Efficiency Lower Bounds for Commit-and-Prove Constructions

Christian Badertscher*, Sandro Coretti†, Chen-Da Liu Zhang*, Ueli Maurer*
*ETH Zurich, Switzerland; Email: {badi, lichen, maurer}@inf.ethz.ch
†New York University, United States; Email: scoretti@cs.nyu.edu

***Abstract*—Commitment schemes that admit zero-knowledge proofs for relations among committed values are known as commit-and-prove functionalities or notarized envelopes. An important role in this context play equality proofs among commitments. They appear in various contexts of multi-party computation, circuit satisfiability or inclusion proofs. Using commit-and-prove functionalities admitting equality, we investigate black-box constructions of commit-and-prove functionalities admitting more complex relations. Typically, these constructions have to create commitments to additional values to achieve a certain level of soundness. An important efficiency measure is the number of such additional commitments. We prove that, for the natural and quite general class of 3-round public-coin zero-knowledge protocols, implementing the inequality relation, or any of the relations NAND, NOR, or XOR, essentially requires at least $2n$ additional commitments in order to achieve a soundness of 2^{-n}. A folklore protocol shows that this bound is tight for inequality.**

I. Introduction

An *interactive proof*, originally introduced by Goldwasser, Micali, and Rackoff [GMR], is an interactive protocol between a *prover* Peggy and a *verifier* Vic. An interactive proof needs to be *complete* and *sound*. It is called complete, if an honest prover succeeds in convincing the honest verifier of true statements. A protocol is called sound, if a dishonest prover cannot convince an honest verifier of false statements. An interactive proof in which Peggy does not convey any information to Vic apart from the fact that the claimed statement is true, is further called a zero-knowledge protocol.

Commitment schemes [Gol] are a fundamental cryptographic primitive and often appear as part of an interactive protocol or in multi-party computation (MPC). A commitment scheme allows Peggy to commit to a chosen value while keeping it hidden from Vic. It guarantees that the value remains hidden from Vic until Peggy decides to *open* the commitment. This is the hiding property. On the other hand, Vic is ensured that once Peggy is committed to a value, she can only open exactly one value, supposedly the one she committed to. This is the binding property.

The combined view of zero-knowledge and commitments is typically known as *notarized envelopes* [Kil92], [BOGG$^+$90], [BFOR90], [BFS90]. They allow Peggy to commit to a set of bits $b_1, \dots, b_n$ and at some point later prove in zero-knowledge that some predicate $P(b_1, \dots, b_n)$ holds for these bits. Constructions of notarized envelopes are known from commitment schemes [Kil92], [BOGG$^+$90] or from oblivious transfer [Kil88]. In MPC, of particular interest are universally composable commitments that admit such zero-knowledge proofs. They are modeled as commit-and-prove functionalities [CLOS02] within a composable framework, such as Canetti's UC Framework [Can01] or the framework of Constructive Cryptography by Maurer and Renner [Mau11], [MR11]. Of special interest are commit-and-prove functionalities that admit equality proofs. Equality proofs are often used to construct MPC protocols [CCD88], [GMW87], zero-knowledge proofs for circuit satisfiability [BCC87] or inclusion proofs [CL02].

Another core task is to construct, based on the above equality proofs, protocols that prove more complex relations in zero-knowledge based. Of specific interest are inequality of two bits [LLX07], [CvHP91], or NAND of three bits. The efficiency of such constructions is typically measured as the number of additional commitments that the protocol uses to achieve a certain level of soundness [KMO89]. In this work, we prove efficiency lower bounds of such constructions.

A. A Folklore Protocol for Inequality

We present a folklore protocol to prove that two commitments to secret bits b_1 and b_2 contain different values. Note that this means proving the statement $b_1 \neq b_2$ without revealing the secret bits: First, Peggy chooses two additional bits (e_1, e_2), uniformly over the set $\{(0,1),(1,0)\}$, and commits to them. In the second step, Vic chooses a challenge $c \in \{1,2\}$ and sends c to Peggy. If $c = 1$, Peggy opens the two additional bits to show to Vic that they are different. If $c = 2$, Peggy proves to Vic, in zero-knowledge, equality of the two committed values b_1 and e_i (where Peggy selects the correct i according to the additional bits) and proves equality of the two committed values b_2 and e_{3-i}. This protocol is zero-knowledge: all the prover reveals is either two bits that are independent of the secret bits (if $c = 1$), or two zero-knowledge equality proofs for a random choice of i (if $c = 2$). This does not reveal anything about the secret values. Also, since a cheating prover cannot correctly answer both challenges in case $b_1 = b_2$, we see that the probability of cheating, i.e., the soundness, is one-half. By repeating this protocol for t rounds, we can achieve soundness 2^{-t} by requiring $2t$ additional commitments.

