Contamination in Cryptographic Protocols

Nan Yang

A Thesis

in

The Department

of

Computer Science & Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Computer Science) at

Concordia University

Montréal, Québec, Canada

March 2021

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Nan Yang**
Entitled: **Contamination in Cryptographic Protocols**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Rabindranath Raut

_____ External Examiner
Dr. Frédéric Dupuis

_____ External to Program
Dr. Chantal David

_____ Examiner
Dr. Amr Youssef

_____ Examiner
Dr. Denis Pankratov

_____ Supervisor
Dr. Jeremy Clark

_____ Co-supervisor
Dr. Claude Crépeau

Approved by _____
Lata Narayanan, Chair
Department of Computer Science & Software Engineering

_____ 2021

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Contamination in Cryptographic Protocols

Nan Yang, Ph.D.

Concordia University, 2021

We discuss a foundational issue in multi-prover interactive proofs (MIP) which we call "contamination" by the verifier. We propose a model which accounts for, and controls, verifier contamination, and show that this model does not lose expressive power. A new characterization of zero-knowledge naturally follows. We show the usefulness of this model by constructing a practical MIP for $\mathbf{NP}$ where the provers are spatially separated. Finally, we relate our model to the practical problem of e-voting by constructing a functional voter roster based on distributed trust.

Acknowledgments

I would like to thank my mom, Renato, all of my friends, and every teacher I ever had.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 A Blind Spot in Interactive Proofs

An *interactive proof* is a dialog between two parties: an efficient *verifier* and an all-powerful but possibly malicious *prover* [Babai (1985); Goldwasser, Micali, and Rackoff (1989)]. The prover is attempting to convince the verifier of some computational claim (such as "this graph is not 3-colorable"). If the claim is true, the prover should succeed almost all the time; if not, the prover should fail almost all the time. This is a generalization of the complexity class **NP**, except instead of simply being handed a polynomial-sized witness, the verifier is allowed to interact with the prover.

The *multi-prover interactive proof* (MIP) model was introduced in Ben-Or, Goldwasser, Kilian, and Wigderson (1988). This model consists of multiple, "non-communicating" provers talking to a single verifier. Like a detective interrogating multiple suspects in isolation, the verfier is able to cross-examine the provers and, as a result, is able to correctly accept more computationally difficult claims compared to the single-prover model.

We will call one-verifier-multi-provers *the standard MIP model*. The standard MIP model is the one which is used in existing literature on multi-prover interactive proofs. However, we believe that it has a glaring disadvantage. We invite the readers to consider the following ridiculous two-prover

interactive proof:

Protocol 1.1. *(Ridiculous Interactive Proof)*

(1) Verifier sends Prover 1 a random string S.

(2) Prover 1 replies with a string T.

(3) Verifier sends Prover 2 the string T.

(4) Prover 2 replies with a string S'.

(5) Verifier accepts if $S = S'$.

Suppose that we claim the following ridiculous theorem:

Theorem 1.1. *(Ridiculous Theorem) The probability that the verifier accepts in the Ridiculous Interactive Proof is exponentially small.*

Proof. By the definition of MIPs, the provers cannot communicate. If Prover 2 can output an S' that is the same as the uniformly random S that only Prover 1 knows, then they must have communicated. Contradiction. □

The reader is astute in pointing out that steps 2 and 3 of the Ridiculous Interactive Proof clearly show that the verifier is helping the provers by relaying the very answer it is supposed to keep secret. The "proof" of the Ridiculous Theorem is flawed: it overlooks the verifier's cross-prover interactions. We will call this *contamination* by the verifier.

There exist MIPs in which this type of verifier behavior is necessary [Ben-Or et al. (1988)]. But then we find ourselves in a precarious situation: the verifier must contaminate for the MIP to achieve a certain property, but it must not contaminate too much as to not break another property. Existing

literature does not address this. Furthermore, the standard MIP model does not offer an easy way to do so.

The theme of this thesis is *contamination*. Accounting for it, enforcing its negation, and exploring the consequences of this enforcement are the problems which we will solve in this work. Specifically, we will answer the following questions:

- How do we account for contamination?
- How do we enforce non-contamination in theory?
- How do we enforce non-contamination physically?
- What are the consequences of enforcing non-contamination?
- Can the techniques we develop along the way be used in practice?

1.2 Our Contributions

In Chapter 3, we look at how the standard setup for multi-prover interactive proofs leaves an ambiguity about verifier contamination. We re-examine the well-known result $\mathbf{MIP} = \mathbf{ZKMIP} = \mathbf{NEXP}$ [Babai, Fortnow, and Lund (1992); Ben-Or et al. (1988); Dwork, Feige, Kilian, Naor, and Safra (1992); Feige and Kilian (1994, 2000); Fortnow, Rompel, and Sipser (1994); Kilian (1990b)] under a novel model which we call *locality-explicit*. We show that in this new model, the old results hold unambiguously. We show that as a consequence of enforcing non-contamination, we gain a new perspective on zero-knowledge: the simulator's *advantage*. This answers the questions of accounting for contamination, and enforcing non-contamination in theory and its consequences.

In Chapter 4, we implement MIPs under the assumption of no-signaling faster-than-light. Under this setup, commitments are expensive to initiate and maintain. We propose a new succinct protocol for $\mathbf{NP}$ (in terms of simultaneous commitments sustained) and show a variant that is resistant to entanglement. This protocol is presented in the locality-explicit form. This answers the question of enforcing non-contamination physically.

Finally, in Chapter 5, we explore the (hitherto unexplored) relationship between enforcing non-contamination, distributed trust, and the commitment protocol used in the locality-explicit model. We construct a new type of functional voter roster which is information-theoretically secure under the assumption of distributed trust. This answers the question of applying the techniques we have developed in the previous chapters.

Chapter 2

General Preliminaries

2.1 Probability and Entropy

Let S be a finite set. Let $X : S \to \mathbb{R}$ be a function. We call S the *sample space* and X a *discrete random variable*. We will only deal with discrete random variables in this work, so we will omit the word 'discrete' from now on.

Associated with each random variable X is a probability mass function, $f_X : S \to [0, 1]$ which defines the probability that X takes on a particular value. That is $f_X(x) = \mathbf{Pr}(X = x)$. This function satisfies

$$\sum_{x \in S} f_X(x) = 1.$$

The Shannon entropy H of a random variable describes the amount of uncertainty a random variable contains. We define it as

$$H(X) = -\sum_{x \in S} f_X(x) \log_2 f_X(x).$$

In this work, we will be looking at the case where $S = \{0, 1\}^n$, or the entropy of bit-strings. In the special case where $n = 1$, $H(X)$ represents the amount of uncertainly we have about the value of a binary random variable. When $H(X) = 0$, we know its value with certainty, whereas when

$H(X) = 1$, it could be zero or one with equal probability. For example, if we use X to describe the outcome of a coin-toss, then $H(X)$ describes the bias of the coin, from a coin which only lands on one side ($H(X) = 0$) to a perfectly fair coin ($H(X) = 1$) and everything in between.

2.2 Asymptotics

Definition 2.1. *Let $f, g : \mathbb{N} \to \mathbb{R}$ be functions. We define the following notations:*

- $f \in O(g)$ *if there exists* $k, C \in \mathbb{R}$ *such that* $x > k \Rightarrow |f(x)| < C|g(x)|$
- $f \in o(g)$ *if* $\lim_{x\to\infty} f(x)/g(x) = 0$
- $f \in \Omega(g)$ *if there exists* $k, C \in \mathbb{R}$ *such that* $x > k \Rightarrow f(x) > Cg(x)$
- $f \in \Theta(g)$ *if* $f = O(g)$ *and* $f = \Omega(g)$
- $f \sim g$ *if* $\lim_{x\to\infty} f(x)/g(x) = 1$
- f *is a negligible function if* $f(x) = o(1/\mathbf{poly}(x))$ *for every polynomial* $\mathbf{poly}$

The same asymptotic notations can be defined for functions whose domain or range is a subset of $\mathbb{R}$ analogously.

2.3 Interactive Turing Machines

Our main objects of study are cryptographic protocols. We will use interactive Turing machines to model these protocols. It is a consequence of the Church-Turing thesis [Church (1936); Turing (1937)] that any interactive protocol can be computed by a set of interactive Turing machines. We adopt the definition of interactive Turing machines from Goldreich (2006).

Definition 2.2. *An* interactive Turing machine *(ITM) is a Turing machine with the following tapes: (read only) input tape, (read only) random tape, work tape, (append only, at least one) sending communication tape(s), (read only, at least one) receiving communication tape(s), (append only) output tape, switch tape (consisting of a single cell).*

Each ITM is associated with a bit string $\mathbf{id}$*, called its* identity. *The machine is said to be* active *if the content of the switch tape is equal to its identity; otherwise, it is* idle.

We will most likely not ever consider a single ITM. We will configure them into networks in order to define cryptographic protocols.

Definition 2.3. *A set of ITMs is a* network *if their identities are all distinct, they are pairwise connected by sending and receiving communication tapes, and they share a single switch tape.*

The joint computation *of a network is a sequence of local configurations of all machines, on some common input* x*. In each step of the configuration, exactly one machine is active, all others are idle. The initial switch configuration is set to one of the machines.*

If a machine halts while the switch tape is equal to its identity, then we say that the network has halted. The outputs *of all machines on that network are determined at that point.*

The time complexity *of a network is a function* $t : \mathbb{N} \to \mathbb{N}$ *such that on common input* x*, all machines halt with* $t(|x|)$ *steps. If no such function exists, then the time complexity is unbounded.*

The definition from Goldreich (2006) included only a pair of ITMs. In order to study multi-prover interactive proofs, we use an extension of its definition here.

An execution *of ITMs* $M_1, \ldots, M_k$ *on common input* x*, denoted* $[M_1 \ldots M_k](x)$*, is the final states and outputs of all machines. When these machines are probabilistic,* $[M_1 \ldots M_k](x)$ *will denote the random variable, over their randomness, of their outputs and final states.*

If we only care about a particular machine's output or state, we will denote that machine's output and state as $[M_1 \ldots M_k](x)|_{M_i}$*. When the machine in question is clear from context, we will simply write* $[M_1 \ldots M_k](x)$*.*

Machine M_i accepts *the interactive computation on input* x *if it stops in state* $ACCEPT$ *in the execution* $[M_1 \ldots M_k](x)$*. The machines* jointly accept *if every machine accepts.*

2.4 Interactive Proofs

One particular kind of cryptographic protocol that will be the focus of our attention are *interactive proofs*. The simplest interactive proof can be defined by the interactions between a pair of ITMs. Chapter 3 deals with generalizations of the interactive proofs to multiple parties, so we will leave the general definition until that point.

The following definition for (two-party) interactive proofs comes from Goldreich (2006).

Definition 2.4. *A pair of interactive machines* (P, V) *is called an* interactive proof system for a language L *if machine* V *is polynomial-time and the following two conditions hold:*

(1) Completeness: *For every* $x \in L$, $\mathbf{Pr}([P, V](x) = ACCEPT) \geq 2/3$

(2) Soundness: *For every* $x \notin L$ *and every ITM* B, $\mathbf{Pr}([B, V](x) = ACCEPT) \leq 1/3$

Where $[A, V](x)$ *is* V*'s acceptance state in its execution with machine* A*.*

Multi-provers interactive proofs (MIPs) – also called *multi-prover interactive protocols* – are protocols involving a set of *provers* modeled by interactive Turing machines, each of them interacting with an interactive probabilistic polynomial-time (PPT) Turing machine called the verifier V. Although all provers may share an infinite read-only auxiliary input tape at the outset of their computation, they do not interact with each other.

Definition 2.5. *Let* $P_1, \ldots, P_k$ *be computationally unbounded interactive Turing machines and let* V *be an interactive PPT Turing machine. The* P_i*'s share a joint, infinitely long, read-only random tape. Each* P_i *interacts with* V *but cannot interact with* P_j *for any* $1 \leq j \neq i \leq k$*. We call* $[P_1, \ldots, P_k, V]$ *a* k-prover interactive protocol (k–prover IP).

A $[P_1, \ldots, P_k, V]$ k-prover interactive protocol is a *multi-prover interactive proof system for* L if it can be used to show V that a public input x is such that $x \in L$. At the end of its computation, V concludes $x \in L$ if and only if it ends up in state $ACCEPT$. We restrict our attention to interactive proof systems with perfect completeness since all our protocols have this property.

Definition 2.6. *The k–prover interactive protocol $\Pi = (P_1, \ldots, P_k, V)$ is said to be a* k-prover interactive proof system with perfect completeness for L *if there exists a negligible function $q(n)$ such that the following holds:*

(1) Completeness: $(\forall x \in L)\left[\mathbf{Pr}\big([P_1, \ldots, P_k, V](x) = ACCEPT\big) = 1\right]$,

(2) Soundness: $(\forall x \notin L)(\forall \widetilde{P}_1, \ldots, \widetilde{P}_k)\left[\mathbf{Pr}\big([\widetilde{P}_1, \ldots, \widetilde{P}_k, V](x) = ACCEPT\big) \leq q(|x|)\right]$.

The parameter $q(|x|)$ is called the soundness error *of* Π.

2.5 Single-Prover Zero-Knowledge

An interactive proof (of membership) is *zero-knowledge* if the verifier learns "nothing" except for the fact that "$x \in L$". We have adopted the formal definition of zero-knowledge from Goldreich (2006).

Definition 2.7. *Let (P, V) be an interactive proof for some language L. We say that (P, V) is* perfect zero-knowledge *if for every probabilistic polynomial-time interactive Turing machine V^* there exists a probabilistic polynomial-time algorithm M^* such that for all $x \in L$ and all auxiliary inputs $z \in \{0,1\}^*$.*

(1) $\mathbf{Pr}[M^*(x, z) = REJECT] \leq 1/2$.

(2) Assuming $M^*(x, z) \neq REJECT$, then $[P, V^*(z)](x)|_{V^*}$ and $M^*(x, z)$, as random variables, are identically distributed.

We define *statistical zero-knowledge* and *computational zero-knowledge* the same way, except that $[P, V^*(z)](x)|_{V^*}$ and $M^*(x, z)$ are statistically close and computationally indistinguishable, respectively.

2.6 Multi-Prover Zero-Knowledge

The zero-knowledge [Goldwasser et al. (1989)] version of MIPs were defined in Ben-Or et al. (1988). We give the textbook version of MIP zero-knowledge for completeness. We will discuss extensions to this definition in Chapter 3.

Definition 2.8. *Let $[P_1, \ldots, P_k, V]$ be a k-prover interactive proof system for L. We say that $[P_1, \ldots, P_k, V]$ is* perfect zero-knowledge *if for all PPT interactive Turing machines $\widetilde{V}$ there exists a PPT machine H^*, having blackbox access to $\widetilde{V}$, such that for all $x \in L$, and for all auxiliary inputs $z \in \{0,1\}^*$, $[P_1, \ldots, P_k, \widetilde{V}(z)](x)|_{\widetilde{V}}$ and $H^*(x, z)$ are identically distsributed.*

2.7 Circuits

We require circuits as an additional model of computation. The reason is that in zero-knowledge interactive proofs, we will, most of the time, be using circuits as a way of committing the steps and outputs of a computation (see, for example, Protocol 3.4).

A *circuit* is a finite simple directed acyclic graph where the edges (or *wires*) take on a value from some alphabet and for every vertex there corresponds a function which takes the values of the in-edges and outputs values for the out-edges. Vertices of in-degree zero are *input gates*. Vertices of out-degree zero are *output gates*.

Any particular circuit can only take inputs of a fixed length since there is a fixed number of input gates. Therefore, to represent a function whose domain might contain strings of (possibly infinitely many) dif and only iferent lengths, we use families of circuits $\{C_i\}_{i \in I}$, where each C_i is a circuit, and I is an index set.

A family of circuits $\{C_i\}_{i \in I}$ is *f-uniform* if there exists a Turing machine M such that for all $i \in I$, $M(i)$ produces an appropriately encoded description of C_i, taking less than $f(i)$ steps.

2.8 Complexity of Turing Machines and Circuits

Let M be a Turing machine, x be an input on which M halts. The *time complexity* of M on input x is the number of steps M takes before halting. The *space complexity* of M on input x is the maximum number of tape cells used before halting.

Very often, we will speak of the time and space complexities of a Turing machine as functions of the *lengths* n of the input, instead of the input itself. In such cases, we call the *worst case complexity* the maximum space/time used over all inputs of length n, whereas the *average case complexity* the average of the space/time used over all inputs of length n.

The complexity functions of most Turing machines are not easy to compute, so we bound them asymptotically with simpler functions.

Let C be a circuit. The *circuit size* of C is its number of vertices. The *circuit depth* is the length of its longest directed path.

We will adopt the usual convention in deeming Turing machines of polynomial complexity and polynomial-uniform circuits to be those that are *efficient* or *feasible*.

2.9 Complexity Classes

The following common complexity classes will be discussed in this work. They contain languages L with the following properties.

- $\mathbf{P}$ – L can be accepted in polynomial-time.
- $\mathbf{P/poly}$ – L can be accepted by a polynomially-sized family of circuits.
- $\mathbf{BPP}$ – L can be accepted in probabilistic polynomial-time with an error for soundness and completeness of at most 1/3.
- $\mathbf{NP}$ – Every $x \in L$ has a *witness* w_x such that $\{(x, w_x) : x \in L\} \in \mathbf{P}$, where $|w_x|$ is polynomial in $|x|$.

- co-$\mathbf{NP}$ – Every $x \notin L$ has a *witness* w_x such that $\{(x, w_x) : x \notin L\} \in \mathbf{P}$, where $|w_x|$ is polynomial in $|x|$.

- $\mathbf{PSPACE}$ – There exists a polynomial p and a Turing machine M such that, for all $x \in L$, M accepts x while not using more than $p(|x|)$ cells of M's tape.

- $\mathbf{EXP}$ – L can be accepted by a Turing machine in exponential-time.

- $\mathbf{NEXP}$ – Every $x \in L$ has a *witness* w_x such that $\{(x, w_x) : x \in L\} \in \mathbf{EXP}$, where $|w_x|$ is exponential in $|x|$.

Other complexity classes will be defined if the need arises.

2.10 Commitment Schemes

Commitment schemes are important cryptographic primitives widely used in theory and practice. Here we give a formal definition of a commitment scheme for a single bit from two provers to one verifier. For generalization to multiple parties, see Goldreich (2006).

Definition 2.9. *A* bit commitment scheme *is a cryptographic protocol between a sender S and a receiver R. The sender has a secret bit b which the receiver does not know. The protocol has two phases, the* commit *phase, and the* unveil *phase. There is a unary security parameter λ. Let* $\mathbf{negl}$ *be a negligible function. The protocol is subject to the following constraints:*

(1) Secrecy/Hiding/Concealing: At the end of the commit phase with an honest sender, the probability that any polynomial-time receiver outputs b is $< 1/2 + \mathbf{negl}(\lambda)$.

(2) Unambiguity/Binding: Given the transcript of the commit phase and a random bit b, the probability that any polynomial-time sender can complete the unveil phase and have the receiver output b is $< 1/2 + \mathbf{negl}(\lambda)$.

(3) Correct/Complete/Viable: If both the sender and receiver are honest, then at the end of the unveil phase, the receiver outputs b.

A commitment scheme with the above properties would be one with *computational hiding, computational binding* security. It is easy to see how stronger notions of hiding and binding can be defined.

We will encounter a special type of commitment scheme, known colloquially as the BGKW-, CHSH-, or PR-type commitment scheme. It is between two senders (provers) and one receiver (verifier). We define a variant here.

