 1 the transcript is determined: $h_0 = h_1 = s_b(G_b)$. Figure 7 shows the protocol.

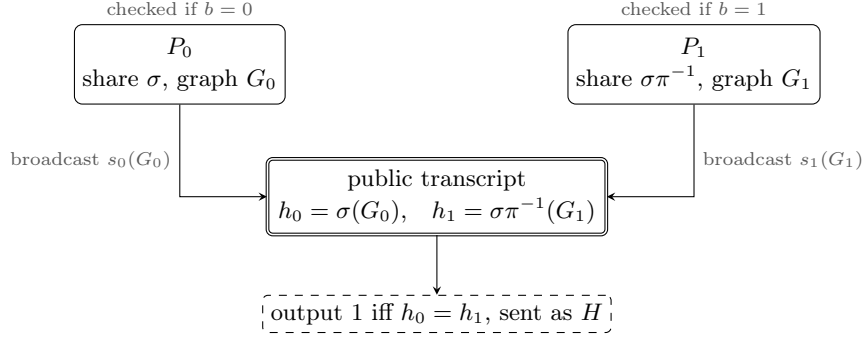


Fig. 7. The virtual two-party protocol Π_{GI} . The parties hold a 2-out-of-2 sharing of the isomorphism π , and each broadcasts its share applied to its own public graph. Since the output condition makes the two broadcasts equal, the prover sends a single graph H , and each opened party is checked against it alone.

Compiling. Compile Π_{GI} with Figure 2 and single opening, which for $N = 2$ is also all-but-one opening. The output condition $h_0 = h_1$ lets the prover send a single graph H , and the challenge b opens party P_b , whose state s_b is checked against the transcript by $s_b(G_b) = H$. This check is a vertex check. GMW’s protocol uses no commitments because the transcript already ties each opened share to public data, as for Schnorr in Section 4.2. One difference is that Lemma 2 does not apply in general: $s_b(G_b) = H$ determines s_b only up to an automorphism of G_b . As for Okamoto (Section 4.4) and QR, this does not matter, because extraction reconstructs from the two opened shares.

Proposition 8. *The protocol is perfectly complete, perfectly special HVZK without any hiding assumption, and 2-special sound: from accepting transcripts $(H, 0, \tau_0)$ and $(H, 1, \tau_1)$, the permutation $\tau_1^{-1}\tau_0$ is an isomorphism from G_0 to G_1 . Moreover, if G_0 and G_1 are not isomorphic, then for every H at most one party can be opened, so the soundness error is $1/2$ even against unbounded provers.*

Proof. Completeness is the identity $\sigma(G_0) = \sigma\pi^{-1}(G_1)$. For special HVZK, given b , the simulator samples a uniform permutation τ and sets $H = \tau(G_b)$; in a real execution $\tau = s_b$ is uniform and $H = s_b(G_b)$, so the distributions are identical. For special soundness, $\tau_0(G_0) = H = \tau_1(G_1)$ gives $\tau_1^{-1}\tau_0(G_0) = G_1$. For soundness, party P_b can be opened only if H is isomorphic to G_b . Isomorphism classes are disjoint, so if G_0 and G_1 are not isomorphic, H can be isomorphic to at most one of them.

With only two challenges, one execution is zero-knowledge even against malicious verifiers, by the guess-and-rewind argument of Remark 6. As for QR, no hiding assumption is needed: the simulated $H = \tau(G_b)$ is a uniformly random graph isomorphic to G_0 whatever b is, so each guess succeeds with probability exactly $1/2$. This is the simulator of [GMW91]. Without commitments, the protocol is a perfect zero-knowledge proof with statistical soundness.

A group action, not a cut-and-choose. Graph isomorphism is usually classed as a combinatorial protocol, but its structure is that of the algebraic protocols of Section 4, with the homomorphism replaced by the action of S_n on graphs. The broadcast graph H is an implicit commitment that fixes the check for both parties, every check is a vertex check, and special soundness is reconstruction from two shares. The three phenomena of Section 4 appear here as well.