An interesting open question for this type of zero-knowledge protocol is: is it actually possible to achieve a better soundness than one-half with two additional commitments? And with three additional commitments, can one improve the soundness one-half? In general, using at most n additional commitments, what is the best challenge space, i.e., a combination of equality checks and openings that does not reveal values of the secret bits, that minimizes the probability that Vic accepts in case $b_1 = b_2$? In this work, we settle these questions proving that no protocol of the above kind can be more efficient in terms of additional commitments to achieve a soundness of 2^{-t}. This further implies that constructions for more complex relations such as NAND, NOR or XOR are subject to this lower bound.

II. Preliminaries and Notation

In this work, we use the following conventions: Given a set A, we denote by 2^A the powerset of A. For a distribution D on A, $\mathsf{supp}(D) \subseteq A$ denotes the support of D, and we denote by $D(a)$ the probability assigned to element $a \in A$. For tuples $X_n = (x_1, \ldots, x_n) \in A^n$ and $Y_k = (y_1, \ldots, y_k) \in A^k$ $(n, k > 0)$, (X_n, Y_k) denotes concatenation, i.e., $(x_1, \ldots, x_n, y_1, \ldots, y_k)$. Finally, a multiset M is a generalization of a set in the sense that it allows multiple instances of its elements. Given a multiset M and an ordinary set A, we write $M \subseteq A$ to express that $\forall x \in M : x \in A$.

A. Commit-and-Prove Functionality

Commit-and-Prove functionalities are systems, i.e., random systems [Mau02], with one interface P for the prover and one interface V for the verifier. The behavior at these interfaces can roughly be described as follows: The prover can store bits within the resource (representing idealized commitments) and subsequently prove to the verifier that a certain number of stored bits, identified by their location, fulfill a relation R. The verifier does not learn anything beyond the validity of this statement. The prover can also reveal certain values to the verifier (representing idealized openings of commitments).

Definition 1 (Commit-and-Prove)**.** *A commit-and-prove functionality or resource $\mathsf{CP}_{R,n}$ is parameterized by a k-ary relation R and the number n of bits that can be stored within the resource. We denote by x_i the value stored at position i. Initially, $x_i = \perp$ for all positions i. The behavior is as follows:*

1) *On input (store, b, i), $b \in \{0, 1\}$ and $i \in [1, n]$, at interface P, store b at position i if the value is undefined, i.e., set $x_i \leftarrow b$ and output (stored, i) at interface V.*
2) *On input (open, i) at interface P, if the value at position i is defined, i.e., $x_i \neq \perp$, output (x_i, i) at interface V. Otherwise, ignore the input.*
3) *On input $(\mathsf{prove}, i_1, \ldots, i_k)$ at interface P, output $(\mathsf{accept}, i_1, \ldots, i_k)$ at interface V if (and only if) all values $x_{i_1}, \ldots, x_{i_k}$ are defined and $R(x_{i_1}, \ldots, x_{i_k}) = 1$. Otherwise, do not output anything.*

III. Formalization of the Problem

In this work, we assume commit-and-prove resources where the relation R denotes equality of two bits. We denote this resource by $\mathsf{CP}_{=,n}$. The goal of a construction is to realize a stronger commit-and-prove functionality which in addition supports proving that two bits are different. We denote this resource by $\mathsf{CP}_{\neq,n'}$. We show in the next section that any protocol that realizes $\mathsf{CP}_{\neq,n'}$, requires an overhead of at least 2ℓ additional commitments per inequality proof in order to get the cheating probability down to $2^{-\ell}$.

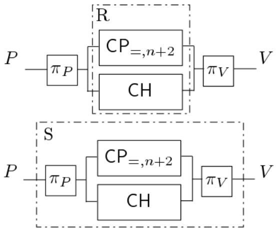

Figure 1: Attaching converters to a resource.

To prove this lower bound, we focus on the particular problem of realizing $\mathsf{CP}_{\neq,2}$, i.e., a resource that can store exactly the two secret bits and prove that they are different, from $\mathsf{CP}_{=,n+2}$, i.e., from a resource that can store the two secret bits, but also can store n "helper bits" that can be used by a protocol to implement the inequality proof.