Definition 2.10. *Statistically binding, perfectly concealing 2-prover bit-commitment protocol.*

All parties agree on a security parameter k. P_1 and P_2 partition some of their private random tape into $k+1$-bit strings $\{(c_i, w_i)\}_{i \leq \mathbb{N}}$, where c_i are bits and $|w_i| = k$.

Pre-computation phase:

- *V chooses a k-bit string z uniformly at random and sends it to P_2.*
- *P_2 responds with $d_i = w_i \oplus c_i \cdot z$, for $1 \leq i \leq N$, where N is sufficiently large (depending on the protocol which uses this bit-commitment scheme as a sub-protocol), and $c_i \cdot z$ are thought of as the product between a scalar c_i and a vector z, over $\mathbb{Z}_2$.*

Commit phase:

- *P_1 wishes to commit b_i to V as $[b_i] = b_i \oplus c_i$.*

Unveil phase:

- *P_1 sends w_i to V.*
- *V computes $c_i = 1$ if $d_i \oplus w_i = z$, or $c_i = 0$ if $d_i \oplus w_i = \vec{0}$ and recovers $b_i = [b_i] \oplus c_i$. V rejects if $d_i \oplus w_i$ does not equal to either z or $\vec{0}$.* □

Commitment schemes such as Pedersen's scheme [Pedersen (1992)] that are computationally binding are *verifiable* in the sense that the receiver can show the transcript of the entire protocol to a third party and have them accept the outcome of the protocol. The CHSH-type commitment above

does not have this property, unless the senders sign their messages, or unless the third party trusts the receiver and a majority of senders (if one generalizes the CHSH-type commitment to more than one sender).

2.11 Non-Locality, No-Signaling, and Quantum Non-Local

Although the standard MIP model simply assumes that the provers do not communicate, subsequent work [Cleve, Hoyer, Toner, and Watrous (2004)] augmented the provers with so-called *non-local resources*. An example of this would be quantum entanglement which, by itself, does not allow communication between entangled parties [Popescu and Rohrlich (1998)]. Formally, this will be defined as probability distributions on the inputs and outputs of bipartite games. We focus on single-round games and strategies as they are sufficient to analyze most MIPs.

Definition 2.11. *Let V be a predicate on $A \times B \times X \times Y$ (for some finite sets A, B, X, and Y) and let π be a probability distribution on $A \times B$. Then $(A \times B \times X \times Y, V, \pi)$ is called a* single-round game.

Given two players, Alice and Bob, and a pair of questions (a, b) sampled according to π, where $a \in A$ is sent to Alice and $b \in B$ is sent to Bob. Suppose that Alice responds with $x \in X$ and Bob with $y \in Y$. Then we say that Alice and Bob win *if $V(a, b, x, y) = 1$ and* lose *otherwise.*

Definition 2.12. *A* strategy *for Alice and Bob is a probability distribution $P_{(x,y|a,b)}$ describing their answer (x, y) on pairs of questions (a, b). The* winning probability *of a strategy is the probability of the strategy producing a correct answer, given a uniformly random question (a, b). A strategy is* winning *if its winning probability is 1.*

The value *of a game is the supremum over the winning probabilities of all strategies.*

Definition 2.13. *A strategy $P_{(x,y|a,b)}$ is* local *if there exists a finite set R and functions $f_A : A \times R \to X$, $f_B : B \times R \to Y$ such that*

$$P_{(x,y|a,b)} = \frac{|\{r \in R : x = f_A(a, r) \text{ and } y = f_B(b, r)|}{|R|}.$$

A local strategy corresponds to the situation where Alice and Bob agree on an individual deterministic strategy selected uniformly among $|R|$ such possibilities. The choice r of Alice and Bob's strategy, and the choice of inputs (a, b) provided to Alice and Bob are generally agreed to be statistically independent random variables.

Definition 2.14. *We define the class* local hidden variable $(\mathbb{LHV})$ *to be the set of local strategies for any two-party game.*

Of particular interest are games where there are winning or high-probability strategies which are not local, but still do not violate the no-communication assumption of MIPs:

Definition 2.15. *A strategy* $P_{(x,y|a,b)}$ *is* no-signaling *if the marginal distributions satisfy* $P_{(x|a,b)} = P_{(x|a)}$ *and* $P_{(y|a,b)} = P_{(y|b)}$.

In this work, we will think of non-local resources for the provers as an honest third party (a *correlator*, formally defined later) interacting with the provers (and another with the verifiers). We can use it to quantify the amount or strength of the non-local resources available to the provers. When the behavior of this third party is well defined as a game strategy, we can associate it with an existing non-local resource in the literature (such as quantum entanglement, see Barrett et al. (2005)).

Of some interest are a class of non-local, no-signaling strategies called *quantum non-local* ($\mathbb{QNL}$), which represents strategies players can adopt when they share quantum entanglement (but still not allowed to communicate in any way). We need a bit of background in quantum information before we can properly define it. We adopt the definitions from Nielsen and Chuang (2010).

Definition 2.16. *Associated to every quantum system is a* state space*, which is a complex inner product space (Also known as a Hilbert space). A* state vector *is a unit-length vector in a state space. The state vector completely describes the quantum system.*

A qubit *is the state vector of a two-dimensional state space.*

We will adopt Dirac's Bra-Ket notation and write qubits as

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

where the qubit is labeled ψ, $|0\rangle$ and $|1\rangle$ are a pair of orthonormal basis for the state space, and $a, b \in \mathbb{C}$. We will denote the conjugate transpose of a qubit $|\psi\rangle$ as $\langle\psi|$.

Evolutions of quantum systems are *unitary operations* from the state space to itself. We will interpret them as analogs of logical operations in classical computer science.

In order to extract (classical) information out of a quantum system, we must perform a *measurement*.

Definition 2.17. *A* quantum measurement *is a collection $\{M_m\}$ of operators acting on the state space being measured. The index m refers to the outcome of the measurement. Suppose that $|\psi\rangle$ is the state of the quantum system being measured, then the following holds:*

- *The probability that outcome m occurs, $p(m)$, is given by $p(m) = \langle\psi|\, M_m^\dagger M_m\, |\psi\rangle$,*
- *Given outcome m, the system post-measurement becomes $M_m\,|\psi\rangle\,/\sqrt{p(m)}$, and*
- *The measurement operators satisfy $\sum_m M_m^\dagger M_m = I$, where I is the identity.*

Composite quantum systems are constructed using *tensor products*, denoted $\otimes$. We refer the reader to Page 71 of Nielsen and Chuang (2010) for the definition and algebraic properties of tensor products.

Definition 2.18. *The state space of a composite system is the tensor product of its component systems. If each of n components is in state $|\psi_i\rangle$, then the composite system is in state $|\psi_1\rangle \otimes \ldots \otimes |\psi_n\rangle$. We will abbreviate this last tensor with the notation $|\psi_1\rangle \ldots |\psi_n\rangle$ or $|\psi_1 \ldots \psi_n\rangle$. In general, the state vector of a composite system may not always be the tensor product of smaller state vectors.*

The phenomenon which forms the basis of much (but not all) of what we call non-local resources is *quantum entanglement*. It is those states which cannot be factored into a tensor product of smaller, composite states.

Definition 2.19. *Given a state space, a state vector* $|\psi\rangle$ *is* entangled *if there does not exist state vectors* $|a\rangle, |b\rangle$ *such that* $|\psi\rangle = |a\rangle |b\rangle$.

What will be of interest to us are state spaces which are the tensor products of two state spaces – each of which belongs to one of two non-communicating parties – wherein the state vector is entangled. Entanglement theory is an active area of research. What is presented here is the barebones framework needed to define quantum non-local. We invite the reader to explore quantum information and entanglement in its full generality in works such as Nielsen and Chuang (2010) and Wilde (2013).

Definition 2.20. *If a state space* $H = H_1 \otimes H_2$*, we can assign* H_1 *and* H_2 *to two parties (or, in the context of games, two players). And if* $U = U_1 \otimes U_2$ *is a unitary operator on* H*, then* U_1 *and* U_2 *are* local operations *which the two parties can apply on their share of the state. The parties can perform* local measurements $\{M_m \otimes \mu_n\}$ *on their respective states. These measurements are a special case of the general measurements, defined above.*

Definition 2.21. *A strategy is* entanglement-assisted *if it is output from players who share entangled states.*

We define the class quantum non-local $(\mathbb{QNL})$ *to be the set of entanglement-assisted strategies for any two-party game.*

Chapter 3

Contamination in Interactive Proofs

3.1 Introduction

From a *complexity* perspective, the zero-knowledge[1] aspect of interactive proofs is characterized by $\mathbf{IP} = \mathbf{CZKIP} = \mathbf{PSPACE}$ for single-prover IPs [Ben-Or et al. (1990); Impagliazzo and Yung (1988); Shamir (1992)], and $\mathbf{MIP} = \mathbf{ZKMIP} = \mathbf{NEXP}$ for multi-prover IPs [Babai et al. (1992); Ben-Or et al. (1988); Dwork et al. (1992); Feige and Kilian (1994, 2000); Fortnow et al. (1994); Kilian (1990b)]. The (conjectured) necessity of complexity assumptions for zero-knowledge in the single-prover case was the initial motivation for the multi-prover model.

Zero-knowledge is also where contamination becomes a proper problem for the standard MIP model. The most important (and the most subtle) of those contaminations are ones where the verifier helps the provers perform a so-called *no-signaling* correlation; examples of this can be found in the following section, and also in Crépeau, Salvail, Simard, and Tapp (2011).

In the Ridiculous Interactive Proof, the Ridiculous Verifier is clearly contaminating. This is not obvious when the verifier is more complex. It is an even subtler point when we consider that the

This Chapter is based on a paper written by Claude Crépeau and Nan Yang entitled *Non-Locality and Zero-Knowledge MIPs*. It is currently under submission. A preliminary version of the paper was accepted as an invited paper published in the post-proceedings of Mycrypt 2016 [Crépeau and Yang (2016)].

[1] Computational zero-knowledge. We omit statistical zero-knowledge from this introduction.

verifier could be helping the provers in a no-signaling manner. We believe that proofs within the standard model must be reconsidered in light of this observation. We will further discuss this last point in section 3.3.

In the existing MIP literature, proofs of soundness do not account for this blind spot. If soundness depends specifically that the provers are correlated in a certain way, then contamination can be problematic. At the very least, any proof which does not address soundness would be incomplete. Proofs may implicitly assume non-contamination, but we would like to make this explicit.

Our solution, discussed in this chapter, is a multi-prover, multi-verifier model. We will borrow the term *locality* from physics and call our model *locality-explicit multi-prover interactive proofs* (LE-MIPs). LE-MIPs consist of prover-verifier pairs who are talking, but no communication *between* any of the pairs. At the end of a locality-explicit protocol, a special, read-only verifier accepts or rejects. LE-MIPs, by design, account for contamination in their very specification. This makes it easy to prove that the desired security properties are unaffected by contamination.

In this chapter, we answer the following questions which were raised in the introduction:

- How do we account for contamination?
- How do we enforce non-contamination in theory?

3.2 Previous Work

The early claims by Ben-Or, Goldwasser, Kilian and Wigderson that $\mathbf{ZKMIP} = \mathbf{MIP}$ from Ben-Or et al. (1988) and Kilian (1990b) use multi-round protocols and their (honest) verifiers are inherently signaling. This is precisely the situation we address in this work. Proving soundness is quite subtle in this case because the provers could use the (signaling) verifier to break binding of the commitments. In particular, soundness will not be valid if the protocol is composed concurrently with other executions of itself or even used as a sub-routine. In recent conversations with Kilian [Kilian (2018)], we have realized that controlling the impact of *signaling* via the verifier has been a concern since the early days of MIPs. In particular, extra care had to be taken in the zero-knowledge

protocols described in Ben-Or et al. (1988) and Kilian (1990b) because the verifier couriered messages from one prover to the other. The protocols as they are might be sound but it is not fully proven. However, it is also clear that no considerations had been given to general contaminations made possible via the verifier. If soundness rests on the binding property of a commitment scheme (such as those zero-knowledge proofs) and this binding property rests on the inability to achieve a certain non-local correlation then impossibility to achieve this correlation via the verifier must be demonstrated.

The reader may think that the entire issue we address may seem trivial because it is a known fact that multi-round MIPs may be reduced to a single round using techniques of Lapidot-Shamir [Lapidot and Shamir (1991)] and Feige-Lovasz [Feige and Lovász (1992)]. Nevertheless, if we are interested in *zero-knowledge* MIPs, commitment schemes are generally used to obtain the zero-knowledge property and thus the single-round structure is lost in the process. Although single-round protocols – if implemented properly – bypass verifier's contamination problems we describe in this work, converting multi-round protocols into single-round ones is highly inefficient and complex. Preserving zero-knowledge while achieving single-round has turned out to be a major challenge. Practically, keeping a multi-round protocol's structure, using only commitments to achieve zero-knowledge is very appealing.

In Lapidot and Shamir (1991), the authors proposed a parallel ZKMIP for $\mathbf{NEXP}$, but they removed the zero-knowledge claim in the journal version Lapidot and Shamir (1997) of their work without any explanation as to why. Feige and Kilian (1994) were the last ones to follow this approach combining techniques drawn from Lapidot and Shamir (1991), Feige and Lovász (1992) and Dwork, Feige, Kilian, Naor, and Safra, [Dwork et al. (1992)] to achieve a "2-prover 1-round 0-knowledge" proof for $\mathbf{NEXP}$.

As far as we can tell, this is the only paper in the ZKMIP literature that appears to address the problems that we will discuss. However, note that the analysis of Feige and Kilian (1994) is partly based of that of Lapidot and Shamir (1991), and the journal version of Feige and Kilian (2000) does not contain their prior claim of zero-knowledge either. All other ZKMIPs for $\mathbf{NEXP}$ in the

literature are multi-round, and thus our work applies to them.

Similar issues are possible using more recent results such as the proof from Ito and Vidick (2012) that $\mathbf{NEXP} \subseteq \mathbf{MIP}^*$ and the proof from Kalai, Raz, and Rothblum (2014) that $\mathbf{MIP}^{ns} = \mathbf{EXP}$; the multi-round structure of their protocols requires that any straightforward extensions to $\mathbf{ZKMIP}^*$ and $\mathbf{ZKMIP}^{ns}$ via commitment schemes be analyzed carefully and the locality of the verifiers be established.

Bellare, Feige, and Kilian (1995) considered a multi-verifier model similar to ours in order to analyze the role of randomness in multi-prover proofs. This is completely unrelated to our goal of analyzing verifier contamination.

Finally, the notion of relativistic commitment schemes (using multiple provers and verifiers) put forward by Kent (1999) leads to several results [Adlam and Kent (2015); Chailloux and Leverrier (2017); Lunghi et al. (2015)] where a similar multi-verifier model is necessary in order to assess spatial separation of the provers.

3.3 The Standard MIP Model

Multi-prover interactive proofs were introduced in Ben-Or et al. (1988). The intuition for their model was that of a detective interrogating two suspects held in different rooms. This was formalized as follows:

Definition 3.1. *Let $P_1, \ldots, P_k$ be computationally unbounded Turing machines and let V be a probabilistic polynomial-time Turing machine. All machines have a read-only input tape, a read-only auxiliary-input tape, a private work tape and a random tape. The P_i's share a joint, infinitely long, read-only random tape. Each P_i has a write-only communication tape to V, and vice-versa. We call $(P_1, \ldots, P_k, V)$ a* k-prover interactive protocol (k-prover IP).

This model is essentially equivalent to that of J. S. Bell (1964) who introduced his famous Bell's inequality to distinguish *local* parties from *entangled* parties.

Zero-knowledge MIPs were also defined in Ben-Or et al. (1988):

Definition 3.2. *Let* $(P_1, \ldots, P_k, V)$ *be a k-prover IP for a language* L*. Let* $\mathbf{view}(P_1, \ldots, P_k, V, x)$ *denote the verifier's incoming and outgoing messages with the provers, including his coin tosses. We say that* $(P_1, \ldots, P_k, V)$ *is* perfect zero-knowledge *for* L *if there exists an expected polynomial-time machine* M *such that for all* V'*,* $\mathbf{view}(P_1, \ldots, P_k, V', x)$ *and* $M(x)$ *are identically distributed.*

Let us call the above two definitions the *standard MIP model*. There have also been augmentations of the model by giving the provers various non-local resources, such as entanglement [Ito and Vidick (2012)], or arbitrary no-signaling power [Kalai et al. (2014)].

The first work to point out the aforementioned contamination problem in the standard MIP model, though implicitly, was Crépeau et al. (2011). In order to understand their point, we need to understand the following two-prover protocol.

Protocol 3.1. *(BGKW-type commitment for bit* b *)*

P_1 and P_2 pre-share a random n-bit string w. Let b be a bit. Let x, w, r, w' be from some finite field.

(1) V sends a random n-bit strings r to P_2.

(2) P_2 replies with $x \leftarrow b \times r \oplus w$.

(3) P_1 announces to V a string w'.

(4) V accepts iff $(w' \oplus x) \in \{0, r\}$.

This is a two-prover commitment protocol. Steps 1 and 2 commit, while steps 3 and 4 unveil. An intuitive proof of its binding condition is that, since the provers cannot signal, and they both need to know r in order to unveil the commitment in the way they want, therefore they cannot cheat. This

intuition is incomplete, as was pointed out in Crépeau et al. (2011), because breaking the binding condition *does not require signaling*. The following protocol, known as a PR-box, can be used to break binding without signaling.

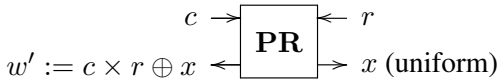


Figure 3.1: A **PR**-box.

By having P_1, P_2 obtain w', x via the PR-box, P_1 can unveil the commitment the way it wishes, c. This fact will become extremely important in Sections 3.5 and 3.4.1.

The punchline of Crépeau et al. (2011) is that *the verifier itself can act as a PR-box for the provers without violating their no-signaling assumption*. Consider the following:

(1) Any security proof of protocol 3.1 must show that it does not contain a PR-box as a subroutine.

(2) More generally, any security proof of a protocol must show that no subroutine within itself can be commandeered by the provers to achieve a non-local functionally (like the PR-box).

(3) Composition of protocols, for instance between the committing and the opening of commitments, must be done in such a way that provably does not create a non-local box.

The solution proposed in Crépeau et al. (2011) was that of *verifier isolation*. Informally, this means that any message an "isolating" verifier sends to a set S of provers must be computed solely from messages that are received from S. The end result is that an isolating verifier can never accidentally implement a PR-box and, in general, it will always enforce the locality of the provers. In a sense, we can think of an isolating verifier as "local". Our new model will make this more precise and more general.

Furthermore, existing zero-knowledge MIPs such as Kilian (1990b) *require* that the verifier courier an authenticated message between the provers in order to obtain soundness while ensuring zero-knowledge. The gist of it goes like this:

(1) V asks P_1 some questions.

(2) V wants to check one of P_1's answers with P_2 for consistency.

(3) In order for zero-knowledge to hold, V *must* ask P_2 a question it has already asked P_1.

(4) P_1 authenticates a question with a key that was committed at the beginning of the protocol and sends it to V.

(5) V sends the question and the authentication to P_2, who proceeds only if authentication succeeds.

Steps 4 and 5 consists of V sending a message from P_1 to P_2. Proofs that this act does not contaminate non-locally (such as simulating a PR-box) is not found in any existing MIP. This needs to be proven, and the proof contained in Kilian (1990b) does not address this issue. Moreover, the zero-knowledge protocol of Kilian (1990b) allows P_1 to send an arbitrary message to P_2 (via the authentication tag). Therefore, one cannot compose such a protocol in a nested fashion (as a subroutine call) since the inner instance would violate the no-communication assumption of the outer instance. For more details on the problems of the standard **MIP** model, see Crepeau and Yang (2017).