- *Kernel.* The automorphisms of G_0 play the role of the kernel of φ in Okamoto’s protocol. There are $|\text{Aut}(G_0)|$ isomorphisms from G_0 to G_1 , the extractor recovers one of them, and the protocol is perfectly witness indistinguishable among them.
- *Cokernel.* Non-isomorphic pairs are false statements. The orbit of G_b under S_n plays the role of the image of φ , and the proof of Proposition 8 is that of Propositions 2 and 4: an implicit commitment lies in at most one of two disjoint orbits.
- *Number of parties.* A third party would hold $\sigma\pi^{-2}$, and checking it requires the graph $\pi^2(G_0)$, which is not public. The action yields implicit commitments only for the public graphs, so there are two parties and the error is $1/2$. With S public graphs $G_k = \pi_k(G_0)$, party P_k holds $\sigma\pi_k^{-1}$, giving $S + 1$ parties and error $1/(S + 1)$. Two opened parties then reconstruct an isomorphism between two public graphs, a relaxed witness. This is the multiple-key technique of isogeny-based identification, which has the same group-action structure.

6 Discussion and Further Instances

The framework of Section 3 covers more classical Sigma protocols than those treated in Sections 4 and 5. We sketch two of them here, isogeny-based identification and Blum’s Hamiltonian-cycle protocol. Both fit with little change, and each raises a point that the framework handles only informally so far. We leave their full treatment as a complement to this work.

6.1 Isogeny-Based Identification

Let a finite commutative group $\mathcal{G}$ act freely and transitively on a set $\mathcal{E}$, written $(g, E) \mapsto g \star E$. The main example is the class group of an imaginary quadratic order acting on a set of supersingular elliptic curves, as in CSIDH [CLM⁺18]. The secret is $a \in \mathcal{G}$ and the public key is $E_1 = a \star E_0$ for a fixed curve E_0 . The basic identification protocol [FG19, BKV19, BFGP23] has the shape of the graph-isomorphism protocol of Section 5.3. The prover samples $b \in \mathcal{G}$ and sends $E' = b \star E_0$. The verifier sends $c \in \{0, 1\}$. The prover answers $s_0 = b$ or $s_1 = ba^{-1}$, and the verifier checks $s_c \star E_c = E'$.

In MPCitH terms, this is Section 5.3 with the action of S_n on graphs replaced by the action of $\mathcal{G}$ on curves. The two parties hold the 2-out-of-2 sharing $(s_0, s_1) = (b, ba^{-1})$ of a , with $\text{Rec}(s_0, s_1) = s_1^{-1}s_0$. Each party broadcasts $s_c \star E_c$, the output is 1 if and only if the two broadcasts agree, and the single curve E' is the transcript. Every check is a vertex check, and two opened parties reconstruct the secret. As for graphs, a third party would need the curve $a^2 \star E_0$, which is not public: unlike the homomorphisms of Section 4, a group action offers no way to compute $a^c \star E_0$ from $a \star E_0$. With S public curves $E_k = a_k \star E_0$, one obtains $S + 1$ parties and error $1/(S + 1)$, which is how SeaSign and CSI-FiSh enlarge the challenge space [FG19, BKV19].

Two differences from graph isomorphism deserve a fuller treatment.

- *No kernel and no cokernel.* Because the action is free and transitive, every statement has exactly one witness and every statement is true. Structurally, isogeny-based identification is therefore Schnorr’s protocol with binary challenge for a group action: only the knowledge error is meaningful.
- *Representing the shares.* Privacy requires an opened share to reveal nothing about a , which depends on how elements of $\mathcal{G}$ are represented. When the structure of the class group is known, as in CSI-FiSh [BKV19], shares are uniform elements of a cyclic group and privacy is perfect. Otherwise, as in SeaSign [FG19], shares are short exponent vectors, and the share ba^{-1} is uniform only after rejection sampling. Privacy is then statistical and obtained by aborting, much as the integer shares of GPS (Section 4.8) are hidden by a large mask. Making rejection sampling a privacy mechanism inside the MPCitH framework is the natural next step, and would also cover lattice-based identification.

6.2 Blum’s Hamiltonian-Cycle Protocol

Let G be a graph on $[n]$ and let the witness be a Hamiltonian cycle C of G , viewed as a set of n edges. In Blum’s protocol [Blu86], the prover samples a random permutation σ of $[n]$ and commits, entry by entry, to the adjacency matrix M of $\sigma(G)$. On challenge $b = 0$, it reveals σ and opens all of M , and the verifier checks $M = \sigma(G)$. On challenge $b = 1$, it reveals the cycle $\sigma(C)$ and opens only the n entries of M on it, and the verifier checks that $\sigma(C)$ is a Hamiltonian cycle on $[n]$ and that all opened entries equal 1.