A. The Construction Notion of Constructive Cryptography

A construction is achieved by means of a protocol between the prover Peggy and the verfier Vic. A protocol π is modeled as a pair of *converters* (π_P, π_V) that specify the actions for both parties. As illustrated in Figure 1, each party attaches its converter to the interfaces of the assumed resource $\mathbf{R} = [\mathsf{CH}, \mathsf{CP}_{=,n+2}]$, which consists of a commit-and-prove functionality and a standard communication channel CH (as in [Mau11]). Peggy's interface P is on the left and Vic's interface V on the right. Attaching the converters changes the local behavior at the interfaces and hence yields the new resource $\mathbf{S} = \pi_P[\mathsf{CH}, \mathsf{CP}_{=,n+2}]\pi_V$. To show that a protocol constructs $\mathsf{CP}_{\neq,2}$ from $\mathsf{CP}_{=,n+2}$ and CH, we have to prove three conditions that are derived from the general construction notion of constructive cryptography in [MR11], [Mau11].

Random experiments. The three conditions make statements about random experiments $\mathbf{DR}$ in which a distinguisher $\mathbf{D}$ plays the role of an interactive environment for some resource $\mathbf{R}$. The distinguisher $\mathbf{D}$ is a system that provides inputs to the connected resource and receives the outputs generated by the resource. For example, $\mathbf{D}(\pi_P[\mathsf{CH}, \mathsf{CP}_{=,n+2}]\pi_V)$ is the experiment that captures "the protocol in action" in the environment provided by $\mathbf{D}$. More concretely, in each step of these experiments, the distinguisher provides an input to one of the interfaces and observes the output that is generated in reaction to that input. This process continues iteratively by having $\mathbf{D}$ providing adaptively the next input and receiving the next output. The experiment ends by $\mathbf{D}$ outputting a bit 0 or 1 that indicates its guess to which system it is connected. The *distinguishing advantage* of $\mathbf{D}$ for two resources $\mathbf{R}$ and $\mathbf{S}$ is defined as $\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) := |\Pr[\mathbf{DR} = 1] - \Pr[\mathbf{DS} = 1]|$.

The three conditions of constructive cryptography in our two-party setting are depicted in Figure 2. The first condition places a bound on the advantage of any distinguisher $\mathbf{D}$ in distinguishing the left system $\pi_P[\mathsf{CH}, \mathsf{CP}_{=,n+2}]\pi_V$ and the right system $\mathsf{CP}_{\neq,2}$, i.e., on

$$\Delta^{\mathbf{D}}(\pi_P[\mathsf{CH}, \mathsf{CP}_{=,n+2}]\pi_V, \mathsf{CP}_{\neq,2}). \tag{1}$$

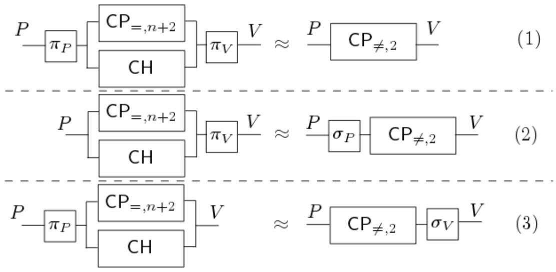

Figure 2: Illustration of the construction notion.

It ensures that the constructed resource is achieved if both parties honestly apply the protocol.

Simulating the actions of a dishonest prover. The second condition requires that all attacks of a dishonest prover, who does not attach its converter but uses its interface to the assumed resource to follow an arbitrary strategy, can be translated by means of a simulator σ_P to an attack on the constructed resource. Turned around, since the resource on the right does not allow the prover to cheat by definition, there cannot be a successful attack in the real world depicted on the left. Formally, the condition places, for any distinguisher $\mathbf{D}$, a bound on the advantage

$$\Delta^{\mathbf{D}}([\mathsf{CH}, \mathsf{CP}_{=,n+2}]\pi_V, \sigma_P\mathsf{CP}_{\neq,2}). \qquad (2)$$

Simulating the actions of a dishonest verifier. The third condition captures that any attack of dishonest Vic in the real world (on the left), who tries to provoke Peggy to reveal more information on the committed bits beyond the fact that they are not equal, can be translated by a simulator σ_V to an attack on the constructed resource. Since the constructed resource does not reveal anything beyond the validity of the statement to Vic, there cannot be a successful attack on the protocol on the left-hand side. The third condition places, for any distinguisher $\mathbf{D}$, a bound on the advantage

$$\Delta^{\mathbf{D}}(\pi_P[\mathsf{CH}, \mathsf{CP}_{=,n+2}], \mathsf{CP}_{\neq,2}\sigma_V). \qquad (3)$$

B. General Protocol Structure

To prevent cheating, Vic needs to demand from Peggy to assure to him certain properties about the idealized commitments stored in $\mathsf{CP}_{=,n+2}$: for example showing that a certain value b is stored at position i by opening it, or proving equality of two positions. We now provide the necessary definitions.