Existing simulators for zero-knowledge protocols such as those found in Kilian (1990b) needs to know how to break commitments in order to simulate. The simulator accomplishes this by acting as both provers, thereby receiving the secret string r which was meant for one prover only. This standard model of zero-knowledge gives the simulator *unnecessary power*, in a sense. We will discuss this further in section 3.4.1.

3.4 Locality-Explicit MIP

The standard MIP model allows the verifier to non-locally contaminate the provers. We neutralize this problem by defining a model with multiple verifiers, each of which talks to a single prover; in turn, each prover talks to a single verifier. There are no communication tapes between the verifiers, nor are there between provers. There is a special verifier V_0 which *only reads* the outputs of the other verifiers; this is the verifier that will decide to accept or reject membership to L. We call this model

"locality-explicit" since the provers and verifiers are explicitly local, and if any non-local resources (such as entanglement) are available to them, then it is explicitly specified via a supplementary entity named $\widehat{P}$ for the provers and $\widehat{V}$ for the verifiers.

This model is a *generalization* of the standard model because the special setting where $\widehat{P}$ is empty and $\widehat{V}$ signals for the verifiers corresponds to the standard MIP model.

Definition 3.3. *Let $(\widehat{P}, P_1, \ldots, P_k, \widehat{V}, V_0, V_1, \ldots, V_k)$ be a tuple of ITMs, where the P_i's are computationally all-powerful and the V_i's are polynomial-time. For each i, there are two-way communication tapes between V_i and P_i, and that for all j, there is a two-way communication tape between $\widehat{V}$ and V_j and also between $\widehat{P}$ and P_j. In addition, for each ℓ, there is a read-only tape going from V_ℓ to V_0 (where V_0 reads). Then, this is said to be a* locality-explicit multi-prover interactive proof.

We call $\widehat{P}$ and $\widehat{V}$ correlators *and say that the provers and verifiers are $\widehat{P}$-*local *and $\widehat{V}$-*local *respectively.*

It is perhaps easier to understand our definition with the help of figure 2.

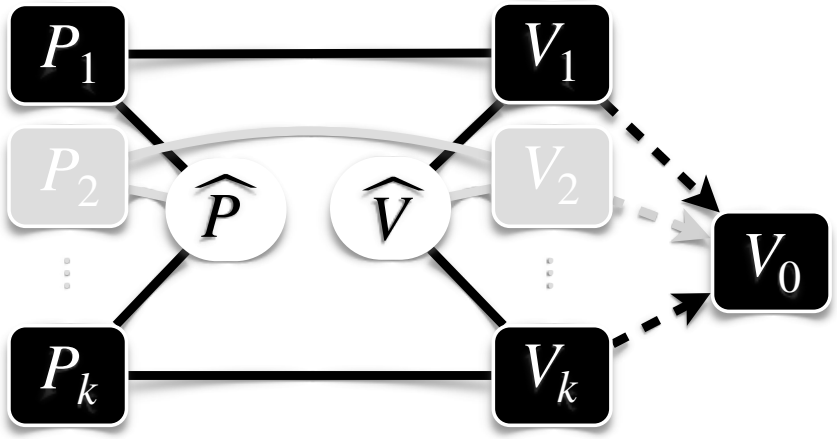


Figure 3.2: Locality-Explicit MIP.

The solid lines represents two-way communication and the dashed arrows represents one-way communication, with the arrow indicating the direction of information flow.

We can define that an LE-MIP accepts a language L if the usual soundness and completeness conditions hold:

Definition 3.4. *An LE-MIP* $(\widehat{V}, V_0, V_1, \ldots, V_k, \widehat{P}, P_1, \ldots, P_k)$ *accepts a language* L *if and only if*

- *(completeness)* $\forall x \in L, \mathbf{Pr}[V_0(x, t_1, \ldots, t_k) = \mathsf{accept}] > 2/3$,
- *(soundness)* $\forall x \notin L, \forall P'_1, \ldots, P'_k, \mathbf{Pr}[V_0(x, t_1, \ldots, t_k) = \mathsf{accept}] < 1/3$,

where t_i *is the read-only tape from* V_i *to* V_0 *at the end of the interaction of* V_i *with* P_i *(or* P'_i*) on input* x*.*

We will denote by $\mathbf{MIP}^B_A$ the set of languages accepted by an LE-MIP where the verifiers share the correlator A, and the provers share the correlator B.

Note that we do not quantify over $\widehat{P}$ (nor $\widehat{V}$), as we want to use them not as (possibly malicious) participants to the protocol, but as a description of non-local resources available to the provers and verifiers.

Definition 3.5. *An LE-MIP is* local *if* $\widehat{V} = \widehat{P} = \emptyset$ *and all of the provers' (resp. verifiers') random tapes are initialized with the same uniformly random string* R *(resp. verifiers with another, independent uniformly random string* S*)*[2].

Note that (single-verifier) standard MIPs in which provers do not have non-local resources are equivalent to LE-MIPs where $\widehat{P} = \emptyset$ and $\widehat{V}$ acts as a bulletin board. That is, a single verifier communicating with multiple provers is equivalent to multiple verifiers communicating with provers and among each other.

In standard MIPs, it is possible that the honest (single) verifier bridges the provers non-locally. If a protocol does not desire this – and most existing MIPs do not – it must be proven. With local LE-MIPs, the special verifier V_0 decides to accept or reject. This verifier cannot communicate with anyone else, avoiding the aforementioned contamination.

[2] By $\emptyset$ we mean the empty correlator that provides everyone with nothing at all as output.

3.4.1 Zero-Knowledge LE-MIPs

Zero-knowledge is defined by simulations, the fundamental idea that if a transcript can be produced by an entity (simulator) with no more power than one (verifier) interrogating all-powerful provers, then no knowledge is gained.

The simulator of single-prover IP and standard MIP are equal to the verifier in computational power, but they do have "advantages" which allow them to fake transcripts. For single-prover IPs, the simulator is allowed to rewind computation; for standard MIPs, the simulator is given a (commitment-breaking) secret (though rewinding is allowed, it is unnecessary). Those advantages are, of course, independent of knowledge.

LE-MIPs naturally induces a new advantage for the simulator: non-local correlations. This is a very powerful advantage. Using the correct non-local correlations, simulators do not need to rewind, do not need to pretend to be multiple (isolated) provers, and do not need to know any commitment-breaking secrets. Multiple, no-signaling simulators can even produce transcripts in "real-time" (example will follow) if the proper correlations are used.

Definition 3.6. *Let $\mathcal{M} = (\widehat{M}, M_1, \ldots, M_k)$ be a tuple of polynomial-time ITMs. Each machine has a random tape, and every random tape is initialized with the same random bits. For $1 \leq i \leq k$, there is a two-way communication tape between $\widehat{M}$ and M_i. There are no communication tapes between any of the M_i's. Then this is called a tuple of* locality-explicit simulators *and $\widehat{M}$ is the* locality class *of $\mathcal{M}$, which will be abbreviated $\widehat{M}$-*local*.*

Definition 3.7. *Let $\mathcal{S} = (\widehat{P}, P_1, \ldots, P_k, \widehat{V}, V_0, V_1, \ldots, V_k)$ be an LE-MIP for language L. If there exists a correlator $\widehat{M}$ such that for all verifiers $(\widehat{V}', V_0', V_1', \ldots, V_k')$ and for all $x \in L$, there exists $(M_1, \ldots, M_k)$, such that the views between*

$$(\widehat{P}, P_1, \ldots, P_k, \widehat{V}', V_0', V_1', \ldots, V_k')$$

and

$$(\widehat{M}, M_1, \ldots, M_k, \widehat{V}', V_0', V_1', \ldots, V_k')$$

are identically distributed (on input x), where $(\widehat{M}, M_1, \ldots, M_k)$ is a tuple of locality-explicit simulators, then we say that $\mathcal{S}$ is a perfectly indistinguishable, $\widehat{M}$-local zero-knowledge LE-MIP *for L.*

We will denote by $\mathbf{ZK}^S\mathbf{MIP}^B_A$ a zero-knowledge LE-MIP where the simulators share correlator S, and verifiers/provers share A/B as before.

Our motivations for the above definitions are twofold.

First, a simulator (or simulators) should not have more power than necessary. If two *local* simulators can output for two *local* verifiers, then it is not necessary to have a single simulator (equivalent to two *signaling* simulators) do the job. Allowing simulators to signal (equivalently, having a single simulator) in the multi-prover setting is analogous to allowing unbounded running-time simulation in single-prover zero-knowledge. In general, finding the minimal $\widehat{M}$ that will allow simulation may be of some theoretic interest.

Second, the non-locality of simulators is a characterization of the resilience of zero-knowledge. A protocol with local simulators can withstand arbitrary (malicious) verifiers is more resilient than one where signaling simulators are needed.

This may be of practical interest, if transcripts are timestamped. For example, under the relativistic assumption that one may not signal faster-than-light, one may be able to distinguish two spatially separated simulators from two spatially separated verifiers, if the simulators need to signal (transmit a commitment-breaking secret) in order to generate a transcript. On the other hand, if two entangled simulators are sufficient to produce the transcript, then they are indistinguishable from real verifiers and provers. Our protocol 3.6 can be modified as to let entangled simulators do their work, without needing PR-boxes or signaling. Details in section 3.5.

3.4.2 The Power of LE-MIPs

Local LE-MIPs form a subclass of standard MIPs. They are, by design, more restricted in what you can make the verifier do. An immediate question is whether this is *too* restrictive. Perhaps, in all

interesting cases, it is necessary for a single verifier to go back-and-fourth between provers, using previous discussions to generate new questions.

The answer is that, of all the literature we have surveyed, almost all protocols can be re-written in a local-verifier manner without any loss of functionality. We explicitly demonstrate this for the multi-prover protocol for oracle-3-SAT in Babai et al. (1992). For the purpose of our discussion, we only need to look at the general form of the protocol:

Protocol 3.2. *(BFL Classic, Single-Verifier)*

(1) V asks P_1 some questions non-adaptively.

(2) V chooses a question Q from the pool of questions which were asked to P_1.

(3) V asks Q to P_2.

(4) V accepts if the interaction with P_1 was successful, and the answer from P_2 is consistent with those of P_1.

The crucial observation is that V does not *adaptively* ask questions to P_1. Therefore, the questions asked on that entire side of the conversation can be selected in advance, and thus they can be shared in advance with a second verifier. We can therefore naturally rewrite the BFL classic protocol as a local LE-MIP in the following way. The reader can check the details in section 3 of Babai et al. (1992).

Protocol 3.3. *(BFL as an LE-MIP)*

(1) V_1 prepares the questions which it will ask P_1.

(2) V_1 chooses a question Q from the above list and shares it with V_2.

(3) LE-MIP begins. All parties are local as per definitions.

(4) V_1 asks the questions to P_1.

(5) V_2 asks Q to P_2.

(6) V_0, reading the responses, decides to accept or reject, based on the same criteria as in protocol 3.2.

The BFL protocol is for oracle-3-SAT, which is $\mathbf{NEXP}$-complete. Rewritten as a local LE-MIP, it circumvents all non-locality issues we have mentioned. Thus, we can conclusively say that "local $\mathbf{LE\text{-}MIP}$" $= \mathbf{MIP} = \mathbf{NEXP}$; no transformation to single-round MIP necessary, and no need to invoke the general theory of PCPs.

3.5 A Local, Zero-Knowledge LE-MIP for NEXP

The question which follows naturally is whether there exists a *zero-knowledge*, local LE-MIP for $\mathbf{NEXP}$. The existing technique for achieving zero-knowledge in MIP [Ben-Or et al. (1988); Kilian (1990b)] requires the (single) verifier to courier an authenticated message between provers. This is not possible with local-verifier LE-MIPs. We show that there is a way around that constraint.

By adapting the protocol from Babai et al. (1992), we will exhibit a protocol with the following properties:

(1) The provers and verifiers are local: $\widehat{V}, \widehat{P} \in \mathbb{LHV}$. For simplicity, we will assume that $\widehat{V} = \widehat{P} = \emptyset$.

(2) The simulators need only access to instances of PR-boxes to work. That is, $\widehat{M}$ simply computes indexed instances of PR-boxes. We will abbreviate this as "PR-local."

Let us call the set of multi-prover protocols with these properties "PR-local **ZK**, local **LE-MIP**" which we shall denote $\mathbf{ZK}^{PR}\mathbf{MIP}^{\emptyset}_{\emptyset}$. This implies that $\mathbf{ZK}^{PR}\mathbf{MIP}^{\emptyset}_{\emptyset} = \mathbf{ZKMIP} = \mathbf{NEXP}$.

The generic way of turning an interactive proof into a zero-knowledge one is by running it in committed form [Ben-Or et al. (1988); Kilian (1990b)]. With this technique, provers commit their answers instead of directly responding, and use cryptographic techniques to convince the verifier that the answers are correct.

As shown in section 3.4.2, the BFL protocol can be turned into a local LE-MIP. If we try to turn it into a zero-knowledge LE-MIP by having the provers commit their answers (for example using protocol 3.1 as commitment), we run into a problem. In order to achieve zero-knowledge, the provers *must* ensure that the question P_2 receives from V_2 is one of the questions which V_1 has asked P_1. On the other hand, since the provers and verifiers are local, the provers cannot communicate, nor can they ask the verifiers to courier authenticated messages between them.

Our solution essentially asks the provers to (strongly-universal-2) hash [Carter and Wegman (1979)] the selected committed answer with a key that is based on the verifier's question. We force V_2 to behave honestly (to ask a question that V_1 has asked) by making bad questions meaningless. If the verifiers ask the provers the same question, they will receive the same hash of the same answer. Otherwise, they will receive two unrelated random hash values.

We need the PR commitment (protocol 3.4), which is secure in the local setting as previously proved in Crépeau et al. (2011); Kent (1999); Lunghi et al. (2015).

3.5.1 The Protocols

The following is a PR-type commitment that is perfectly concealing and statistically binding. In general, we use the commitment-box notation "$\boxed{b}$" as the name of a commitment to bit b in the next two protocols.

Protocol 3.4. *A statistically binding, perfectly hiding commitment protocol to bit b.*

All parties agree on a security parameter 1^k.

P_1 and P_2 partition their private random tape into two k-bit strings w_1, w_2.

Pre-computation phase:

- V_1 samples two k-bit strings z_1, z_2 independently and uniformly, and provides them to V_2.
- V_1 sends z_1 to P_1 and V_2 sends z_2 to P_2.

Commit phase:

- P_1 commits b to V_1 as $\boxed{b} = (b \times z_1) \oplus w_1$, where $b \times z_1$ is a multiplication in $\mathbb{F}_{2^k}$.
- P_2 sends V_2: $d = (w_1 \times z_2) \oplus w_2$.

Unveiling phase:

- P_1 sends w_1, w_2 to V_1.
- V_1 computes $b = 1$ if $\boxed{b} \oplus w_1 = z_1$, or $b = 0$ if $\boxed{b} = w_1$.
- V_0 **rejects** if $\boxed{b} \oplus w_1$ is anything but z_1 or 0, or if $d \oplus w_2 \neq w_1 \times z_2$ and **accepts** b otherwise.

A proof sketch of the hiding and binding properties of the above commitment scheme is as follows:

- *Hiding* – Without any information on w_1, which is sampled uniformly at random, w_1 and $z_1 \oplus w_1$ are indistinguishable by the verifiers.
- *Binding* – We assume that the provers are local. Therefore, we can rewind the provers. Thus, if the provers can break binding, it must be able to unveil the commitment to both a 0 and

a 1 (which is not true for non-local provers). Specifically, P_1 must be able to send w_2 and $w_2 \oplus ((w_1 \oplus z_1) \times z_2)$. But then P_1 would be able to solve for z_2, contradicting locality.

We will use a particular **NEXP**-complete language for our protocol, *oracle-3-SAT*. We adapt its definition from Babai et al. (1992), in which a proof of its **NEXP**-completeness can be found.

Definition 3.8. *Let $w = z||a||b||c$ be a Boolean string of length $r + 3s$ where $|a| = |b| = |c| = s$. Let $B(w, i, j, k)$ be a Boolean formula of $r+3s+3$ variables where i, j, k are Boolean variables. A Boolean function F is a* 3-satisfying oracle for B *if for all $w = z||a||b||c$, $B(w, F(a), F(b), F(c))$ is true. B is* oracle-3-satisfiable *if such a function F exists.*

Another technique we will use is called *arithmetization of quantified Boolean formulas*. We adapt its definition from Shamir (1992).

Definition 3.9. *Let $B(x_i)_{0 \le i \le n}$ be an n-variable quantified Boolean formula.*

- *Pick a finite field $\mathbb{F}$.*
- *Change the domain of the variables from $\{0, 1\}$ to $\mathbb{F}$.*
- *Replace each occurrence of a negation ($\bar{x_i}$) by $(1 - x_i)$. Replace $\wedge$ with field multiplication $*$. Replace $\vee$ with field addition $+$.*
- *Replace universal quantifications $\forall x_i$ with product $\prod_{x_i=0,1}$. Replace existential quantification $\exists x_i$ with sum $\sum_{x_i=0,1}$.*

The resulting formula over F is the arithmetization *of the B.*

We will need the following *sumcheck protocol* from Babai et al. (1992). A detailed explanation can be found there. For our purposes, it is sufficient that we follow the steps in such a way that the answers from the corresponding prover are committed.

Protocol 3.5. *(Sumcheck Protocol)*

Let $\phi(x_1, \ldots, x_m)$ be the 3-CNF formula which the prover P is trying to show to be a tautology to a verifier V. Let $\mathbb{F}$ be a field of sufficient size (of order at least $(3c+1)m$ will suffice where c is the number of clauses of ϕ).

(1) V takes ϕ and computes its arithmetization f according to Babai et al. (1992) Proposition 3.1 and sends it to P.

(2) V and P agree on a set $I \subset \mathbb{F}$ of size at least $2dm$ where d is the degree of f.

(3) V assigns $b_0 = 0$, which is supposed to be equal to the sum

$$\sum_{x_1=0}^{1} \cdots \sum_{x_m=0}^{1} f(x_1, \ldots, x_m)^2 = 0$$

(4) $i \leftarrow 1$.

(5) P sends the coefficients of the univariate polynomial in x,

$$g_i(x) = h(r_1, \ldots, r_{i-1}, x) = \sum_{x_{i+1}=0}^{1} \cdots \sum_{x_m=0}^{1} f(r_1, \ldots, r_{i-1}, x, x_{i+1}, \ldots, x_m)^2$$

(6) V checks whether $b_{i-1} = g_i(0) + g_i(1)$. If not, abort.

(7) V chooses a random $r_i \in I$, computes $b_i = g_i(r_i)$ and sends r_i to P.

(8) If $i \leq m$ then $i \leftarrow i + 1$ and go to step 4.

(9) V checks whether $b_m = f(r_1, \ldots, r_m)^2$.

We construct our zero-knowledge, local LE-MIP for oracle-3-SAT below. A note on notation: for a circuit f, we will denote $f\left(\boxed{x}\right)$ as the gate-by-gate committed circuit evaluated with x as the input. We also use statements such as "P_1 proves to V_1 that $\boxed{\Omega_1}$ was computed correctly" by invoking at a high level the result *everything provable is provable in zero-knowledge* [Ben-Or et al. (1990)].

The reader is expected familiarity with zero-knowledge computations on committed circuits as put forward by Brassard, Crepeau (1986, 1987); Impagliazzo and Yung (1988); Kilian (1990b).