In MPCitH terms, two parties hold $s_0 = \sigma$ and $s_1 = \sigma(C)$, a 2-out-of-2 sharing of C with $\text{Rec}(s_0, s_1) = s_0^{-1}(s_1)$; each share alone is uniform, a random permutation or a random Hamiltonian cycle on $[n]$. Party P_0 contributes $M = s_0(G)$, and P_1 outputs 1 if and only if every edge of s_1 is an edge of M , which holds exactly when $\text{Rec}(s_0, s_1)$ is a Hamiltonian cycle of G . The committed matrix M is the committed broadcast of Remark 8. Each party is checked against it alone, so these are vertex checks, two parties, and error $1/2$. Two answers for the same committed M give σ and s_1 with $M = \sigma(G)$ and $s_1 \subseteq M$, hence the Hamiltonian cycle $\sigma^{-1}(s_1)$ of G .

The point that deserves a fuller treatment is the partial opening of M . In Blum’s protocol it is needed for privacy, not only for efficiency. Together with the share $s_1 = \sigma(C)$, the whole matrix would reveal the pair $(\sigma(G), \sigma(C))$, a relabeled copy of (G, C) . This can distinguish Hamiltonian cycles of G that are not related by an automorphism. Opening only the entries on $\sigma(C)$ reveals a

random Hamiltonian cycle whose entries are all 1, independently of C . The cleanest reading is a two-party functionality in which P_1 selects the n entries of M on its cycle, an oblivious-transfer-like call, with P_0 's input M committed. The compiler then opens a party together with the part of the committed channel that it uses. This combines the committed channels of Section 3.2 with OT-like functionalities, which the committed-channel variant of [IKOS09] does not support, and making it precise is left to future work.

6.3 Further Directions

The same lens suggests several further complements. Brands' proofs of Boolean formulas over linear relations [Bra97] fit the linear template of Section 4. An AND restricts the mask of the sharing to the solution space of the relations, so that every opened share satisfies them; a NOT re-encodes the witness for a different homomorphism; and an OR secret-shares the challenge. Beyond the classical protocols, modern MPCitH protocols for problems such as syndrome decoding can be read as the large- N counterparts of Stern's protocol, with vertex checks over a broadcast channel and seed trees in place of a few explicitly committed views. Optimizations such as batching, compressed Sigma protocols and VOLE-in-the-head can likewise be read as ways of reducing the views that must be committed and opened. We leave these directions for future work.

Acknowledgment

Initial ideas for this work originated from a 2019 seminar hosted by Man Ho Au at PolyU. Our understanding was subsequently refined through collaborative discussions with Meng Hao, Guomin Yang, and Guofeng Tang at SMU.

The author gratefully acknowledges the use of large language models (Claude and Gemini) for drafting assistance in the preparation of this manuscript. The conceptualization, insights, and final verification of all proofs and claims remain entirely the author's responsibility.

References

- ABP13. M. Abdalla, F. Ben Hamouda, and D. Pointcheval. Tighter reductions for forward-secure signature schemes. In *Public-Key Cryptography – PKC 2013, Lecture Notes in Computer Science* 7778, pages 292–311. Springer, 2013.
- ACK21. T. Attema, R. Cramer, and L. Kohl. A compressed Σ -protocol theory for lattices. In *Advances in Cryptology – CRYPTO 2021, Part II, Lecture Notes in Computer Science* 12826, pages 549–579. Springer, 2021.
- AF22. T. Attema and S. Fehr. Parallel repetition of $(k_1, \dots, k_\mu)$ -special-sound multi-round interactive proofs. In *Advances in Cryptology – CRYPTO 2022, Part I, Lecture Notes in Computer Science* 13507, pages 415–443. Springer, 2022.
- AFLT12. M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly-secure signatures from lossy identification schemes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 572–590. Springer, 2012.
- Bab85. L. Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC '85)*, pages 421–429. ACM, 1985.
- BCC88. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.
- BCK10. E. Bangerter, J. Camenisch, and S. Krenn. Efficiency limitations for Σ -protocols for group homomorphisms. In *Theory of Cryptography – TCC 2010, Lecture Notes in Computer Science* 5978, pages 553–571. Springer, 2010.
- BFGP23. W. Beullens, D. Feo, S. D. Galbraith, and C. Petit. Proving knowledge of isogenies: A survey. *Designs, Codes and Cryptography*, 91(11):3425–3456, 2023.
- BG92. M. Bellare and O. Goldreich. On defining proofs of knowledge. In *Annual International Cryptology Conference*, pages 390–420. Springer, 1992.
- BKV19. W. Beullens, T. Kleinjung, and F. Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Advances in Cryptology – ASIACRYPT 2019, Part I, Lecture Notes in Computer Science* 11921, pages 227–247. Springer, 2019.
- Blu86. M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, page 2, 1986.