Definition 2 (Instructions, actions, challenges)**.** *Let $n' \in \mathbb{N}$. We define the set of instructions $\mathcal{I}_{n'} = \mathsf{OP} \cup \mathsf{EQ}$, where*

$$\mathsf{OP} = \{(\mathsf{open}(i); b) : i \in [n'] \wedge b \in \{0,1\}\}$$
$$\mathsf{EQ} = \{(\mathsf{equal}, i, j) : i, j \in [n']\}\,.$$

An action *a is a non-empty subset of instructions, i.e., $a \in \mathcal{A}_{n'}$ where $\mathcal{A}_{n'}$ is the set of all possible actions $2^{\mathcal{I}_{n'}} \setminus \{\varnothing\}$. A* challenge *$c$ is a non-empty set of actions. The challenge space $\mathcal{C}$ of a protocol is a set of challenges (see Definition 4 below).*

Definition 3 (Matching instructions)**.** *Let $n' \in \mathbb{N}$. For a vector $X \in \{0,1,\perp\}^{n'}$ and an instruction α, we define the predicate* match *as follows:* $\mathsf{match}(X, \alpha) = 1$ *if and only if*

1.) $\alpha = (\mathsf{open}(i); b)$, $i \le n'$, $x_i \neq \perp$, and $x_i = b$
or
2.) $\alpha = (\mathsf{equal}, i, j)$, $i, j \le n'$, $x_i, x_j \neq \perp$, and $x_i = x_j$.

For an action a, the induced predicate (abusing a bit of notation) match *is defined as follows:* $\mathsf{match}(X, a) = 1$ *if and only if $\forall \alpha \in a : \mathsf{match}(X, \alpha) = 1$.*

Definition 4 (Protocol class Γ)**.** *A protocol (π_P, π_V) belongs to the protocol class Γ of 3-round public coin protocols that assume resource $\mathsf{CP}_{=,n+2}$ and prove inequality of two committed input bits, if they have the following structure:*

Setup: The protocol is parameterized by an arbitrary challenge space $\mathcal{C}$ as in Definition 2 for $n' = n + 2$. For simplicity, we assume Peggy stores the two input bits, denoted x_{n+1} and x_{n+2}, at locations $n + 1$ and $n + 2$ of $\mathsf{CP}_{=,n+2}$.

Round 1: Peggy chooses a vector of additional values $X_n = (x_1, \ldots, x_n)$ according to a distribution $D_{x_{n+1},x_{n+2}}$ on $\{0,1,\perp\}^n$. Peggy stores each additional bit $x_i \neq \perp$ at position i of resource $\mathsf{CP}_{=,n+2}$. The contents of $\mathsf{CP}_{=,n+2}$ can be described by $X_{n+2} := (X_n, x_{n+1}, x_{n+2})$.

Round 2: Vic chooses a challenge $c \in \mathcal{C}$ uniformly at random and sends c, i.e., a description of it, to Peggy via channel CH*.*

Round 3: Peggy chooses an action a from the given challenge c according to a distribution $D'_{X_{n+2},c}$ on the set $\{a \in c \mid \mathsf{match}(X_{n+2}, a) = 1\}$ (and aborts in case this set is empty). Peggy then announces a towards Vic and executes each instruction α contained within the chosen action. This means the following: For each $\alpha = (\mathsf{open}(i); b)$ Peggy instructs $\mathsf{CP}_{=,n+2}$ to open the bit at position i towards Vic, who verifies that the output of $\mathsf{CP}_{=,n+2}$ equals (b, i). For each $\alpha = (\mathsf{equal}, i, j)$ Peggy instructs $\mathsf{CP}_{=,n+2}$ to assure that the bits i and j are equal, and Vic verifies that the output of $\mathsf{CP}_{=,n+2}$ equals (accept, i, j). If any of Vic's checks do not succeed, or if he observes that Peggy stores new bits in the resource, he aborts. Vic accepts if he did not abort in round 3.

Definition 5 (View)**.** *The view of an execution of a protocol $\pi \in \Gamma$ with n additional commitments as in Definition 4 with an honest prover, consists of all outputs generated by $\mathsf{CP}_{=,n+2}$ in round 1 (notifications of store-commands), round 3 (openings and equality proofs), and the action a chosen by the prover. We denote by $\mathsf{View}^{\pi}_{X_{n+2}}(c)$ the random variable, over the randomness in round 3, that describes the view as a function of the chosen challenge c and conditioned on the contents X_{n+2} of $\mathsf{CP}_{=,n+2}$ (cf. Definition 4). We denote by View^{π} the set of all possible views.*

C. Important Combinatorial Properties

Definition 6 (Completeness)**.** *Let $\pi \in \Gamma$ be a protocol with n additional commitments as in Definition 4. Let $x_{n+1} \neq x_{n+2}$ denote the two secret bits. We say π is* complete *if for all $X_n \in \mathsf{supp}(D_{x_{n+1},x_{n+2}})$ and all challenges $c \in \mathcal{C}$, we have that $|\{a \in c \mid \mathsf{match}(X_{n+2}, a) = 1\}| > 0$ (for X_{n+2} defined as in Definition 4).*

If the above is fulfilled, the distinguishing advantage in Equation (1) is zero and vice-versa.