Protocol 3.6. *A local zero-knowledge LE-MIP for oracle-3-SAT*

Let $x = (B, r, s)$, an instance of oracle-3-SAT, be the common input, let $k = |x| = r + 3s + 3$, and let Λ be the verifier's program in the protocol from Babai et al. (1992).

(1) **Pre-computation:**

(a) V_1 samples two k-bit strings z_1, z_2 independently and uniformly, and provides them to V_2.

(b) V_1 selects $k+3$ random bit strings $R_1, ..., R_{k+3}$ (size specified implicitly by Λ) and evaluates the circuit of Λ using the R_i as randomness, resulting in questions $Q_1, ..., Q_{k+3}$, and provides them to V_2

(c) V_1 randomly chooses i, $1 \leq i \leq k+3$, the index of an oracle query that will be made to both P_1 and P_2. V_1 provides i to V_2.

(d) V_1 sends z_1 to P_1 and V_2 sends z_2 to P_2 for future commitments.

(e) All parties agree on a family of strongly-universal-2 hash functions $\{H_i\}$ indexed by k-bit keys.

(f) P_1 and P_2 agree on a k-bit key γ, an index to the above family.

(g) P_1 commits $\boxed{\gamma}$ to V_1.

(2) **Sumcheck with oracle:**

- Let f be the arithmetization obtained in protocol 3.5, let z be a string from I^r and $Q_{k+1}, Q_{k+2}, Q_{k+3}$ be strings of I^s as generated in protocol 3.5. V_1

and P_1 execute protocol 3.5 in committed form. At the end of this phase, P_1 shows that the committed final value is equal to

$$f\Big(z, Q_{k+1}, Q_{k+2}, Q_{k+3}, \boxed{A(Q_{k+1})}, \boxed{A(Q_{k+2})}, \boxed{A(Q_{k+3})}\Big)\,,$$

an evaluation in committed form of f using the committed values that were used during the protocol's loop. If this fails, V_1 instructs V_0 to reject.

(3) **Multilinearity test:**

(a) For $1 \leq i \leq k$:

i. V_1 sends Q_i to P_1,

ii. P_1 commits his answer as $\boxed{A(Q_i)}$.

(b) P_1 and V_1 evaluate a circuit description of Λ in committed form with inputs $\boxed{A(Q_1)}, \ldots, \boxed{A(Q_k)}$ to verify proper linearity among them. P_1 unveils the circuit's committed output. If it rejects, V_1 instructs V_0 to reject.

(4) **Consistency test:**

(a) V_1 sends i to P_1.

(b) P_1 computes $\boxed{\Omega_1} = \boxed{A(Q_i)} \oplus H_{\boxed{\gamma}}(Q_i)$ and sends $\boxed{\Omega_1}$ to V_1.

(c) P_1 proves to V_1 that $\boxed{\Omega_1}$ was computed correctly, from the existing commitments.

(d) P_1 unveils $\boxed{\Omega_1}$ for V_1, who gets Ω_1.

(e) V_2 sends Q_i to P_2 (recall that this was pre-agreed in step 1.(c))

(f) P_2 responds to V_2 with $\Omega_2 = A(Q_i) \oplus H_\gamma(Q_i)$.

(g) V_0 accepts if and only if all of the following conditions are met:

- $\Omega_1 = \Omega_2$
- All commitments which have been unveiled are valid.
- V_1 did not reject in the two previous cases

3.5.2 Proofs of Security

Locality

Since the protocol is written as an LE-MIP in which $\widehat{P} = \widehat{V} = \emptyset$, the protocol is local by definition 3.5.

Completeness

Completeness follows from the completeness of the underlying protocol of Babai et al. (1992), and the fact that the commitment protocol (protocol 3.4) is well-defined for honest provers (who will never send a commitment that they cannot unveil).

Soundness

Without loss of generality, we may assume that the soundness error in the BFL protocol to be $1/3$, through sequential amplification. The probability that our commitment scheme (protocol 3.4) fails binding is exponentially small in k. Local probabilistic provers are equivalent to local deterministic provers. This is because the success probability α of randomized provers of breaking soundness is an average over the randomized provers' random tapes. Each instance of a random tape represents a deterministic strategy. Therefore there is a deterministic strategy which succeeds with probability at least α, and hence we only need to consider local deterministic provers.

Since P_1 is deterministic, we may unambiguously consider what happens if we were to "rewind" the prover machine. Suppose that at some point P_1 unveils a particular commitment c to 0. We rewind P_1 and let V_1 make different choices before that point. Suppose that, with these alternate

choices, P_1 then unveils c to 1 (an attempt to break binding). Because of locality, P_1's behavior is independent of what P_2 receives (namely z_2). Therefore, there is only *one* such z_2 which V_0 will ultimately accept as a valid unveiling of c in both ways (recall that our commitment is statistically binding).

Therefore, in the worst case, for every commitment there exists a sequence of interactions between V_1 and P_1 such that P_1 will attempt to break the binding of that commitment. Each such commitment-breaking corresponds to at most one string z_2 that will actually work.

Let us denote the set of such binding-breaking strings by B. If $z_2 \notin B$, then the provers *will not break binding*, and the soundness error is reduced to that of the underlying protocol (at most $1/3$). On the other hand, since $|B| < \mathbf{poly}(k)$, the probability that $z_2 \in B$ is at most $\mathbf{poly}(k)/2^k$.

Therefore, the soundness error of our protocol is at most

$$Pr[z_2 \notin B \text{ and underlying protocol accepts}] + Pr[z_2 \in B] \leq \frac{1}{3} + \frac{\mathbf{poly}(k)}{2^k}.$$

Zero-Knowledge

The simulation will be divided in two parts. In the first part, the simulator produces a transcript of the *pre-computation*, *multilinearity test* and *sumcheck with oracle* parts, which involves only interactions with V_1. In the second part, the simulator will fake a valid *consistency test*.

Protocol 3.7. *(Perfectly Indistinguishable, PR-Local Simulator for Protocol 3.6, Part 1)*

The setup:

- Let $(\widehat{M}, M_1, M_2)$ be a set of locality-explicit simulators.
- M_1 and M_2 can send $\widehat{M}$ an index along with a bit.

- $\widehat{M}$ completes the indexed PR box (protocol 3.1) for both simulators.

The simulation strategy:

(1) The simulators agree on unique indices for every commitment used in the protocol.

(2) M_1 interacts with V_1 the way P_1 would. Whenever P_1 should commit, M_1 commits to random bits, just like the single-simulator from section 3.5.

(3) For each commitment, V_2 sends M_2 a string s. M_2 sends to $\widehat{M}$ the index of the commitment and s.

(4) $\widehat{M}$ runs the PR box (protocol 3.1) and replies with V_2's half of the output.

(5) Whenever M_1 needs to unveil a commitment, it can be unveiled in the way M_1 desires by sending the corresponding index and bit to $\widehat{M}$.

(6) $\widehat{M}$ completes the corresponding PR box which outputs t. $\widehat{M}$ sends t to M_1.

(7) M_1 sends t to V_1.

The second part (the consistency test) can be done by having the simulators ignore the question.

Protocol 3.8. *(Perfectly Indistinguishable, PR-Local Simulator for Protocol 3.6, Part 2)*

(1) V_1 sends i to M_1.

(2) M_1 computes $\boxed{\Omega_1} = H_{\boxed{\gamma}}(Q_i)$.

(3) Using $\widehat{M}$ to break binding, M_1 convinces V_1 that $\boxed{\Omega_1}$ is actually $\boxed{A(Q_i)} \oplus$

$H_{\boxed{\gamma}}(Q_i)$ instead.

(4) M_1 unveils $\boxed{\Omega_1}$ for V_1, who gets $\Omega_1 = H_\gamma(Q_i)$.

(5) V_2 sends Q'_i to M_2.

(6) M_2 responds with $\Omega_2 = H_\gamma(Q'_i)$.

By the properties of the strongly-universal-2 hash H, if $Q_i = Q'_i$ then $\Omega_1 = \Omega_2$. Otherwise $\Omega_1 \neq \Omega_2$ with probability exponentially close to one. This produces the result as desired. The simulators then feed the transcripts to V_0, and terminates simulation.

3.5.3 Entangled Simulators

The binding condition of commitment used above (protocol 3.4) can be broken given PR-boxes. However, if the verifier were willing to tolerate approximately 15% of errors in the provers' unveiling string (z_1 or 0), then it is possible to break binding with shared entanglement [Brassard, Broadbent, and Tapp (2003)] while maintaining soundness against local provers. Using this weakened version of commitment in place of protocol 3.4 still yields a local LE-MIP for oracle-3-SAT, but easier to simulate (using weaker non-local resources). We leave the details of this modified protocol to the reader.

3.6 Zero-Knowledge and Non-Locality

The heart of zero-knowledge is the idea of a *simulator*: a machine, with no more power than the verifier, can output a transcript indistinguishable from an interaction involving an all-powerful prover.

This vast asymmetry in computational power means that this simulator cannot accomplish this task without some kind of *advantage*; this advantage must be *independent of knowledge*, else we would lose zero-knowledgeness.

In the case of single-prover zero-knowledge proofs, this advantage can be in the form of the ability

to *rewind* computation, the ability to discard failed simulations, or knowledge of a trapdoor in a commitment scheme. The simulator can use these things because we only care that the transcript is generated in polynomial-time (or expected polynomial-time).

In the case of multi-prover zero-knowledge proofs, the advantage in existing literature can be summed up as *signaling*: the simulator, pretending to be several provers, knows secrets which real provers, in a real instance of the protocol with separated provers, would not. That is, real provers in a real run of a ZKMIP are unable to communicate (or they are unable to communicate faster-than-light, in the case of relativistic instantiations of MIPs).

In either case, from a complexity perspective, the simulator's advantage can be anything as long as it is truly independent of knowledge – we do not want to exclude anything a priori. But, in practice, zero-knowledge is ultimately applied cryptography and, from a cryptographic perspective, *not all advantages are equal*.

3.7 Minimal Simulator Advantage

In existing ZKMIP literature, the (single) simulator's advantage is its ability to interact with both verifiers at once. This is equivalent to having a pair of signaling simulators. However, it turns out that simulators do not need to signal in order to break the above commitment (section 3.3); a weaker non-local distribution will do. The "blind spot" of the previous chapter, which was a gap in the *soundness* analysis of MIPs, is an unexplored dimension of characterization in the *zero-knowledge* analysis of ZKMIPs.

The framework in which this "non-local advantage" can be analyzed is locality-explicit MIPs; specifically, locality explicit simulators (definition 3.6) and zero-knowledge LE-MIPs (definition 3.7). This naturally leads us to ask: *what is the minimal simulator advantage needed for achieving zero-knowledge for* $\mathbf{NEXP}$*?*

It is clear that signaling simulators can succeed in our protocol from the previous chapter; this is

how standard ZKMIP simulators achieve indistinguishability. We can summarize this as

$$\mathbf{ZK}^{\mathbf{SIG}}\mathbf{MIP}^{\emptyset}_{\emptyset} = \mathbf{NEXP},$$

where $\mathbf{SIG}$ is a signaling correlator.

However, signaling is unnecessary, as the binding condition of commitment used above (protocol 3.4) can be broken given $\mathbf{PR}$-boxes. Thus, the simulator's advantage can be lowered to $\mathbf{PR}$-boxes, or

$$\mathbf{ZK}^{\mathbf{PR}}\mathbf{MIP}^{\emptyset}_{\emptyset} = \mathbf{NEXP}.$$

If the verifiers were willing to tolerate approximately 15% of errors in the provers' unveiling string (z_1 or 0), then it is possible to break binding with shared entanglement [Brassard et al. (2003)] while maintaining soundness against local provers. Making this slight change in the protocol reduces the simulator advantage further:

$$\mathbf{ZK}^{\mathbf{QNL}}\mathbf{MIP}^{\emptyset}_{\emptyset} = \mathbf{NEXP},$$

where $\mathbf{QNL}$ denotes polynomial amount of shared entanglement for the simulators.

Ideally, the simulators would not need any non-local advantage over the verifiers. However, we are unable to find a zero-knowledge MIP where the simulators are *local* which can accept $\mathbf{NEXP}$; nor can we prove that it is impossible. We make the following conjecture:

Conjecture 3.1. $\mathbf{ZK}^{\emptyset}\mathbf{MIP}^{\emptyset}_{\emptyset} = \mathbf{SZK}$*, where* $\mathbf{SZK}$ *is the set of languages with statistical zero-knowledge interactive proofs without computational assumptions (e.g., graph isomorphism).*

3.8 Advantage Trade-Offs

As a further example of the drastic differences between MIP simulators' non-local advantages and single-prover IP simulators' advantages (e.g., rewinding), consider the following:

Theorem 3.1. *Suppose that the provers in protocol 3.6 have access to PR-boxes (thus they are*

no-signaling, but not local), then the protocol is not sound.

Proof. The provers adopt the simulators' strategy. Since commitment binding is broken with the aid of PR-boxes, the verifiers will always accept. □

In this case, a change in non-locality which would render a protocol unsound is exploited to produce simulations. Conversely, if the simulators' strategy is dependent entirely on using the different non-locality, then malicious provers with this change in non-locality can use it to break soundness.

This is contrasted with single-prover zero-knowledge, in which a prover having the ability to rewind computations, although enough for simulators in IPs, is not enough to break soundness. The relationship between zero-knowledge and soundness will have to be explored in a future work. We leave this chapter with the following conjecture:

Conjecture 3.2. *If an LE-MIP is* $\mathbf{ZK}^{\widehat{M}}\mathbf{MIP}^{\widehat{P}}_{\widehat{V}}$*, then it is not in* $\mathbf{MIP}^{\widehat{M}}_{\widehat{V}}$*. That is, if the provers have access to an* $\widehat{M}$ *correlator, then the protocol is not sound.*

3.9 Chapter Conclusions

Although protocol 3.6 is a *local* LE-MIP, the only known ways of simulating the transcript are to give the simulators some kind of non-local resource such as a PR box (or a fully signaling box, but that is not necessary). We do not know whether it is possible to simulate protocol 3.6 with *local* simulators, but we are unable to show this to be impossible.

$\mathbf{MIP}$ is cryptographic. $\mathbf{NEXP}$ is complexity theoretic. Although there exists a MIP which accepts $\mathbf{NEXP}$ (resolving the complexity of $\mathbf{MIP}$), the presence of non-locality with respect to zero-knowledge has not been explored. Zero-knowledge simulators need advantages in order to function. In the case of MIPs, it was always implicitly assumed this advantage is necessarily signaling. We have shown that this is not true.

Chapter 4

Succinct Zero-Knowledge under Relativistic Assumptions

4.1 Relativistic Motivation

The need for more nuanced simulators is motivated by relativistic cryptography, an example of which can be found in Lunghi et al. (2015). Relativistic cryptography exploits the fact that it is impossible to signal faster than light. We can enforce the no-signaling condition of MIPs by spatially separating the provers from each other. In order to enforce the provers' spatial separation during the execution of the protocol, each prover is paired with a verifier of its own, which is located nearby. The verifiers can use the timing of the replies of their respective provers to judge their relative distance.

In practice, this means that we can implement MIPs under relativistic assumptions if the verifier can be "split" into multiple verifiers, each locally interacting with its corresponding prover. An example of relativistic cryptography can be found in Lunghi et al. (2015), where a commitment was sustained for over 24 hours.

This chapter is adapted from a paper by Claude Crépeau, Arnaud Y. Massenet, Louis Salvail, Lucas Shigeru Stinchcombe, and Nan Yang. It has been accepted for publication in the Proceedings of Information Theoretic Cryptography 2020.

In this chapter, we answer the question which was raised in the introduction, "How do we enforce non-contamination physically?"

Some MIPs have verifiers which, intrinsically, cannot be split. Examples include Ben-Or et al. (1988) and Kilian (1990b). In these examples, the verifier is used to courier an authenticated message between provers. In the relativistic setting, if the verifier has time to pass a message between provers, then the provers just signal between themselves.

Luckily, most MIPs in the literature have verifiers that are *non-adaptive*. These verifiers' questions to one prover are independent of the answers from all the provers. MIPs with non-adaptive verifiers can be rewritten into a format with multiple, split verifiers; this format we will call *locality-explicit*, and is formally defined in section 3.4.

As an example of what we mean, consider the following two-prover interactive proof for graph 3-coloring:

Protocol 4.1. *(Simple MIP, Single-Verifier)*

Two provers P_1, P_2, one verifier V. On input graph G, P_1 and P_2 agree on a 3-coloring.

(1) V asks P_1 for the colors of an edge e.

(2) V asks P_2 for the colors of one of the nodes of e.

V accepts if and only if the colors of that edge from P_1 are not equal, and P_2 corroborates with P_1's answer by replying with the same color for the same node.

In the above protocol, V's questions to either prover does not depend on answers from any prover. This is what is commonly known as a *non-adaptive* verifier. We can therefore split the above verifier into a two-verifier version:

Protocol 4.2. *(Simple MIP, Multi-Verifier)*

Two provers P_1, P_2, two verifiers V_1, V_2. On input graph G, P_1 and P_2 agree on a 3-coloring, V_1 and V_2 agree on an edge e.

(1) V_1 asks P_1 for the colors of e.

(2) V_2 asks P_2 for the colors of one of the nodes of e.

Post execution, V_1 and V_2 confer with each other, and accept if and only if the colors of that edge from P_1 are not equal, and P_2 corroborates with P_1's answer by replying with the same color.

This version of the protocol is naturally suited for relativistic implementation. However, it is not zero-knowledge because even if P_1 and P_2 agreed on a randomly selected 3-coloring each time, a dishonest verifier V_2 may sample a node which is not from e. We can make a zero-knowledge, multi-verifier MIP with the help of the following commitment scheme, which is adapted from Ben-Or et al. (1988):

Protocol 4.3. *(Multi-Verifier Commitment)*

Two provers P_1, P_2, two verifiers V_1, V_2. The provers share a random string w, and the verifiers share a random string r. Operations are over a finite field. P_1 wishes to commit b.

(1) (Commit) V_1 sends P_1 the string r. P_1 replies with $x = w + br$.

(2) (Unveil) P_2 sends V_2 the string w.

Post execution, the verifiers confer. They accept if and only if $x + w = r$ or $x + w = 0$.

Combining protocol 4.3 and the zero-knowledge protocol of Goldreich, Micali, and Wigderson (1991) gives us a zero-knowledge, multi-verifier MIP.

Protocol 4.4. *(ZKMIP, Multi-Verifier)*

Two provers P_1, P_2, two verifiers V_1, V_2. On input graph G, P_1 and P_2 agree on a randomly selected 3-coloring and $2|V|$ strings w_i, V_1 and V_2 agree on an edge e and $2|V|$ strings r_i.

(1) P_1 commits the coloring of G to V_1 using the $2|V|$ w_i, r_i they pre-agreed.

(2) V_2 asks P_2 to unveil the colors of the edge e.

Post execution, V_1 and V_2 confer with each other, and accept if and only if the commitment is valid, and the colors unveiled are not equal.

What makes this protocol zero-knowledge? In the commitment scheme (protocol 4.3), if P_2 has knowledge of r, then it can break the commitment by unveiling either way (by sending w or $w + r$ as needed). Following the precedents set by existing literature's definition of zero-knowledge, the (*single*) simulator, interacting with both verifiers, learns r. Therefore it can break the commitment and always unveil a color that will be accepted by the verifiers.

4.2 The Hidden Cost of Zero-Knowledge

Using relativistic assumptions to build cryptography can be costly, even in theory. Specifically, as in the case of Lunghi et al. (2015), the spatial (in the complexity sense) and the communication

complexity of sustaining n commitments is linear *in the length of time that the commitments is sustained*.