- BN20. C. Baum and A. Nof. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In *Public-Key Cryptography – PKC 2020, Part I, Lecture Notes in Computer Science* 12110, pages 495–526. Springer, 2020.
- Bra97. S. Brands. Rapid demonstration of linear relations connected by boolean operators. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 318–333. Springer, 1997.
- BS23. D. Boneh and V. Shoup. A graduate course in applied cryptography, draft 0.6. <https://toc.cryptobook.us/>, 2023.
- CDS94. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology – CRYPTO ’94, Lecture Notes in Computer Science* 839, pages 174–187. Springer, 1994.
- CLM⁺18. W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. CSIDH: An efficient post-quantum commutative group action. In *Advances in Cryptology – ASIACRYPT 2018, Part III, Lecture Notes in Computer Science* 11274, pages 395–427. Springer, 2018.
- CP92. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Annual international cryptography conference*, pages 89–105. Springer, 1992.
- Cra97. R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, 1997.
- FFS88. U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.
- FG19. D. Feo and S. D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In *Advances in Cryptology – EUROCRYPT 2019, Part III, Lecture Notes in Computer Science* 11478, pages 759–789. Springer, 2019.
- FR23. T. Feneuil and M. Rivain. Threshold linear secret sharing to the rescue of MPC-in-the-head. In *Advances in Cryptology – ASIACRYPT 2023, Part I, Lecture Notes in Computer Science* 14438, pages 441–473. Singapore, 2023. Springer.
- GLSY04. R. Gennaro, D. Leigh, R. Sundaram, and W. Yerazunis. Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 276–292. Springer, 2004.
- GMO16. I. Giacomelli, J. Madsen, and C. Orlandi. ZKBoo: Faster zero-knowledge for boolean circuits. In *25th USENIX Security Symposium, USENIX Security 16*, pages 1069–1083, Austin, TX, USA, 2016. USENIX Association.
- GMR89. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.
- GMW86. O. Goldreich, S. Micali, and A. Wigderson. How to prove all np statements in zero-knowledge and a methodology of cryptographic protocol design. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 171–185. Springer, 1986.
- GMW91. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.
- Gol01. O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.
- GPS06. M. Girault, G. Poupard, and J. Stern. On the fly authentication and signature schemes based on groups of unknown order. *Journal of Cryptology*, 19(4):463–487, 2006.
- GQ88. L. C. Guillou and J.-J. Quisquater. A “paradoxical” identity-based signature scheme resulting from zero-knowledge. In *Conference on the Theory and Application of Cryptography*, pages 216–231. Springer, 1988.
- IKOS09. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM Journal on Computing*, 39(3):1121–1152, 2009.
- KKW18. J. Katz, V. Kolesnikov, and X. Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, pages 525–537, Toronto, ON, Canada, 2018. ACM.
- KW03. J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003*, pages 155–164. ACM, 2003.
- Mau15. U. Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. *Designs, Codes and Cryptography*, 77(2):663–676, 2015.
- Oka92. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Annual international cryptography conference*, pages 31–53. Springer, 1992.
- PS98. G. Poupard and J. Stern. Security analysis of a practical “on the fly” authentication and signature generation. In *Advances in Cryptology – EUROCRYPT ’98, Lecture Notes in Computer Science* 1403, pages 422–436. Springer, 1998.

- Sch91. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.
- Ste93. J. Stern. A new identification scheme based on syndrome decoding. In *Annual International Cryptology Conference*, pages 13–21. Springer, 1993.
- ZCYW23. M. Zhang, Y. Chen, C. Yao, and Z. Wang. Sigma protocols from verifiable secret sharing and their applications. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 208–242. Springer, 2023.