Definition 7 (Soundness)**.** *Let $\pi \in \Gamma$ be a protocol with n additional commitments as per Definition 4. Let $\mathcal{C}_X := \{c \in \mathcal{C} \mid \exists a \in c : \mathsf{match}(X, a) = 1\}$ and let $\mathcal{X} := \{X_{n+2} \in \{0,1\}^{n+2} \mid x_{n+1} = x_{n+2}\}$. The soundness of π is defined as $\mathsf{sound}(\pi) := \frac{1}{|\mathcal{C}|} \cdot \max_{X \in \mathcal{X}} |\mathcal{C}_X|$.*

Elements of set $\mathcal{X}$ in Definition 7 describe the strategies with which a dishonest prover can try to cheat in case $x_{n+1} = x_{n+2}$. A cheating attempt is only successful, if the tuple stored in round 1 of a protocol run can be used to convince the verfier in round 3. Hence, $\mathsf{sound}(\pi)$ upper bounds the distinguishing advantage in Equation (2) and vice-versa.

Definition 8 (Zero-Knowledge)**.** *Let $\pi \in \Gamma$ be a protocol with n additional commitments as in Definition 4. The protocol π is zero-knowledge if $\forall c \in \mathcal{C}\ \forall v \in \mathsf{View}^\pi$:*

$$\sum_{X_n \in \mathsf{supp}(D_{0,1})} D_{0,1}(X_n) \cdot \Pr[\mathsf{View}^\pi_{(X_n,0,1)}(c) = v]$$
$$= \sum_{X_n \in \mathsf{supp}(D_{1,0})} D_{1,0}(X_n) \cdot \Pr[\mathsf{View}^\pi_{(X_n,1,0)}(c) = v].$$

The above property directly implies a simulator σ_V that can simulate the view perfectly without knowledge of the secret bits yielding advantage zero in Equation (3). The reverse is also true: the existence of a perfect simulator σ_V implies the above condition.

IV. Efficiency Lower Bounds

In this section, we prove a lower bound for inequality proofs and show the implications on more complex relations.

Theorem 1 (Lower bound, Main theorem)**.** *A protocol $\pi \in \Gamma$ which is complete and zero knowledge, and which proves the inequality of two secret bits x_{n+1}, x_{n+2} using (at most) n additional commitments, has soundness $\mathsf{sound}(\pi) \geq 2^{-\lfloor \frac{n}{2} \rfloor}$.*

We first state two technical lemmata.

Definition 9. *Let $n \in \mathbb{N}$ and let $A \subseteq \mathcal{A}_{n+2}$ be a set of actions and let M be a multiset with elements $a \in A$ and let $\mathcal{X} := \{X \in \{0,1\}^{n+2} \mid x_{n+1} = x_{n+2}\}$. Then we define the following quantity: $s(M) := \max_{X \in \mathcal{X}} \sum_{a \in M} \mathsf{match}(X, a)$.*

Lemma 1. *Let $n \in \mathbb{N}$. Let $\mathcal{A}_{res} \subseteq \mathcal{A}_{n+2}$ be the set of actions a that satisfy the following two requirements:*

1.) The action a contains exactly n instructions, i.e., $a = \{\alpha^a_1, \ldots, \alpha^a_n\}$.

2.) $\forall i : \alpha^a_i \in \{(\mathsf{open}(i); b), (\mathsf{equal}, i, n+k)\}$ for some $b \in \{0,1\}$ and $k \in \{1,2\}$.

Let further $l \in \mathbb{N}$, let $X_n \in \{0,1\}^n$, and let $(x_{n+1}, x_{n+2}) \in \{(0,1),(1,0)\}$. Consider any set $A \subseteq \mathcal{A}_{res}$ such that for all $a \in A$, it holds that $\mathsf{match}((X_n, x_{n+1}, x_{n+2}), a) = 1$. Then, the size $|M|$ of any multiset $M \subseteq A$ with $s(M) \leq l$ is bounded by $l \cdot 2^{\lfloor \frac{n}{2} \rfloor}$.