The idea of using distance and special relativity (a theory of motion justifying that the speed of light is a sort of asymptote for displacement) to prevent communication between participants to multi-prover proof systems can be traced back to Kilian (1990a). Probably, the original authors (Ben Or, Goldwasser, Kilian and Wigderson) of Ben-Or et al. (1988) had that in mind already, but it is not explicitly written anywhere. Kent was the first author to venture into *sustainable* relativistic commitments [Kent (1999)] and introduced the idea of arbitrarily prolonging their life span by playing some ping-pong protocol between the provers (near the speed of light). This idea was made considerably more practical by Lunghi *et al.* in Lunghi et al. (2015) who made commitment sustainability much more efficient. This culminated into an actual implementation by Verbanis *et al.* in Verbanis et al. (2016) where commitments were sustained for more than a day!

As nice as this may sound, such *long-lasting* commitments have found so far very little practical use. Consider for instance the zero-knowledge proof for Hamiltonian Cycle as introduced by Chailloux and Leverrier [Chailloux and Leverrier (2017)]. Proving in zero-knowledge that a 500-vertex graph contains a Hamiltonian cycle would require transmitting $250\,000$ bit commitments (each of a couple hundreds of bits in length) and eventually sustaining them before the verifier can announce his choice of unveiling the whole adjacency matrix or just the Hamiltonian cycle. For a graph of $|V|$ vertices, this would require an estimated $200|V|^2$ bits of communication before the verifier can announce his choice $chall$ (see Fig. 4.1). This makes the application prohibitively expensive. If you use a larger graph, you will need more time to commit, leading to more distance to implement the protocol of Chailloux and Leverrier (2017). Either a huge separation is necessary between the provers (so that one of them can unveil according to the verifier's choice $chall$ before he finds out the committal information B used by the other prover while the former must commit all the necessary information before he can find out the verifier's choice $chall$) or we must achieve extreme communication speeds between prover-verifier pairs. This would only be possible by vastly parallelizing communications between them at high cost. Modern (expensive) top-of-line communication equipment may reach throughputs of roughly 1Tbits/sec. A back of the envelope calculation estimates

that the distance between the verifiers must be at least 100 km to transmit 250 000 commitments at such a rate.

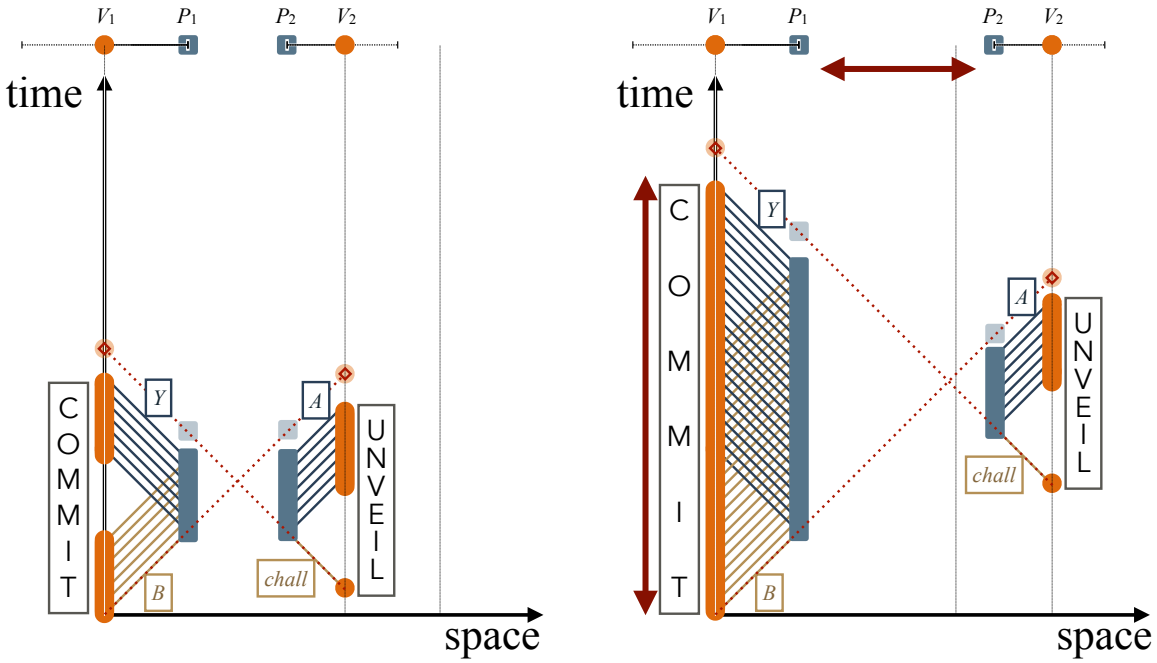


Figure 4.1: Space-Time diagrams of Chailloux and Leverrier (2017)'s ZK-MIP* for **NP**. (45° diagonals are the speed of light.)

In the above two diagrams, V_1 at a first location sends a random matrix $\boxed{B}$ to P_1 who uses each entry to commit an entry of the adjacency matrix $\boxed{Y}$ of G. At another location, V_2 sends a random challenge $\boxed{chall}$ to P_2 who unveils all or some commitments as $\boxed{A}$. At all times, V_1 and V_2 must make sure that the answers they get from P_1 and P_2 come early enough that the direct communication line between V_1 and V_2 (even at the speed of light) is not crossed. The transition from left to right shows that increasing the number of vertices (and thus increasing the total commit time) pushes the verifiers further away from each other. In Chailloux and Leverrier (2017) the distance must increase quadratically with the number of vertices in the graph.

In this work we consider the following problem: in a Multi-Prover environment, how little spatial separation is sufficient to assert the validity of an **NP** statement in perfect zero-knowledge? We exhibit a set of two novel zero-knowledge protocols for the 3-COLorability problem that use two (*local*) provers or three (*entangled*) provers and only require them to communicate two trits each after having each received an edge and two bits each from the verifier. This greatly improves the ability to prove zero-knowledge statements on very short distances with very minimal communication equipment. In comparison, the protocol of Chailloux and Leverrier (2017) would require transmitting millions of bits between a prover and his verifier before the latter may disclose what

to unveil or not. This implies the provers would have to be very far from each other because all of these must reach the verifier *before* the formers can communicate.

Although certain algebraic zero-knowledge multi-prover interactive proofs for $\mathbf{NP}$ and $\mathbf{NEXP}$ using explicitly no commitments at all have been presented before in Lapidot and Shamir (1995), Feige and Kilian (1994) (sound against local provers) and Chiesa, Forbes, Gur, and Spooner (2018),Grilo, Slofstra, and Yuen (2019) (sound against entangled provers), in the local cases making these protocols entanglement sound is absolutely non-trivial, whereas in the entangled case the multi-round structure and the amount of communication in each round makes implementing the protocol completely impractical as well. (In their defense, the protocols were not designed to be *practical*).

The main technical tool we use in this work is a general Lemma of Kempe, Kobayashi, Matsumoto, Toner, and Vidick (2011) to prove soundness of a three-prover protocol when the provers are *entangled* based on the fact that a two-prover protocol version is sound when the provers are only *local*. More precisely, they proved this when the three-prover version is the same as the two-prover version but augmented with an extra prover who is asked exactly the same questions as one of the other two at random and is expected to give the same exact answers.

Our protocols build on top of the earlier protocol in Cleve et al. (2004) who presented an extremely simple and efficient solution to the 3-COL problem that uses only two provers, each of which is queried with either a vertex from a common edge, or twice the same vertex. In the former case, the verifier checks that the two ends of the selected edge are of distinct colors, while in the latter case, the verifier checks only that the provers answer the same color given the same vertex. On the bright side, their protocol did not use commitments at all but unfortunately it did not provide zero-knowledge either. Moreover, it is a well established fact that this protocol cannot possibly be sound against entangled provers, because certain graph families have the property that they are not 3-colorable while having entangled-prover pairs capable of winning the game above with probability one. This was already known at the time when they introduced their protocol. The reason this protocol is not zero-knowledge follows from the undesirable fact that dishonest verifiers can discover the (random) coloring of non-edge pairs of vertices in the graph, revealing if they are of the same color or not in

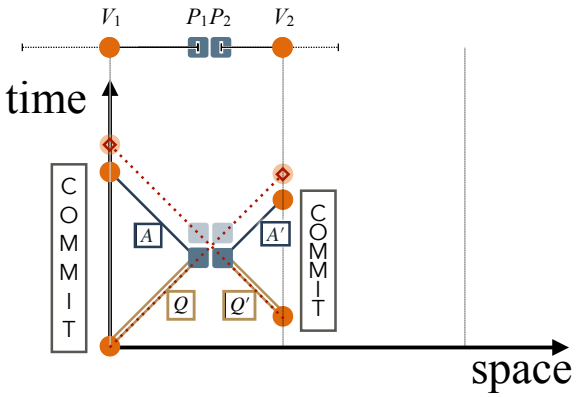


Figure 4.2: Space-Time diagram of our ZK-MIP* for **NP**. ($45°$ diagonals are the speed of light.)

the provers' coloring.

We are able to remedy to the zero-knowledge difficulty by allowing the provers to use commitments for the color of their vertices. However they use these commitments in an innovative way that we call the *unveil-via-commit principle* (of independent interest) explained below. For this purpose we use commitments similar to those of Lunghi et al. (2015) but in their simplest form possible, over the field $\mathbb{F}_3$ (or $\mathbb{F}_4$ if you insist working in binary), and thus with extremely weak binding property but also minimal in communication cost: a complete execution of the basic protocol transmits a question Q of exactly one edge number (using only $\log|E|$ bits) and two bits from verifiers to provers and an answer A of two trits back from the provers to verifiers (see Fig. 4.2). This implies that for a fixed communication speed, the minimal distance of the provers in our protocol increases logarithmically with the number of vertices whereas the same parameter grows quadratically in Chailloux and Leverrier (2017). Nevertheless, this is good enough to obtain a zero-knowledge version of the protocol that remains sound against *local* pairs of provers. The main idea being that the provers will each commit to the colors of two requested vertices only if they form an edge of the graph. To unveil the color of any vertex, the verifiers must request commitment of the *same* vertex by *both* provers but using different randomizations. This way the verifiers may compute the color of a vertex from the *linear system* established by the two commitments and not by explicitly requesting anyone to unveil. This is the unveil-via-commit principle (very similar to the double-spending detection mechanism of the untraceable electronic cash of Chaum, Fiat, and Naor (1990)). We then use the Lemma of Kempe et al. (2011) to prove soundness of the three-prover version of

this protocol even when the provers are *entangled*. A positive side of the protocol of Chailloux and Leverrier (2017), however, is the fact that only two provers are necessary while we use three. Zero-knowledge follows from the fact that only two edge vertices can be unveiled by requesting the same edge to both provers. Otherwise only a single vertex may be unveiled. Finally, we show that even the three-prover version of this protocol retains the zero-knowledge property: requesting any three edges from the provers may allow the dishonest verifiers to unveil the colors of a triangle in the graph but never two end-points that do not form an edge (going to four provers would however defeat the zero-knowledge aspect).

An actual physical implementation of this protocol is currently being developed.

4.2.1 Implementations Issues

Traditionally in the setup of Multi-Prover Interactive Proofs, there is a single verifier interacting with the many provers. However, when implementing no-communication via spatial separation (the so called relativistic setting) it is standard to break the verifier in a number of verifiers equal to the number of provers, each of them interacting at very short distance from their own prover. The verifiers can use the timing of the replies of their respective provers to judge their relative distance. In practice, this means that we can implement MIPs under relativistic assumptions if the verifier are "split" into multiple verifiers, each locally? interacting with its corresponding prover. The verifiers use the distance between *themselves* to enforce the impossibility of the provers to communicate: no message from a verifier can be used to reply to another verifier faster than the speed of light *wherever the provers are located*.

Moreover, multi-prover interactive proof systems may have several rounds in addition to several provers. In general, protocols with several rounds may cause a threat to the inherent assumption that the provers are not allowed to communicate during the protocol's execution. Nevertheless, most of the existing literature resolves this issue by providing an honest verifier that is *non-adaptive*. To simplify this task, most of the protocols are actually single-round. We stick to these guidelines in this work. Moreover, in order to prove soundness of our protocols against entangled provers, we use a theorem that is currently only proven for single-round protocols. The protocols we describe

are indeed single-round and non-adaptive.

4.3 Preliminaries

4.3.1 Notations

In the following, $G = (V, E)$ denotes an undirected graph with vertices V and edges E. If $n = |V|$ then we denote the set of vertices in G by $V = \{1, 2, \ldots, n\}$. We suppose that $(i, i) \notin E$ for all $1 \leq i \leq n$ (i.e. G has no loop). We denote uniquely each edge in E as (i, j) with $j > i$. For $i \in V$, let $\mathsf{Edges}(i) := \{(j, i) \in E\}_{j<i} \cup \{(i, j) \in E\}_{j>i}$ be the set of edges connecting vertex i in G. For $e, e' \in E$, we define $e \cap e' = i \in V$ if e and e' have only one vertex $i \in V$ in common. When e and e' have four distinct vertices in V, we set $e \cap e' = 0$. Finally, when $e = e'$, we set $e \cap e' := \infty$. For readability, we use the following special notations: $(a, b) \neq\!\!\!\!\!\!\;\;\; (c, d)$ means $a \neq c$ **and** $b \neq d$, while as always, $(a, b) \neq (c, d)$ simply means $a \neq c$ **or** $b \neq d$.

4.3.2 Non-local Games and Relativistic Multi-Prover Interactive Proofs

Consider a k–prover interactive proof system $\Pi(x)$ (with or without perfect completeness) for L executed with public input $x \notin L$. In this situation, $\Pi(x)$ defines a so-called *quantum game*. The minimum $q(|x|)$ such that for all $\mathsf{P}'_1, \ldots, \mathsf{P}'_k$, $\Pr\{([\mathsf{P}'_1, \ldots, \mathsf{P}'_k, \mathsf{V}](x) = ACCEPT)\} \leq q(|x|)$ is called the *classical value of game* $\Pi[x]$ and is denoted $\omega(\Pi(x))$ when the provers are restricted to be classical and unable to communicate with each other upon public input x. When the provers, still unable to communicate with each other, are allowed to carry their computation quantumly and share entanglement, we denote by $\omega^*(\Pi(x)) \geq \omega(\Pi(x))$ the minimum $q(|x|)$ such that for all such quantum provers $\mathsf{P}'_1, \ldots, \mathsf{P}'_k$, $\Pr\{([\mathsf{P}'_1, \ldots, \mathsf{P}'_k, \mathsf{V}](x) = ACCEPT)\} \leq q(|x|)$. In this case, $\omega^*(\Pi(x))$ is called the *quantum value of game* $\Pi(x)$. A k–prover interactive proof system for L is said to be *symmetric* if V can permute the questions to all provers without changing their distribution.

The following result of Kempe, Kobayashi, Matsumoto, Toner, and Vidick [Kempe et al. (2011)] shows that the classical value of a symmetric one-round classical game cannot be too far from the

quantum value of a *modified* game. Given a symmetric one-round two-prover game Π, one can always add a third prover P_3 and V asks P_3 the same question than P_1 with probability $\frac{1}{2}$ or the same question than P_2 with probability $\frac{1}{2}$. Then, V accepts if P_1 and P_2 would be accepted in $\Pi(x)$ and if P_3 returns the same answer as the one issued by the prover it emulates. We call $\Pi'(x)$ the modified game obtained that way from $\Pi(x)$.

Lemma 4.1 (Kempe et al. (2011), Lemma 17)**.** *Let $\Pi(x)$ be a two-prover one-round symmetric game and let $\Pi'(x)$ be its modified version with three provers. If $\omega^*(\Pi'(x)) > 1-\varepsilon$ then we always get $\omega(\Pi(x)) > 1-\varepsilon-12|Q|\sqrt{\varepsilon}$ where Q is the set of V's possible questions to a prover in Π.*

Lemma 4.1 remains true for non-symmetric two-prover one-round protocol by first making them symmetric at the cost of increasing the size of Q. This is always possible without changing the classical value of the game and by using at most twice the number of questions $|Q|$ of the original game (Lemma 4 in Kempe et al. (2011)).

4.3.3 Multi-Prover Commitments with Implicit Unveiling

Our multi-prover proof systems for 3COL use a simple 2-committer commitment scheme with a property allowing to guarantee perfect zero-knowledge. In this section, we give the description of this simple commitment scheme with its important properties for our purposes.

Assume that provers P_1 and P_2 share ℓ values $c_1, c_2, \ldots, c_\ell \in \mathbb{F}$ where $\mathbb{F}$ is a finite field. V wants to check that these values satisfy some properties without revealing the specific values.

Bit commitment schemes have been used in the multi-prover model ever since it was introduced in Ben-Or et al. (1988). The original scheme was basically, for $1 \leq i \leq l$, $w_i := b_i \cdot r_i + c_i$, a commitment w_i to value $c_i \in \mathbb{F}$ using pre-agreed random mask $b_i \in_R \mathbb{F}$ and randomness $r_i \in \mathbb{F}^*$ provided by V. Kilian [Kilian (1990b)] had a binary version where each bit $c_i := c_i^1 \oplus c_i^2 \oplus c_i^3$ is shared among provers P_1 and P_2 (and therefore $\mathbb{F}$ needs only to be a group). To commit c_i, V samples c_i^h from P_1 and c_i^j from P_2 at random. If $j = h$ but $c_i^j \neq c_i^h$, V immediately rejects the commitment. Otherwise either P_1 or P_2 may unveil by disclosing c_i^1, c_i^2, c_i^3 at a later time. Somehow, Crépeau's bad recollection of the scheme in Ben-Or et al. (1988) lead Brassard, Crépeau,

Mayers, and Salvail (1998) to a similar but different scheme defining $w_i := c_i \cdot r_i + b_i$, a commitment w_i to bit $c_i \in \{0,1\}$ using pre-agreed bit mask $b_i \in_R \{0,1\}$ and binary randomness r_i provided by their corresponding verifiers. Although this latter form of commitment is intimately connected to the CHSH game [Clauser, Horne, Shimony, and Holt (1969)] and the Popescu-Rohrlich box [Popescu and Rohrlich (1994)], this proximity is not relevant for the soundness and the completeness of our protocols, even against entangled provers. While the binding property of the latter scheme has been established in Chailloux and Leverrier (2017); Crépeau et al. (2011); Fehr and Fillinger (2015); Kent (1999); Lunghi et al. (2015); Verbanis et al. (2016) against entangled provers, it is still not clear how to get sound and complete proof systems against such provers. We shall rather get completeness and soundness against entangled provers using a different technique from Kempe et al. (2011) that uses a third prover.

For an arbitrary field $\mathbb{F}$, the commitment scheme produces commitment $w_i := c_i \cdot r_i + b_i$ to field element $c_i \in \mathbb{F}$ using pre-agreed field element mask b_i (specific to value $1 \leq i \leq \ell$) and random field element $r_i \in \mathbb{F}^*$ provided by their corresponding verifiers. Many results were proven for this specific form of the commitments. Notice however that the two versions discussed above, $w_i := b_i \cdot r_i + c_i$ in the former case and $w_i := c_i \cdot r_i + b_i$ in the latter have equivalent binding property (left as a simple exercise). Considering, the former as being the degree-one secret sharing [Shamir (1979)] of c_i hidden in the degree zero term, while the latter being the degree-one secret sharing of c_i hidden in the degree one term, we decided to use the former (original BGKW form) because all the known results about secret sharing are generally presented in this form. In particular, this form is more adapted to higher degree generalizations such as $w_i := a_i \cdot r_i^2 + b_i \cdot r_i + c_i$ being the degree-two secret sharing of c_i hidden in the degree zero term, and so on.

Moreover, this choice turns out to simplify our (perfect) zero-knowledge simulator. For the rest of this paper, we use $w_i := b_i \cdot r_i + c_i$ where $w_i, b_i, c_i \in \mathbb{F}_3$ and $r_i \in \mathbb{F}_3^*$. Provers therefore commit to trits, one value for each vertex corresponding to its color in a 3–coloring of graph $G = (V, E)$. The values shared between P_1 and P_2 are therefore, for each vertex $i \in V$, the color c_i of that vertex and a vertex specific random mask b_i.

Suppose that V asks P_1 to commit on the color c_i of vertex $i \in V$ using randomness $r \in_R \mathbb{F}_3^*$. Let $w = b_i \cdot r + c_i$ be the commitment returned to V by P_1. Suppose V asks P_2 to commit on the color c_j of vertex $j \in V$ using randomness $r' \in_R \mathbb{F}_3^*$. Let $w' = b_j \cdot r' + c_j$ be the commitment issued to V by P_2. The following 3 cases are possible depending on V's choices for i, j, r, and r':

(1) *(forever hiding)* if $i \neq j$ then V learns nothing on neither c_i nor c_j since w and w' hide c_i and c_j with random and independent masks $b_i \cdot r$ and $b_j \cdot r'$ respectively. Even knowing $r, r' \in \mathbb{F}_3^*$, $b_i \cdot r$ and $b_j \cdot r'$ are uniformly distributed in $\mathbb{F}_3$.

(2) *(consistency testing)* If $i = j$ and $r = r'$ then V can verify that $w = w'$. This corresponds to the immediate rejection of V in Kilian's two-prover commitment described above. It allows V to make sure that P_1 and P_2 are consistent when asked to commit on the same value.

(3) *(implicit unveiling)* If $i = j$ and $r' \neq r$ then V can learn c_i (assuming $w = b_i \cdot r + c_i$ and $w' = b_i \cdot r' + c_i$) the following way. V simply computes $c_i := -(w + w')$ (Note that over an arbitrary field $c_i := (wr' - w'r)(r' - r)^{-1}$ whenever $r \neq r'$). Interpreting the meaning of this test can be done when considering a strategy for P_1 and P_2 that always passes the consistency test. In this case, $w - b_i \cdot r = c_i = w' - b_i \cdot r'$ are satisfied and V learns c_i.

As long as P_1 and P_2 are *local* (or *quantum non-local*) they cannot distinguish which option V has picked among the three. The consistency test makes sure that if P_1 and P_2 do not commit on identical values for some $i \in V$ then V will detect it when V picks the consistency test for commitment w and w' in position i.

4.4 Classical Two-Prover Protocol

First, consider a small variation over the protocol of Cleve et al. presented in Cleve et al. (2004). In their protocol, when P_1 and P_2 both know and act upon the same valid 3-coloring of G, V asks each prover for the color of a vertex in $G = (V, E)$. Consistency is verified when V asks the same vertex to each prover and compares that the same color has been provided. The colorability is checked

when the provers are asked for the color of two connected vertices in G. This way of proceeding is however problematic for the zero-knowledge condition. V could be asking two vertices that do not form an edge for which their respective color will be unveiled. This certainly allows V to learn something about P_1's and P_2's coloring. Indeed, repeating this many times will allow V to efficiently reconstruct a complete coloring. To remedy partially this problem, V is instead asking each prover the coloring of an entire edge of G. The coloring is (only) checked when both provers are asked the same edge, while consistency is checked when two intersecting edges are asked to the provers.

4.4.1 Distribution of questions

Let $G = (V, E)$ be a connected undirected graph. Let us define the probability distribution $\mathcal{D}_G = \{(p(e, e'), (e, e'))\}_{e,e' \in E}$ for the pair $(e, e') \in E \times E$ that V picks with probability $p(e, e')$ before announcing e to P_1 and e' to P_2. For $e, e' \in E$ such that $e \cap e' = 0$, we set $p(e, e') := 0$ so that V never asks two disconnected edges in G (this would be useless).

The first thing to do is to pick $e = (i, j) \in E$ uniformly at random. With probability ϵ (to be selected later), we set $e' = e$, which allows for an edge-verification test. With probability $1 - \epsilon$, we perform a well-definition test as follows. With probability $\frac{1}{2}$, $e' \in \mathsf{Edges}(i)$ uniformly at random and with probability $\frac{1}{2}$, $e' \in \mathsf{Edges}(j)$ uniformly at random. In other words, the well-definition test picks the second edge e' with probability $\frac{1}{2}$ among the edges connecting $i \in V$ and with probability $\frac{1}{2}$ among the edges connecting $j \in V$. It follows that for $e = (i, j) \in E$ and $e' \in (\mathsf{Edges}(i) \cup \mathsf{Edges}(j)) \setminus \{e\}$, we have

$$p(e, e') = \frac{1 - \epsilon}{2|E|} \left(\frac{|\{e'\} \cap \mathsf{Edges}(i)|}{|\mathsf{Edges}(i)|} + \frac{|\{e'\} \cap \mathsf{Edges}(j)|}{|\mathsf{Edges}(j)|} \right) \ . \tag{1}$$

We also get

$$p(e, e) = \frac{\epsilon}{|E|} + \frac{1 - \epsilon}{2|E|} \left(\frac{1}{|\mathsf{Edges}(i)|} + \frac{1}{|\mathsf{Edges}(j)|} \right) \geq \frac{\epsilon}{|E|} \ . \tag{2}$$

It is easy to verify that $\mathcal{D}_G$ is a properly defined probability distribution over pairs of edges.

4.4.2 A Variant Over the Two-Prover Protocol of Cleve et al.

Distribution $\mathcal{D}_G$ produces two edges where the first one is provided to P_1 while the second one is provided to P_2. Each prover then returns the color of each vertex of the edge to V. We denote the resulting protocol $\Pi^{(2)}_{\text{std}}$.

Protocol $\Pi^{(2)}_{\text{std}}[G]$: Two-prover, 3-COL.

Provers $\mathsf{P}_1, \mathsf{P}_2$ pre-agree on a random 3-coloring of G:

$\{(i, c_i) | c_i \in \mathbb{F}_3\}_{i \in V}$ such that $(i,j) \in E \implies c_j \neq c_i$.

Interrogation phase:

- V picks $((i,j),(i',j')) \in_{\mathcal{D}_G} E \times E$, sends (i,j) to P_1 and (i',j') to P_2.
- If $(i,j) \in E$ then P_1 replies with c_i, c_j.
- If $(i',j') \in E$ then P_2 replies with $c_{i'}, c_{j'}$.

Check phase:

- **Edge-Verification Test:**
  if $(i,j) = (i',j')$ then V accepts iff $c_i = c_{i'} \neq c_{j'} = c_j$.
- **Well-Definition Test:**
  if $(i,j) \cap (i',j') = h \in V$ then V accepts iff $c_h = c'_h$.

The perfect soundness of this protocol is not difficult to establish along the same lines of the proof of soundness for the original protocol in Cleve et al. (2004). On the other hand, zero-knowledge does not even hold against honest verifiers. V learns the color of each vertex contained in any two edges of G. This is certainly information about the coloring that V learns after the interaction. To some extend, the modifications we applied to the 2-prover interactive proof system of Cleve et al.

(2004) leaks even more to V. In the next section, we show that the 2-prover commitment scheme, that we introduced in Sect. 4.3.3, can be used in protocol $\Pi^{(2)}_{\text{std}}$ to prevent this leakage completely.

4.5 Perfect Zero-Knowledge Two-Prover Protocol

We modify the protocol of section 4.4.2 to prevent V from learning the colors of more than two connected vertices in G. The idea is simple, P_1 and P_2 will return commitments for the colors of the vertices asked by V. The implicit unveiling of the commitment scheme described in section 4.3.3 will allow V to perform both the edge-verification and well-definition tests in a very similar way that in protocol $\Pi^{(2)}_{\text{std}}$. The commitments require V to provide a random nonzero trit for each vertex of the edge requested to a prover.

4.5.1 Distribution of questions

We now define the probability distribution $\mathcal{D}'_G$ for V's questions in protocol $\Pi^{(2)}_{\text{lhv}}[G]$ defined in the following section. It consists in one edge and two nonzero trits for each prover:

$$\mathcal{D}'_G = \{(p'(e,r,s,e',r',s'),((e,r,s),(e',r',s'))\}_{e,e'\in E,r,s,r',s'\in\mathbb{F}_3^*}$$

upon graph $G = (V, E)$ and where (e, r, s) is the question to P_1 and (e', r', s') is the question to P_2. $\mathcal{D}'_G$ is easily derived from the distribution $\mathcal{D}_G$ for the questions in $\Pi^{(2)}_{\text{std}}[G]$, as defined in section 4.4.1. First, an edge $e \in_R E$ is picked uniformly at random. Together with e, two nonzero trits $r, s \in_R \mathbb{F}_3^*$ are picked at random. Then, as in $\mathcal{D}_G$, with probability ϵ (to be selected later) the second edge $e' = e$, in which case we always set $r' = -r$ and $s' = -s$. This case allows for an edge-verification test. Finally, with probability $1 - \epsilon$, we pick e' with probability $p(e, e')|E|$ so that the couple $((e, r, s), (e', r, t))$ is produced with probability $\frac{1}{8}p(e, e')$ for all $e, e' \in E$, and $r, s, t \in \mathbb{F}_3^*$. This will allow for a well-definition test. A consequence of (1) is that for $e = (i, j) \in E$, $e' \in \mathsf{Edges}(i) \cup \mathsf{Edges}(j)$

$$p'(e,r,s,e',r,t) \geq \frac{1-\epsilon}{16|E|}\left(\frac{|\{e'\}\cap \mathsf{Edges}(i)|}{|\mathsf{Edges}(i)|} + \frac{|\{e'\}\cap \mathsf{Edges}(j)|}{|\mathsf{Edges}(j)|}\right) , \qquad (3)$$

where the inequality results from $e = e'$ being possible. According to (2), we also get

$$p'(e, r, s, e, -r, -s) = \frac{p(e, e)}{4} \geq \frac{\epsilon}{4|E|} \quad . \tag{4}$$

4.5.2 The Protocol

The protocol is similar to $\Pi^{(2)}_{\text{std}}$ except that instead of returning to V the color for each vertex of an edge in G, each prover returns commitments with implicit unveiling of these colors. If V asks two disjoint edges then V learns nothing about the values committed by the *forever-hiding* property of the commitment scheme. The resulting 2–prover one-round interactive proof system is denoted $\Pi^{(2)}_{\text{lhv}}$.

Protocol $\Pi^{(2)}_{\text{lhv}}[G]$: Two-prover, 3-COL

P_1 and P_2 pre-agree on random masks $b_i \in_R \mathbb{F}_3$ for each $i \in V$ and a random 3-coloring of G: $\{(i, c_i) | c_i \in \mathbb{F}_3\}_{i \in V}$ such that $(i, j) \in E \implies c_j \neq c_i$.

Commit phase:

- V picks $(((i, j), r, s), ((i', j'), r', s')) \in_{\mathcal{D}'_G} \left(E \times (\mathbb{F}_3^*)^2\right)^2$.
- V sends $((i, j), r, s)$ to P_1 and $((i', j'), r', s')$ to P_2.
- If $(i, j) \in E$ then P_1 replies $w_i = b_i \cdot r + c_i$ and $w_j = b_j \cdot s + c_j$.
- If $(i', j') \in E$ then P_2 replies $w'_{i'} = b_{i'} \cdot r' + c_{i'}$ and $w'_{j'} = b_{j'} \cdot s' + c_{j'}$.

Check phase:

Edge-Verification Test:

- if $(i, j) = (i', j')$ and $(r', s') \not\equiv (r, s)$ then V accept iff $w_i + w'_i \neq w_j + w'_j$.

Well-Definition Test:

- If $(i,j) = (i',j')$ and $(r',s') = (r,s)$ then V accepts iff $(w_i = w'_i) \wedge (w_j = w'_j)$.
- if $(i,j) \cap (i',j') = i$ and $r' = r$ then V accepts iff $w_i = w'_i$.
- If $(i,j) \cap (i',j') = j$ and $s' = s$ then V accepts iff $w_j = w'_j$.

Clearly, $\Pi^{(2)}_{\text{lhv}}$ satisfies perfect completeness. The following theorem establishes that in addition to perfect completeness, $\Pi^{(2)}_{\text{lhv}}$ is sound against classical provers.

Theorem 4.1. *The two-prover interactive proof system $\Pi^{(2)}_{\text{lhv}}$ is perfectly complete with classical value $\omega(\Pi^{(2)}_{\text{lhv}}[G]) \leq 1 - \frac{1}{9|E|}$ upon any graph $G = (V,E) \notin \mathsf{3COL}$.*

Proof. Assume $G \notin \mathsf{3COL}$ and let us consider the probability δ that V detects an error in the check phase when interacting with two local dishonest provers $\widetilde{\mathsf{P}}_1$ and $\widetilde{\mathsf{P}}_2$. $\Pi^{(2)}_{\text{lhv}}$ is a one-round protocol where the provers cannot communicate directly with each other nor through V's questions since they are independent of the provers' answers. It follows that the strategy of $\widetilde{\mathsf{P}}_1$ and $\widetilde{\mathsf{P}}_2$ can be made deterministic without damaging the soundness error by letting each prover choosing the answer that maximizes her/his probability of success given her/his question. Therefore, consider a deterministic strategy as a pair of arrays $W^\ell[i,r,j,s] \in \mathbb{F}_3^2$ to be used by prover $\widetilde{\mathsf{P}}_\ell$ for $\ell \in \{1,2\}$ (note: we only care about the entries where $(i,j) \in E$ upon question $((i,j),r,s)$ with $i < j$. V can always present edges in the same order)). For $z \in \{1,2\}$, $W^\ell_z[\cdot,\cdot,\cdot,\cdot]$ is the z-th component of the output pair $W^\ell[\cdot,\cdot,\cdot,\cdot]$. We say that $W[i,r]$ for $[i,r] \in E \times \mathbb{F}_3^*$ is *well defined* if for all j,k such that $(i,j),(i,k) \in E$ and $\forall s,t \in \mathbb{F}_3^*$, one of the following 4 equalities is true depending on which of $j > i$ or $j < i$, $k > i$ or $k < i$ is correct

$$W^1_1[i,r,j,s] = W^2_1[i,r,k,t] = W^1_2[j,s,i,r], \text{ or } W^1_1[i,r,j,s] = W^2_2[k,t,i,r] = W^2_2[k,t,i,r] \tag{5}$$

When $W[i,r]$ is well defined for all $i \in V, r \in \mathbb{F}_3^*$, we say that W is well defined.

We now lower bound the probability $\delta_{\text{wdt}} > 0$ that, when $W[i,r]$ is not well-defined for some

$i \in V$ and $r \in \mathbb{F}_3^*$, the well-definition test will detect it. When (5) is not satisfied , w.l.o.g. we have $W_1^1[i,r,j,s] \neq W_1^2[i,r,k,t]$ for some $(i,j),(i,k) \in E$. The other three cases are treated similarly. Let $e = (i,j)$ and $e' = (i,k)$ be these two edges. According to (3) (and (1) when $e = e'$), the well-definition test will then detect an error with probability

$$\Pr\{(\mathsf{V} \text{ picks } e \text{ and } e' \text{ with randmoness } r,s,t)\} = p'(e,r,s,e',r,t) \geq \frac{1-\epsilon}{16|E||\mathsf{Edges}(i)|} \quad . \tag{6}$$

However, we can do much better: we observe that if $W[i,r]$ is not well defined, we can detect it in at least $2|\mathsf{Edges}(i)|$ places. Consider any $\ell > i$ such that $(i,\ell) \in E$ and $u \in \mathbb{F}_3^*$ (The case where $\ell < i$ is treated similarly). It is obvious that one of the following three statements must be true:

$$W_1^1[i,r,j,s] \neq W_1^2[i,r,\ell,u],\ W_1^1[i,r,\ell,u] \neq W_1^2[i,r,\ell,u],\ \text{or } W_1^1[i,r,\ell,u] \neq W_1^2[i,r,k,t].$$

It follows that if $W[i,r]$ is not well defined then there are $2|\mathsf{Edges}(i)|$ ways for V to catch the provers and each of these has probability at least $\frac{1-\epsilon}{16|E|\cdot|\mathsf{Edges}(i)|}$ to be picked. It follows that,

$$\delta_{\text{wdt}} \geq \frac{(1-\epsilon)\cdot 2|\mathsf{Edges}(i)|}{16|E|\cdot|\mathsf{Edges}(i)|} = \frac{1-\epsilon}{8|E|} \quad .$$

Now, assume that W is well-defined, which means that the commitment values produced by the provers satisfy the consistency test. As discussed in section 4.3.3, when the commitments are consistent, the unique colors committed upon are defined by $c_i := -(W[i,r] + W[i,-r])$ for both values of r. Since $G \notin \mathsf{3COL}$, two of the vertices must be of the same color at the end-points of at least one edge $(i^*,j^*) \in E$. In this case the edge-verification test will detect it when (i^*,j^*) is the edge announced to both provers and if randomness $(r,s) \in \mathbb{F}_3^* \times \mathbb{F}_3^*$ is announced to P_1 then $(-r,-s)$ is the randomness announced to $\widetilde{\mathsf{P}}_2$. Using (4), the probability δ_{evt} to detect such an edge when W is well defined satisfies

$$\delta_{\text{evt}} \geq \sum_{r,s} \min_{e \in E}(p'(e,r,s,e,-r,-s)) \geq \frac{\epsilon}{|E|} \quad .$$

Therefore, the detection probability δ of any deterministic strategy for $G \notin \mathsf{3COL}$ satisfies

$$\delta \geq \min(\delta_{\text{wdt}}, \delta_{\text{evt}}) \geq \frac{1}{9|E|} \text{ (maximized at } \epsilon = 1/9) \ .$$

The result follows as the classical value of the game $\omega\left(\Pi^{(2)}_{\text{lhv}}[G]\right) \leq 1 - \delta$. □

To prove (perfect) zero-knowledge, it suffices to show that if $((i,j),r,s)$ and $((i',j'),r',s')$ are selected arbitrarily, V can determine at most the colors of two vertices (that form an edge). The commitments prevent a dishonest prover $\widetilde{\mathsf{V}}$ to learn the colors of two vertices that are not connected by an edge in G. Proving this is not very hard and will be done in Section 4.6.3 for the three-prover case (although with three provers, $\widetilde{\mathsf{V}}$ may also learn the color of three vertices that form a triangle). The addition of a third prover will allow, using lemma 4.1, to get soundness against entangled provers without compromising zero-knowledge. As shown in Cleve et al. (2004), their protocol is not necessarily sound against two entangled provers. We also do not know whether $\Pi^{(2)}_{\text{std}}$ is sound against two entangled provers.

4.6 Three-Prover Protocol Sound Against Entangled Provers

The three-prover protocol $\Pi^{(3)}_{\text{qnl}}$, defined below, is identical to $\Pi^{(2)}_{\text{lhv}}$ except that P_3 is asked to repeat exactly what P_1 or P_2 has replied. The prover that P_3 is asked to emulate is picked at random by V. An application of lemma 4.1 allows to conclude the soundness of $\Pi^{(3)}_{\text{qnl}}$ against entangled provers. Zero-knowledge remains since the only way to provide V with the colors of more than two connected vertices is if they form a complete triangle of G. This reveals nothing beyond the fact that $G \in \mathsf{3COL}$ to V, since all vertices will then show different colors.