Proof. We prove the lemma for the case $(x_{n+1}, x_{n+2}) = (0,1)$. The case $(x_{n+1}, x_{n+2}) = (1,0)$ is symmetric and omitted. Observe that $|A| \leq 2^n$. For the sake of the argument, assume that the bitstring X_n has k zeros and $n-k$ ones, where $k \leq \lfloor \frac{n}{2} \rfloor$. The proof is analogous for k being the number of ones of X_n. We define an equivalence relation on set A:

$$a_1 \sim a_2 :\leftrightarrow \exists Y_n \in \{0,1\}^n :$$
$$\mathsf{match}((Y_n,1,1), a_1) = \mathsf{match}((Y_n,1,1), a_2) = 1.$$

It is straightforward to see that this relation is symmetric, reflexive, and transitive. We further define the following two sets of actions: $S_0 = \{(\mathsf{open}(i); 0), (\mathsf{equal}, i, n+1) : i \in [n]\}$ and $S_1 = \{(\mathsf{open}(i); 1), (\mathsf{equal}, i, n+2) : i \in [n]\}$. We argue that for any two actions $a_1, a_2 \in A$, $a_1 \sim a_2$ is equivalent to the following condition:

$$\forall i \in [n] : \alpha^{a_1}_i \in S_0 \land \alpha^{a_2}_i \in S_0 \rightarrow \alpha^{a_1}_i = \alpha^{a_2}_i. \qquad (4)$$

To see this, assume that there exists an $i \in [n] : \alpha^{a_1}_i \in S_0 \land \alpha^{a_2}_i \in S_0 \land \alpha^{a_1}_i \neq \alpha^{a_2}_i$. Then $\alpha^{a_1}_i = (\mathsf{open}(i); 0)$ and $\alpha^{a_2}_i = (\mathsf{equal}, i, n+1)$ or vice versa. But if $\exists Y_n : \mathsf{match}((Y_n,1,1), a_1) = 1$ then $y_i = 0$, and if $\exists Y_n : \mathsf{match}((Y_n,1,1), a_2) = 1$ then $y_i = 1$. Hence, $a_1 \not\sim a_2$.

Conversely, define Y_n bit-wise as follows: if $\alpha^{a_1}_i, \alpha^{a_2}_i \in S_1$, choose $y_i = 1$. If $\alpha^{a_1}_i = \alpha^{a_2}_i = (\mathsf{open}(i); 0)$, choose $y_i = 0$. And finally, if $\alpha^{a_1}_i = \alpha^{a_2}_i = (\mathsf{equal}, i, n+1)$, choose $y_i = 1$. Other combinations of instructions $\alpha^{a_1}_i$ and $\alpha^{a_2}_i$ do not occur by definition of the set A. Now, due to Equation (4), we can bound the number of partitions induced by the above equivalence relation on A. To this end, consider an arbitrary action $a \in A$. We know that $\mathsf{match}((X_n,0,1), a) = 1$ and we conclude that for $a = \{\alpha^a_1, \ldots, \alpha^a_n\}$, $\alpha^a_i \in S_b$ if and only if $b = x_i$. Put differently, a partition is uniquely characterized by specifying, for each position i where $x_i = 0$, which of the two possible actions in S_0 is chosen as α^a_i (the other positions do not matter). Since this characterizes a partition, the number of partitions is at most 2^k. Consider now a multiset $M \subseteq A$ with $s(M) \leq l$. We immediately observe that $\sum_{a \in M} \mathsf{match}((Y_n,1,1), a) \leq l$ for all $Y_n \in \{0,1\}^n$. This implies that no more than l actions per partition of A can be contained in M. The largest set M that fulfills this requirement has size $|M| \leq l \cdot 2^k \leq l \cdot 2^{\lfloor \frac{n}{2} \rfloor}$, concluding the statement. □

Lemma 2. *Let $n \in \mathbb{N}$. Let $\mathcal{A}_{zk} \subseteq \mathcal{A}_{n+2}$ be the set of actions a that satisfy the following requirement:*

1.) The action a does neither contain $(\mathsf{open}(n+1); b')$ nor $(\mathsf{open}(n+2); b')$ for some $b' \in \{0,1\}$.

2.) If there exists an $i \in [n]$ such that $(\mathsf{open}(i); b) \in a$ for some b, then, for any $k > 0$, $i_j \in [n+2]$ ($j = 1 \ldots k$), action a does not contain any subset of the form $\{(\mathsf{equal}, i_1, i_2), \ldots, (\mathsf{equal}, i_{k-1}, i_k)\}$, where $i \in \{i_1, i_2\} \land ((i_{k-1} > n) \lor (i_k > n))$.

Let further $l \in \mathbb{N}$, let $X_n \in \{0,1\}^n$, and let $(x_{n+1}, x_{n+2}) \in \{(0,1),(1,0)\}$. Consider any set $A \subseteq \mathcal{A}_{zk}$ such that for all $a \in A$, it holds that $\mathsf{match}((X_n, x_{n+1}, x_{n+2}), a) = 1$. Then, the size $|M|$ of any multiset $M \subseteq A$ with $s(M) \leq l$ is bounded by $l \cdot 2^{\lfloor \frac{n}{2} \rfloor}$.

Proof. As above, we give a proof for the case $(x_{n+1}, x_{n+2}) = (0,1)$ (the other case is symmetric). We first transform the assumed multiset M according to the following procedure:

```
Transformation $\mathbf{T}(M)$
Input: Multiset $M \subseteq \mathcal{A}_{zk}$
  $M' \leftarrow$ empty multiset
  for each $a \in M$ do
    $a' \leftarrow$ empty action
    Partition the set of indices: $i, j \in [n]$ are in the same partition if and only if:
      There is a subset $\{(\mathsf{equal}, i_1, i_2), \ldots, (\mathsf{equal}, i_{l-1}, i_l)\} \subseteq a$,
      with $i_1, \ldots, i_l \in [n]$, $l > 0$, $i \in \{i_1, i_2\}$ and $j \in \{i_{l-1}, i_l\}$.
    for each partition $P \subseteq [n]$ do
      if there is $(\mathsf{equal}, i, j) \in a$, with $i \in P, j \in \{n+1, n+2\}$ then
        for each $k \in P$ do
          Add the instruction $(\mathsf{equal}, k, j)$ to $a'$.
      else if there is $(\mathsf{equal}, j, i) \in a$, with $i \in P, j \in \{n+1, n+2\}$ then
        for each $k \in P$ do
          Add the instruction $(\mathsf{equal}, k, j)$ to $a'$.
      else
        for each $k \in P$ do
          Add the instruction $(\mathsf{open}(i); x_i)$ to $a'$.
    Add action $a'$ to multiset $M'$
  return $M'$
```

The following four important properties hold:

1.) $\mathbf{T}(M) \subseteq \mathcal{A}_{res} \subseteq \mathcal{A}_{zk}$; 2.) $\mathsf{match}((X_n, 0, 1), a') = 1$ for each $a' \in \mathbf{T}(M)$; 3.) $s(\mathbf{T}(M)) \leq s(M) \leq l$; and 4.) $|\mathbf{T}(M)| = |M|$. By properties 1.)-3.) and Lemma 1, we conclude that $|\mathbf{T}(M)| \leq l \cdot 2^{\lfloor \frac{n}{2} \rfloor}$. By property 4.), we can conclude that $|M| \leq l \cdot 2^{\lfloor \frac{n}{2} \rfloor}$. This proves the statement.

Let us justify properties 1.) to 4.): To see 1.), observe that $\mathsf{match}((X_n, 0, 1), a) = 1$ implies that if a contains an instruction $(\mathsf{open}(i); b)$ or $(\mathsf{equal}, i, n+1+b)$ for some $i \in P, b \in \{0, 1\}$, then $b = x_i$. Since $M \subseteq \mathcal{A}_{zk}$, a can not contain both. Hence, the transformation is well-defined in each partition P. Each action $a \in M$ is thus transformed into an action $a' \in \mathcal{A}_{res}$. To see 2.) note that for each $a \in M$ we have $\mathsf{match}((X_n, 0, 1), a) = 1$, and we design a' to either contain open instructions to bits of X_n, or equality proofs that are already implicitly required by a. To see 3.), further observe that for all $b \in \{0, 1\}$ and $Y_n \in \{0, 1\}^n$, $\mathsf{match}((Y_n, b, b), a') \leq \mathsf{match}((Y_n, b, b), a)$. Thus, quantity $s(\cdot)$ cannot increase. Property 4.) holds since for each $a \in M$, exactly one action is inserted into multiset $M' = \mathbf{T}(M)$. □