4.6.1 Distribution of questions

The probability distribution $\widehat{\mathcal{D}}_G$ for V's questions to the three provers is easily obtained from the distribution $\mathcal{D}'_G$ for the questions in protocol $\Pi^{(2)}_{\text{lhv}}[G]$. V picks $((e,r,s),(e',r',s')) \in_{\mathcal{D}'_G} \left(E \times (\mathbb{F}_3^*)^2\right)^2$ and sets $\hat{e} = e$, $\hat{r} = r$, and $\hat{s} = s$ with probability $\frac{1}{2}$ or sets $\hat{e} = e'$, $\hat{r} = r'$, and $\hat{s} = s'$ also with probability $\frac{1}{2}$. Defined that way, $\widehat{\mathcal{D}}_G$ is a properly defined probability distribution for V's three

questions, each one in $E \times (\mathbb{F}_3^*)^2$.

4.6.2 The Protocol

In protocol $\Pi_{\text{qnl}}^{(3)}$, after the three questions picked according $\widehat{\mathcal{D}}_G$ by V have been answered by the the provers, V accepts if and only if the replies of P_1 and P_2 are accepted in $\Pi_{\text{lhv}}^{(2)}$ and in addition, P_3 gave the same reply than the prover it emulates.

Protocol $\Pi_{\text{qnl}}^{(3)}[G]$: Three-prover, 3-COL.

Provers $\mathsf{P}_1, \mathsf{P}_2$, and P_3 pre-agree on random values $b_i \in_R \mathbb{F}_3$ for all $i \in V$ and a random 3-coloring of G: $\{(i, c_i) | c_i \in \{0,1,2\}\}_{i \in V}$ such that $(i,j) \in E \implies c_j \neq c_i$.

Commit phase:

- V picks $(((i,j), r, s), ((i',j'), r', s'), ((\hat{\imath}, \hat{\jmath}), \hat{r}, \hat{s})) \in_{\widehat{\mathcal{D}}_G} \left(E \times (\mathbb{F}_3^*)^2\right)^3$.
- V sends $((i,j), r, s)$ to P_1, $((i',j'), r', s')$ to P_2, and $((\hat{\imath}, \hat{\jmath}), \hat{r}, \hat{s})$ to P_3.
- If $(i,j) \in E$ then P_1 replies $w_i = b_i \cdot r + c_i$ and $w_j = b_j \cdot s + c_j$.
- If $(i',j') \in E$ then P_2 replies $w'_{i'} = b_{i'} \cdot r' + c_{i'}$ and $w'_{j'} = b_{j'} \cdot s' + c_{j'}$.
- If $(\hat{\imath}, \hat{\jmath}) \in E$ then P_3 replies $\hat{w}_{\hat{\imath}} = b_{\hat{\imath}} \cdot \hat{r} + c_{\hat{\imath}}$ and $\hat{w}_{\hat{\jmath}} = b_{\hat{\jmath}} \cdot \hat{s} + c_{\hat{\jmath}}$.

Check phase:

Consistency Test:

- If $((\hat{\imath}, \hat{\jmath}), \hat{r}, \hat{s}) = ((i,j), r, s)$ then V rejects if $(w_i, w_j) \neq (\hat{w}_{\hat{\imath}}, \hat{w}_{\hat{\jmath}})$.
- If $((\hat{\imath}, \hat{\jmath}), \hat{r}, \hat{s}) = ((i',j'), r', s')$ then V rejects if $(w'_{i'}, w'_{j'}) \neq (\hat{w}_{\hat{\imath}}, \hat{w}_{\hat{\jmath}})$.

Edge-Verification Test:

- if $(i,j) = (i',j')$ and $(r',s') \neq (r,s)$ then V accept iff $w_i + w'_i \neq w_j + w'_j$.

Well-Definition Test:

- If $(i,j) = (i',j')$ and $(r',s') = (r,s)$ then V accepts iff $(w_i = w'_i) \wedge (w_j = w'_j)$.
- if $(i,j) \cap (i',j') = i$ and $r = r'$ then V accepts iff $w_i = w'_i$.
- If $(i,j) \cap (i',j') = j$ and $s = s'$ then V accepts iff $w_j = w'_j$.

The soundness of protocol $\Pi^{(3)}_{\text{qnl}}$ against entangled provers can easily be shown a direct consequence of the soundness of protocol $\Pi^{(2)}_{\text{lhv}}$ against classical provers, by an application of Lemma 4.1. Indeed, the soundness error corresponds to the quantum value of the game when $G = (V,E) \notin \mathsf{3COL}$ provided $\Pi^{(2)}_{\text{lhv}}$ is symmetric. As defined in Sect. 4.5.1 however, the distribution of questions $\mathcal{D}'_G$ is not necessarily symmetric since the first edge e is picked uniformly at random in E while the second edge $e' \in E$ is picked from e in a way that the marginal may not be uniform. However, $\Pi^{(2)}_{\text{lhv}}$ can easily be turned into a symmetric protocol by picking $(e,r,s),(e',r',s')$ according $\mathcal{D}'_G$ and announcing (e,r,s) to P_1 and (e',r',s') to P_2 with probability $\frac{1}{2}$ while announcing (e,r,s) to P_2 and (e',r',s') to P_1 with probability $\frac{1}{2}$. The resulting symmetric protocol is equivalent to $\Pi^{(2)}_{\text{lhv}}$ and therefore shares its classical value upper bounded in Theorem 4.1 and the set of questions Q to each player remains the same as for $\Pi^{(2)}_{\text{lhv}}$. In the symmetric version, Q is thus the same for every prover and $|Q| = 4|E|$.

Theorem 4.2. *The three-prover interactive proof system $\Pi^{(3)}_{\text{qnl}}$ is perfectly complete and has quantum value*

$$\omega^*\left(\Pi^{(3)}_{\text{qnl}}[G]\right) \leq 1 - \left(\frac{1}{9|E| + 432|E|^2}\right)^2 \leq 1 - \left(\frac{1}{21|E|}\right)^4 \tag{7}$$

upon any graph $G = (V,E) \notin \mathsf{3COL}$.

Proof. Assume $G = (V,E) \notin \mathsf{3COL}$. The contrapositive of Lemma 4.1 indicates any one-round symmetric game $\Pi^{(2)}_{\text{lhv}}[G]$ with classical value $\omega\left(\Pi^{(2)}_{\text{lhv}}[G]\right) \leq 1 - \delta - 12|Q|\sqrt{\delta}$ is such that the modified game $\Pi^{(3)}_{\text{qnl}}[G]$ has quantum value $\omega^*\left(\Pi^{(3)}_{\text{qnl}}[G]\right) \leq 1 - \delta$. The set Q of questions to each

player satisfies $|Q| = 4|E|$. Theorem 4.1 establishes that $\delta + 12|Q|\sqrt{\delta} \geq \frac{1}{9|E|}$, which implies $\sqrt{\delta} \geq \frac{1}{\left(\sqrt{\delta}+12|Q|\right)\cdot 9|E|} \geq \frac{1}{(1+12|Q|)\cdot 9|E|} = \frac{1}{9|E|+432|E|^2} \geq \frac{1}{441|E|^2}$, and the result follows. □

As an immediate consequence of Theorem 4.2, $\Omega(|E|^4)$ sequential repetitions of $\Pi^{(3)}_{\text{qnl}}$ produces an interactive proof system for 3COL with negligible soundness error. Although the resulting proof system can be implemented on short distances, these many sequential rounds need to be performed at high rate for a given proof to be concluded in reasonable time. A few executions of $\Pi^{(3)}_{\text{qnl}}$ could be ran in parallel without having to greatly increase the distances while reducing the number of sequential rounds. However, we don't know how the soundness error decreases when $\Pi^{(3)}_{\text{qnl}}$ is ran only a few times in parallel, even though the results of Kempe and Vidick, a quantum version of Raz's parallel repetition theorem [Raz (1998)], indicate that $\Omega(|E|^4)$ runs in parallel produces a proof system with negligible soundness error [Kempe and Vidick (2011)].

4.6.3 Proof of Perfect Zero-Knowledge

In this section, we prove that protocol $\Pi^{(3)}_{\text{qnl}}$ is perfect zero-knowledge. As a consequence, $\Pi^{(2)}_{\text{lhv}}$ is also zero-knowledge since everything $\widetilde{\mathsf{V}}$ sees in $\Pi^{(2)}_{\text{lhv}}$ can also be observed in $\Pi^{(3)}_{\text{qnl}}$. The proof of zero-knowledge proceeds using the fact that a vertex must appear at least twice to have its color unveiled. This is the *forever hiding property* of the commitment scheme described in Section 4.3.3. Notice that this would be enough for $\widetilde{\mathsf{V}}$ to learn something about the coloring if no extra condition on these three vertices is observed. In fact, we can easily show that only a few cases of color disclosure are possible and in each of these cases, $\widetilde{\mathsf{V}}$ learns nothing about the coloring that it could not have computed on its own. $\widetilde{\mathsf{V}}$ can only learn the color of two connected vertices in G and nothing else or the colors of three vertices forming a triangle in G. In each of these cases, $\widetilde{\mathsf{V}}$ learns random distinct colors for these vertices, which is to be expected by a valid 3-coloring of G. Let us show why this is enforced by the properties (see Section 4.3.3) of the commitment scheme. Remember that in order to learn the color assigned to a vertex $i \in V$, $\widetilde{\mathsf{V}}$ must ask that vertex to at least 2 distinct provers. Otherwise, $\widetilde{\mathsf{V}}$ sees only random values returned by the provers. There are 7 cases of figure depending on how $\widetilde{\mathsf{V}}$ selects the 3 edges asked. Figure 4.3 shows all cases. The 3 edges indicated for each case are the one picked by $\widetilde{\mathsf{V}}$. The colors associated to white vertices remain hidden by

the forever hiding property of the commitment scheme. For these vertices, the committed values received from the provers are just random and independent elements in $\mathbb{F}_3$. In each of the 7 cases, the unveiled colors of the vertices are displayed in shade of grey. We see that the only way to unveil the color of two vertices (cases 2, 3, 4, 5, and 6) is when they are connected by an edge, which means that the colors of both vertices are random but distinct. The only way for $\widetilde{\mathsf{V}}$ to learn the color of 3 distinct vertices is when they form a triangle (case 7). In this case, $\widetilde{\mathsf{V}}$ learns three random and distinct colors. Clearly, this is nothing more than something necessarily true when $G \in \mathsf{3COL}$.

These properties of the commitment scheme allow, for any quantum polynomial-time dishonest verifier $\widetilde{\mathsf{V}}$, an easy simulator for $\mathbf{view}(\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_3, \widetilde{\mathsf{V}}, G)$ when $G \in \mathsf{3COL}$, thus establishing that $\Pi_{\text{qnl}}^{(3)}$ is perfect zero-knowledge.

Theorem 4.3. *The three-prover interactive proof system $\Pi_{\text{qnl}}^{(3)}$ is perfect zero-knowledge against quantum verifiers.*

Proof. The simulator Sim is classical given blackbox access to $\widetilde{\mathsf{V}}$ (and $\widetilde{\mathsf{V}}$ can be quantum).

Consider an execution $\mathsf{Sim}(G)$ upon graph $G = (V, E)$. It first picks a random permutation $\text{COL}[\cdot] : \mathbb{F}_3 \mapsto \mathbb{F}_3$ over three colors, each corresponding to a distinct element in $\mathbb{F}_3$. Table $\text{MARK}[i, r] \in \{\mathsf{True}, \mathsf{False}\}$, for $i \in V$ and $r \in \mathbb{F}_3^*$, is initialized to False and will indicate if the output of a prover has already been simulated for vertex i with randomness r. Table $\text{COUNT}[i]$, for $i \in V$, counts the number of times vertex i has been asked so far during the simulation. Variable $c \in \mathbb{F}_3$, initialized to 0, indicates the next color index the simulator should use when a new color must be unveiled during the simulation.

Simulator $\mathsf{Sim}(G)$: Simulator for $\widetilde{\mathsf{V}}$'s view upon graph G in $\Pi_{\text{qnl}}^{(3)}$.

All arithmetic below is performed in $\mathbb{F}_3$.

(1) Let $\text{COL}[\cdot]$ be a uniform permutation of $\mathbb{F}_3$ and let $c := 0$.

(2) $\forall i \in V, \forall r \in \mathbb{F}_3^*$, let $\text{MARK}[i, r] := \mathsf{False}$ and $\text{COUNT}[i] := 0$.

(3) Run $\widetilde{\mathsf{V}}$ until it returns $((i_1, j_1), r_1, s_1)$, $((i_2, j_2), r_2, s_2)$, $((i_3, j_3), r_3, s_3)$.

(4) For each $\ell \in \{1, 2, 3\}$ do:

- Whenever $(i_\ell, j_\ell) \in E$ is provided by $\widetilde{\mathsf{V}}$, output $(w^\ell_{i_\ell}, w^\ell_{j_\ell}) \in \mathbb{F}_3 \times \mathbb{F}_3$ to $\widetilde{\mathsf{V}}$, both computed as follows:

  (a) If $\neg\textsc{mark}[i_\ell, r_\ell]$ then

  - If $\textsc{count}[i_\ell] = 0$ then pick $W[i_\ell, r_\ell] \in_R \mathbb{F}_3$.
  - If $\textsc{count}[i_\ell] = 1$ then set $W[i_\ell, r_\ell] := -\textsc{col}[c] - W[i_\ell, -r_\ell]$, $c := c + 1$.
  - $\textsc{count}[i_\ell] := \textsc{count}[i_\ell] + 1$.

  (b) If $\neg\textsc{mark}[j_\ell, s_\ell]$ then

  - If $\textsc{count}[j_\ell] = 0$ then pick $W[j_\ell, s_\ell] \in_R \mathbb{F}_3$.
  - If $\textsc{count}[j_\ell] = 1$ then set $W[j_\ell, s_\ell] := -\textsc{col}[c] - W[j_\ell, -s_\ell]$, $c := c + 1$.
  - $\textsc{count}[j_\ell] := \textsc{count}[j_\ell] + 1$.

  (c) $\textsc{mark}[i_\ell, r_\ell] :=$ True, $\textsc{mark}[j_\ell, s_\ell] :=$ True.

  (d) $w^\ell_{i_\ell} := W[i_\ell, r_\ell]$, $w^\ell_{j_\ell} := W[j_\ell, s_\ell]$.

$\widetilde{\mathsf{V}}$ is then invoked to produce questions $((i_\ell, j_\ell), r_\ell, s_\ell)$ for all provers $\mathsf{P}_\ell, \ell \in \{1, 2, 3\}$. Sim now aims at setting the values $(w^\ell_{i_\ell}, w^\ell_{j_\ell})$ for P_ℓ's commitments. If $(i_\ell, j_\ell) \notin E$, Sim produces no value for $(w^\ell_{i_\ell}, w^\ell_{j_\ell})$, exactly as P_ℓ in $\Pi^{(3)}_{\text{qnl}}$.

When $(i_\ell, j_\ell) \in E$, Sim first produces P_ℓ's commitment $w^\ell_{i_\ell}$ for $i_\ell \in V$ and then produces P_ℓ's

commitment $w^{\ell}_{j_\ell}$ for $j_\ell \in V$. We show how $w^{\ell}_{i_\ell}$, $w^{\ell}_{j_\ell}$ is computed similarly mutatis mutandis:

- if $\text{MARK}[i_\ell, r_\ell]$ then Sim returns the value of $w^{\ell}_{i_\ell}$ already determined for the simulation of the commitment of an *earlier* prover P_h, $h < \ell$. This ensures that both the commitment's *consistency test* performed and the well-definition test are always successful, as in $\Pi^{(3)}_{\text{qnl}}$ with honest provers.

- if $\neg\text{MARK}[i_\ell, r_\ell]$ then Sim has never simulated a commitment of the color for vertex i_ℓ with randomness r_ℓ. The value $\text{COUNT}[i_\ell]$ indicates the number of times prior to this value for ℓ, vertex i_ℓ has been asked:

  - If $\text{COUNT}[i_\ell] = 0$ then $w^{\ell}_{i_\ell} \in_R \mathbb{F}_3$ is picked uniformly at random, as it should be when the commitment value for the color of vertex i_ℓ is observed in isolation.

  - If $\text{COUNT}[i_\ell] = 1$ then the color associated to vertex i_ℓ has been committed to value $w^{h}_{i_\ell}$ by an *earlier* simulated prover P_h, $h < \ell$ upon randomness $-r_\ell$ (otherwise, $\text{MARK}[i_\ell, r_\ell] =$ True). Sim sets $w^{\ell}_{i_\ell} = -\text{COL}[c] - w^{h}_{i_\ell}$, which satisfies the *implicit unveiling* of random color $\text{COL}[c] = -w^{\ell}_{i_\ell} - w^{h}_{i_\ell}$. The current color c is incremented.

  The value of $\text{COUNT}[i_\ell]$ is increased by one and $\text{MARK}[i_\ell, r_\ell] =$ True, as the color of vertex i_ℓ with randomness r_ℓ has been committed upon by the simulated prover P_ℓ.

Let $(w^1_{i_1}, w^1_{j_1})$, $(w^2_{i_2}, w^2_{j_2})$, and $(w^3_{i_3}, w^3_{j_3})$ be all commitment values simulated by Sim. As discussed above and shown in Fig. 4.3, the colors of no more than 3 vertices are unveiled in the process. Sim always unveils as many different colors as there are colors unveiled to $\widetilde{\mathsf{V}}$. If Sim's simulated committed values unveils only the color of one vertex then that color is random, as it should in this case in $\Pi^{(3)}_{\text{qnl}}$.

If Sim's committed values unveils the colors of exactly 2 vertices then these 2 vertices form an edge in G and the colors are two different random colors, as it should be in $\Pi^{(3)}_{\text{qnl}}$. Finally, when Sim's committed values unveil the colors of exactly 3 vertices then these vertices form a triangle in G. The 3 colors unveiled by Sim to $\widetilde{\mathsf{V}}$ are different and assigned randomly to each of the 3 vertices, as

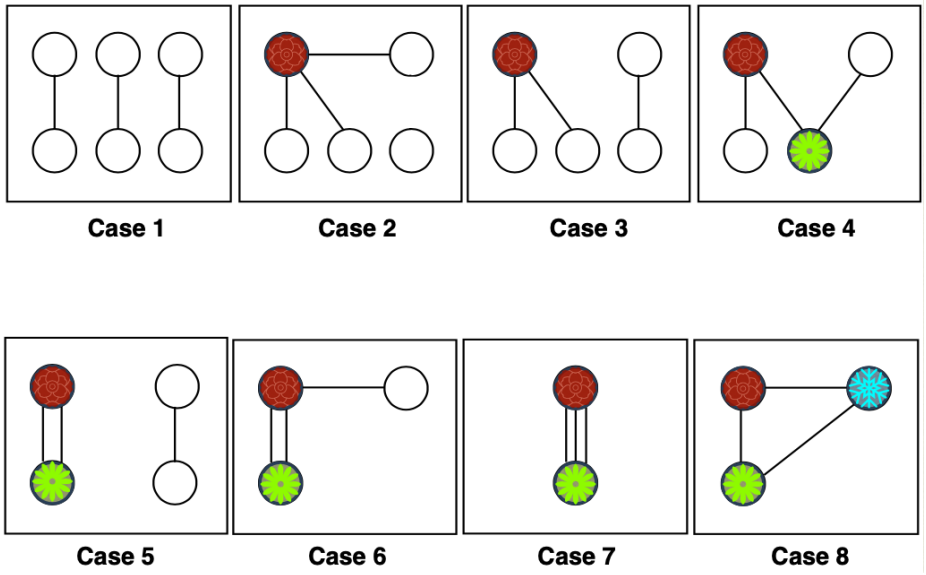


Figure 4.3: The 7 ways to unveil the colors of at most 3 vertices in $\Pi^{(3)}_{\text{qnl}}$.

it is in $\Pi^{(3)}_{\text{qnl}}$. Otherwise, if w_i^ℓ for $i \in V$ has been generated with only one random value then w_i^ℓ is random and uniform in $\mathbb{F}_3$, exactly as it is in $\Pi^{(3)}_{\text{qnl}}$ in the same situation. It is now clear that,

$$\mathbf{view}(\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_3, \widetilde{\mathsf{V}}, G) = \mathsf{Sim}(G) \ ,$$

and $\Pi^{(3)}_{\text{qnl}}$ is perfect zero-knowledge. □

Note: since no *rewinding* is used by our simulator, it is absolutely unnecessary to explicitly handle *auxiliary-inputs* or the fact that V is quantum. No special care is required to handle these considerations that become highly non-trivial in the case where rewinding is required.