Proof of Main Theorem. Given an arbitrary protocol $\pi \in \Gamma$ with challenge space $\mathcal{C}$, let us define l as the maximum number of challenges that a dishonest prover can simultaneously prepare for, i.e., $l := \max_{X \in \mathcal{X}} |\mathcal{C}_X|$, where, as in Definition 7, $\mathcal{C}_X := \{c \in \mathcal{C} \mid \exists a \in c : \mathsf{match}(X, a) = 1\}$ and $\mathcal{X} := \{X \in \{0, 1\}^{n+2} \mid x_{n+1} = x_{n+2}\}$. To lower bound the soundness $\mathsf{sound}(\pi) = \frac{l}{|\mathcal{C}|}$, we upper bound the size of $\mathcal{C}$ using the previous Lemmata as follows: Let us fix $X_n \in \mathsf{supp}(D_{0,1})$. By Definition 6, for any challenge $c \in \mathcal{C}$ the prover is able to choose $a \in c$ with $\mathsf{match}((X_n, 0, 1), a) = 1$ with strictly positive probability. We call such an action a a *successful answer* to challenge c. From $\mathcal{C}$ we construct the multiset M containing for each challenge $c \in \mathcal{C}$ exactly one of those successful answers to the challenge. Hence, $M \subseteq A$, where $A \subseteq \{a \in \bigcup_{c \in \mathcal{C}} c \mid \mathsf{match}((X_n, 0, 1), a) = 1\}$. Since π is zero-knowledge, we must have $M \subseteq \mathcal{A}_{zk}$, as otherwise, with non-zero probability, the values of the secret bits would be revealed by the honest prover. By construction, we have $|M| = |\mathcal{C}|$ and $s(M) \leq l$. By Lemma 2 we get $|M| \leq l \cdot 2^{\lfloor \frac{n}{2} \rfloor}$ and conclude that $\mathsf{sound}(\pi) = \frac{l}{|\mathcal{C}|} \geq 2^{-\lfloor \frac{n}{2} \rfloor}$. □

Our result can be generalized to more complex relations. We show this in the following corollary for the NAND relation. Analogous corollaries can be made for relations XOR or NOR.

Corollary 1 (Lower bound, complex relations)**.** *Assume a protocol class Γ' defined like Γ, but instead of inequality of two bits, the NAND relation of three bits x_{n+1}, x_{n+2}, x_{n+3} is proven. Let the protocol $\pi \in \Gamma'$ be complete and zero-knowledge and use (at most) $n - 1$ additional commitments. Then π has soundness at least $2^{-\lfloor \frac{n}{2} \rfloor}$.*

Proof. Note that $a \neq b$ if and only if (a, a, b) satisfies the NAND relation. Assume there is a complete, zero-knowledge protocol $\pi \in \Gamma'$ that uses $n - 1$ additional commitments and has soundness smaller than $2^{-\lfloor \frac{n}{2} \rfloor}$. We construct a protocol $\pi' \in \Gamma$ for inequality that uses n additional commitments and has soundness $\mathsf{sound}(\pi') < 2^{-\lfloor \frac{n}{2} \rfloor}$, in contradiction to Theorem 1. The protocol π' to prove inequality of two bits x_{n+1}, x_{n+2} is as follows. First, Peggy additionally commits to a bit $x_n = x_{n+1}$ and then runs π on the three input values x_n, x_{n+1}, x_{n+2} and finally proves equality of x_n and x_{n+1} to Vic in zero-knowledge. The protocol π' uses no more than n additional commitments, is complete and zero-knowledge, and has the same soundness as π. □

References

[BCC87] G. Brassard, D. Chaum, and C. Crepeau. Minimum disclosure proofs of knowledge. *Computer and System Sciences 37*, 1987.

[BFOR90] D. Beaver, J. Feigenbaum, R. Ostrovsky, and P. Rogaway. Security with low communication overhead. *CRYPTO*, 1990.

[BFS90] D. Beaver, J. Feigenbaum, and V. Shoup. Hiding instances in zero-knowledge proof systems. *CRYPTO*, 1990.

[BOGG$^+$90] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Hastad, J. Kilian, S. Micali, and P. Rogaway. Everything provable is provable in zero-knowledge. *CRYPTO*, 1990.

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. *FOCS*, 2001.

[CCD88] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. In *STOC*, 1988.

[CL02] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO*, 2002.

[CLOS02] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, 2002.

[CvHP91] D. Chaum, E. van Heijst, and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *CRYPTO*, 1991.

[GMR] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput. 18(1)*.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. *STOC*, 1987.

[Gol] O. Goldreich. *Foundations of Cryptography: Volume 1.*

[Kil88] J. Kilian. Founding cryptography on oblivious transfer. In *STOC*, 1988.

[Kil92] J. Kilian. A note on efficient zero-knowledge proofs and arguments. *STOC*, 1992.

[KMO89] J. Kilian, S. Micali, and R. Ostrovsky. Minimum resource zero knowledge proofs. In *FOCS*, 1989.

[LLX07] J. Li, N. Li, and R. Xue. Universal accumulators with efficient nonmembership proofs. In *ACNS*, 2007.

[Mau02] U. Maurer. Indistinguishability of random systems. *EUROCRYPT*, 2002.

[Mau11] U. Maurer. Constructive cryptography - a new paradigm for security definitions and proofs. *TOSCA*, 2011.

[MR11] U. Maurer and R. Renner. Abstract cryptography. *In ICS*, 2011.