4.7 Conclusion and Open Problems

We have provided a three-prover perfect zero-knowledge proof system for **NP** sound against entangled provers that is implementable in some well controlled environment. In order to make it fully practical, it would be better to find a protocol with smaller soundness error and requiring only two provers.

Our protocols are *proofs of membership* whereas in practice we would like to use them for identification purpose in which scenario *proofs of knowledge* is what we really need.

Moreover, we would like to extend our techniques to prove any language in $\mathbf{QMA}$ or $\mathbf{QCMA}$, the natural quantum extensions of $\mathbf{NP}$.

We would also want to prove whether $\Pi_{\text{std}}^{(2)}$ is sound against entangled provers. Finally, we seek a variant of $\Pi_{\text{std}}^{(2)}$ that would be sound against no-signaling provers and variants of $\Pi_{\text{lhv}}^{(2)}$ and $\Pi_{\text{qnl}}^{(3)}$ that are both sound against no-signaling provers and zero-knowledge.

Chapter 5

Distributed Trust and Non-Locality

5.1 Trading (Non-)Locality for (Dis-)Trust

The connection between non-locality and electronic voting is a small but surprising one. A key component of the construction is the *committed voter roster*. The commitment scheme used (although we state that it can be interchanged) is a CHSH-style commitment. However, instead of assuming that two provers are local, we assume instead that the provers are *adversarial*. That is, the binding and hiding conditions hold not because provers cannot signal (or use a non-local strategy), but because it is not in their interest to break them.

As an example of this adversarial assumption, consider a cryptocurrency (like Bitcoin) and its miners. The payoff to successfully mining a block is winner-takes-all. Thus, miners are adversarial and mutually-distrusting. Yet, it is not in the majority's interest to deny the successful mining of any particular miner, since it would mean that no one ever wins. In elections, where commitments are shared across institutions, a similar equilibrium occurs, where adversarial assumptions keep the commitments hidden and bound.

We will leave the definition of *mutually-distrusting* and *adversarial* informal, as the commitment

This chapter is adapted from a paper which was jointly authored by Nan Yang and Jeremy Clark. It was published in the Financial Cryptography 2017 Voting Workshop [Yang and Clark (2017)].

itself, instantiated in this manner, was not the focus of our paper. A formal game-theoretic treatment of the link between locality and trust is left as an open problem.

This chapter answers the question which was raised in the introduction, "Can the techniques we develop along the way be used in practice?"

Protocol 5.1. *Mutually-Distrusting Commitment (Example)*

We will call the provers "institutions" and the verifier a "voter". We assume that there exists an authenticated broadcast channel over which everyone can communicate, and each two parties have pairwise authenticated, encrypted channels.

3 mutually-distrusting, adversarial institutions A, B, and C. Voter V.

The institutions share a random bitstring w.

A wishes to commit to a bit b during an interaction with V.

V sends A a random bitstring r. A broadcasts $w \oplus b \cdot r$.

When it is time to unveil, B and C broadcast w.

b can then be recovered publicly.

5.2 Background

An end-to-end verifiable (E2E) voting system uses cryptography to provide a verifiable tally while maintaining the secrecy of each voter's ballot. Over decades of research in this area, one trend to emerge is a move toward real-world voting systems suitable for common election scenarios, including governmental elections. For our purposes, we consider a system to be suitable for a governmental election if it has two properties:

(1) **Vote-and-go**: once a voter has completed and submitted their ballot, they do not need to be involved in the tallying process.

(2) **Human-votable**: a voter can cast a vote without having to perform any computations (bare-handed) through a process similar to a traditional (non-verifiable) voting system, such as DRE or optical scan voting

Many E2E systems are designed within these constraints and some have been used in governmental elections [Burton, Culnane, and Schneider (2016); Carback et al. (2010)]. The governmental setting is contrasted with other practical settings, such as a boardroom vote, where all voters might be physically present in the same room with their own trusted computational devices. This setting is less constrained and allows different cryptographic techniques to be used — e.g., an unconditionally secure multiparty computation.

In the governmental setting, vote-and-go requires a third party election authority to collect a representation of the voter's ballot. This representation is often an encryption or commitment to the voter selections for DRE-based systems, or for optical scan systems, a paper-based obfuscation (e.g., code substitution, permutation, split) that is accompanied by some encryption or commitment value on the ballot or in the backend data. Standard encryption and commitment schemes are not secure against a computationally unbounded adversary. Such an adversary can either recover the message (Elgamal or Paillier), change the message (Pedersen commitment) or both (hash-based commitments). When the message is a vote, this translates into, respectively, breaking election integrity or ballot secrecy or both.

Computational assumptions underly nearly all real-world cryptographic applications, whether it is HTTPS, password hashing, or secure messaging. However the exact assumptions evolve over time as new attacks are found, as do the security parameters that realize them. An unconditionally secure protocol alleviates us from monitoring the validity of these assumptions over time and future-proofs the protocol against new innovations like quantum computing.

Category	*Examples*	*Secrecy*		*Integrity*		
				Resilient to Unbounded Adversary	Resilient to Full Collusion	No Special Assumptions
Distributed EA	Pret a Voter, Helios		•	•	•	•
Chaumian	Punchscan, Scantegrity			•	•	•
Everlasting Privacy	Moran-Naor	•	•		•	•
Boardroom	Broadbent-Tapp	•	•	•		•
This work		•		•		•

Table 5.1: A comparison of computational and collusion security assumptions in four common categories of proposed cryptographic voting systems, plus our own system. Note: this table does not attempt to capture all desirable features of a voting system. We acheive the same security assumptions as boardroom voting systems, plus we allow human-voteable ballots and vote-and-go tallying. The 'special assumption' used by Chaumian and this work, for privacy, both correspond to a blackbox described below.

5.3 Prior Work

There are hundreds of papers proposing voting schemes and it is not possible to review even all the relevant ones. Instead, we have broken the literature into four broad categories that classify a majority of the proposals. Table 1 provides a summary of the election integrity and ballot secrecy assumptions for each cluster.

Distributed EA. Beginning with Cramer, Gennaro, and Schoenmakers (1997), many systems homomorphically encrypt ballots under a public key that is distributed amongst a set of trustees forming an election authority (EA). If an unbounded adversary attacks a transcript of the election, they can learn how every voter voted by breaking the encryption key but cannot change the value that is encrypted. Further, assuming true zero knowledge proofs are used, unbounded adversaries cannot undetectably change the tally. Note that in practice, many of these systems use non-interactive zero knowledge proofs based on the Fiat-Shamir heuristic — this enables an unbounded adversary

(whether a voter or a trustee) to lie [citeGK03] in a way that can undetectably change a tally, however this assumption is practical to avoid [Gallegos-Garcia, Iovino, Rial, Ronne, and Ryan (2016); Kiayias, Zacharias, and Zhang (2015)]. If a suitable threshold of trustees are corrupted, they may recover how each voter voted but they cannot change the tally. A few notable systems of this type include: MarkPledge [Neff (2001)], Prêt à Voter [Chaum, Ryan, and Schneider (2005)], Voter-initiated auditing [Benaloh (2006)], Helios [Adida (2008)], STAR-Vote [S. Bell et al. (2013)], and vVote [Burton et al. (2016)].

Chaumian. Beginning with Chaum (2004), a series of systems also use a distributed election authority much like above. However these systems add an additional assumption: trustees can use a special computational device, called a blackbox, to perform computations such that the inputs and intermediate values are not leaked to any participant. This enables an election system based solely on cryptographic commitments and commitment-based cut-and-choose proofs. Assuming the commitment scheme is perfectly binding, an adversary can break ballot secrecy by breaking the commitment scheme (if unbounded), corrupting a sufficient number of trustees to recover the input to the blackbox, or by breaking the blackbox hardware assumption. However an unbounded adversary cannot undetectably change the values committed to, all modifications to the tally are detectable even if made by a fully colluding election authority, and the soundness of the blackbox computations are verifiable and not assumed to be done correctly. Notable systems of this type include Punchscan [Popoveniuc and Hosp (2006)], Scantegrity I/II [Chaum, Carback, et al. (2008); Chaum, Essex, et al. (2008)], Eperio [Essex, Clark, Hengartner, and Adams (2010)], and Remotegrity [Zagórski et al. (2013)].

Everlasting Privacy. Beginning with Cramer, Franklin, Schoenmakers, and Yung (1996) (and related to earlier work in Chaum (1988)), a reasonable observation was made that integrity need only last the lifetime of the election but ballot secrecy could be relevant for decades or centuries. It is possible to invert the resistance of a voting scheme to computationally unbounded adversaries from integrity to privacy. Most modern work uses perfectly hiding homomorphic commitments in lieu of homomorphic encryption, however this creates a dilemma: if the random factors of the

commitments are unknown, a tally cannot be computed (and if they are known, then the commitment's hiding property no longer resists an unbounded adversary). Most systems compromise by using untappable channels to communicate random factors amongst trustees— thus it does not retain unconditional ballot secrecy under collusion. Notable systems of this type include Moran and Naor (2006), split-ballot voting [Moran and Naor (2007)], and extensions to distributed EA systems [Demirel, van de Graaf, and dos Santos Araujo (2012)].

Boardroom Voting. The term boardroom voting was suggested by Benoloh and Fisher [Benaloh (*né* Cohen) and Fisher (1985)] to categorize systems where voters participate in the tallying process (i.e., are not vote-and-go). Like the general literature on unconditionally secure protocols, these schemes tend to use multiparty computation based on verifiable secret sharing. Note that not all boardroom voting schemes are unconditional — many boardroom systems use computational assumptions to be more practical [Hao and Zieliński (2009); Kiayias and Yung (2002); Schoenmakers (1999, 2000)]. However the ones that are resist unbounded adversaries for both integrity and privacy (but collusion between them can break either property). One way to frame our contribution is porting the security properties of these systems to a governmental election. This has been explored by Broadbent and Tapp (2008) and the *vote-and-go* property is achieved, voters need to perform computations in the booth (and it is thus not *human voteable*). One might argue that ThreeBallot [Rivest and Smith (2007)] is a human-voteable instantiation of secret sharing and its properties are very close to what we want to achieve. Unfortunately ThreeBallot is not fully private [Henry, Stinson, and Sui (2009)].

5.4 Framing our Contribution

It has long been asserted within our community that perfect ballot secrecy and perfect election integrity cannot be simultaneously achieved. This trade-off is quite true under certain assumptions but it is often repeated as a simple fact without internalizing the fine print. As it turns out, if you read the fine print, it is possible to achieve both — indeed many boardroom voting systems already do. The challenge is achieving these security properties while also allowing the voter to deposit their ballot with the EA and leave. If the deposited ballot is an encryption or computational commitment, it must be either computationally binding or hiding but not both. If the ballot is secret

shared to the trustees, however, it can be perfectly hiding and binding under an assumption about the number of honest trustees. The immediate difficulty here is that secret sharing a vote will require a computational device.

This paper is intended as exploratory research to understand better how far unconditional privacy and integrity can be extended to a practical governmental voting system. We are not insisting that our system is immediately better than existing approaches because we require certain trade-offs that might be less desirable (discussed below). However we think this area deserves exploration.

In our approach, we begin in the Chaumian model. We noted in our literature review that systems in this model primarily rely on a commitment scheme. As we discuss in Section 5.5.1, verifiable secret sharing can be used as a perfectly hiding commitment that is also perfectly binding but only to the participants in the secret sharing scheme. We take a simple system from this model, Eperio [Essex et al. (2010)], which is already just a backend tallying system that can interface with a variety of paper ballots (permutation-based ballots like Prêt à Voter and code-based ballots like Scantegrity), and we replace the commitment scheme with a protocol based on verifiable secret sharing. We then show that the cut-and-choose protocols continue to provide election integrity, assuming an honest threshold of trustees (which is already assumed in computational Eperio for ballot privacy). The result is an interesting protocol that achieves unconditional privacy and integrity, plus voters can vote with paper ballots.

Universal verification. We pay a price for unconditional secrecy and privacy, namely we have to sacrifice universal verification. We are unaware of a proof that universal verification is impossible to achieve in the unconditional model, but we do note that attempts of adding it to the basic primitive we use (VSS) generally has only been achieved with computationally secure primitives [Schoenmakers (1999); Stadler (1996)]. In our protocol, voters can still perform the traditional *cast-as-intended* and *recorded-as-cast* checks but voters have to trust that a threshold of trustees are honest in reporting that ballots were *tallied-as-recorded*. It is not clear this trade-off is worth the gain in security against unbounded adversaries, but we will say that it is not that different from

cryptographic election where voters defer to others (say each political party) to perform the cryptographic election audit of the tally. Finally, our approach of using paper ballots does not preclude traditional risk-limiting manual recounts done in conjunction with the cryptographic election if the ballots have a cryptographic overlay (as in Scantegrity II).

Blackbox assumption. Finally, like Punchscan, Scantegrity and Eperio, we do make a blackbox assumption that a perfectly private computation can be performed on a tamper-resistant device. Blackboxes are stateless devices without any non-volatile memory. The simply compute an output from a set of inputs without revealing any intermediary values in the function. They could be implemented as a hardware circuit, FPGA, or in software in a trusted execution environment such as Intel TXT (c.f., Mannan, Kim, Ganjali, and Lie (2011)).

Future work might explore the removal of this assumption, through a distributed computation, however we rely on it for this initial work in the area. We do note however that it is not immediately clear that a distributed computation is necessarily better. If an adversary wanted to attack the election by corrupting computational devices, it seems logical that compromising n devices is harder than compromising 1—in fact, this reasoning is seductive enough that the shareholders might use standard computers without extra precautions to perform their computations. In such case, compromising n devices might be as easy as compromising one (e.g., through an exploit for a common operating system) and might indeed be easier if the single blackbox device (it does not even need to be a full fledged computer) is given a lot of attention in terms of hardening it against attack.

Human-voteable & vote-and-go. Some voting schemes require the voter to participate in some multi-party computation. For example, Broadbent and Tapp (2008) requires that voters take their vote and secret-share it with different election authorities. Even Malkhi, Margo, and Pavlov (2002), a voting scheme "without cryptography," requires the voter to perform an amount of arithmetic which is arguably unreasonable in practice. In contrast, a human-voteable (also called barehanded [Riva and Ta-Shma (2007)]) voting scheme is one which does not require any kind of computational device to vote (such as a trusted computer).[1] Vote-and-go refers to the fact that individual voters

[1]Note we do not refer to assistive technology (AT) that helps voters with disabilities cast a vote—for this reason, we dislike the term barehanded. Rather we mean devices that are trusted to perform a computation for the voter, not navigate

are not expected to assist in any kind of post-ballot computations, such as computing the tally. All major governmental elections today are have both properties. It is difficult to see how a scheme that does not have both can escape being an impractical academic exercise. While smartphones are ubiquitous, their use opens up new attack vectors and is no better than trusting a polling machine or a physical ballot counted by humans.

5.5 Protocol Components

5.5.1 Verifiable Secret-Sharing and Commitment

A (k, n) verifiable secret-sharing (VSS) scheme is a multi-party protocol between a *dealer* and n *shareholders* that consists of two functions $\langle \mathsf{Share}, \mathsf{Recover} \rangle$. When invoking share, the dealer distributes some secret string x among the shareholders such that no subset of shareholders less than k can jointly output x and the dealer proves that each share can be consistently used to reconstruct some secret without an error. When invoking Recover, k or more shareholders combine their shares to recover x (if less than k shareholders honestly contribute their shares, $\perp$ is recovered instead).

The guarantees of a VSS scheme can be made information-theoretic while tolerating up to $k < n/2$ malicious shareholders, assuming the existence of a broadcast channel. A broadcast channel is already a standard assumption in an E2E voting scheme. Many VSS schemes exist, each targeting different efficiency metrics. For our purposes, we assume the use of a standard scheme due to Rabin and Ben-Or (1989).

The relationship between a VSS scheme and a commitment function was explored recently by Garay, Givens, Ostrovsky, and Raykov (2014). They observe that VSS is typically used a distributed 'analogue' to a commitment scheme and prove that VSS realizes a commitment-like properties. Informally speaking, the two main properties of bit-commitment are binding and hiding, which respectively mean that the sender can only open the commitment in one way, and that the receiver is unable to distinguish between (chosen) committed messages m_0 or m_1.

an interface.

The respective properties of VSS which will act as the binding and hiding conditions are:

- If no strict majority of shareholder's shares uniquely defines a secret, then there will be an abort. In other words, the dealer is unable to either create a commitment that they cannot open, or a commitment that can be opened in more than one way.

- No strict minority subset of shareholders can reconstruct the secret, or prevent an honest strict majority from reconstructing the secret. If a secret fails to be reconstructed, then the faulty shares can be identified. In other words, no strict minority subset of colluding sShareholders can change an existing commitment, or prevent the honest shareholders from opening the commitment.

- The secret will only be reconstructed when the majority of honest shareholders come to an agreement. In contrast to a two-party bit-commitment, the dealer is not involved in the opening process. Some pre-agreed condition will trigger the honest shareholders to divulge their shares. In our case, they are triggered by an auditor.

Concretely, given a (k, n)-VSS scheme, our commitment scheme will consist of two function $\langle \mathsf{Commit}, \mathsf{Open} \rangle$ realized as follows.

- $\mathsf{Commit}(x)$: The dealer takes a secret x and invokes $\mathsf{Share}(x)$ with the shareholders and proves that the shares are consistent. A failure of the secret-sharing is considered a failure of commitment. If successful, the dealer announces a commitment identifier id to the shareholders used to identify the commitment that should be opened. This identity is output as commitment value c (in a standard commitment, c would be functionally dependent on x).

- $\mathsf{Open}(c)$: The auditor sets $id = c$ broadcasts to the shareholders $\mathsf{Recover}(id)$. The honest shareholders follow the protocol to determine if the commitment should be opened or not. If so, they execute the reconstruct protocol and send to the auditor their shares, who reconstructs the secret. The honest majority will identify any dishonest shareholders, whose shares the auditors will ignore.

5.5.2 Eperio

Our voting protocol is based on the Eperio voting system [Essex et al. (2010)]. Technically Eperio is a backend component that can realizes a variety of voting systems. We summarize some details of that protocol which we will augment with VSS in Section 5.6.

Ballots.

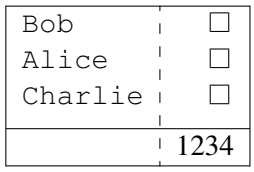
Eperio can utilize different ballot types. We use a ballot in the style of Prêt à Voter (see Figure 5.1): a permuted list of candidates with a serial number. The ballot is assumed to be physically unforgeable and is marked by the voter and split along the dotted line. The candidate ordering is shredded, while the mark position and serial number is optically scanned and then kept by the voter as a privacy-preserving receipt. In Figure 5.1, we also show a tabular form of the ballot that is exactly equivalent. This form of the ballot could be printed out and given to voters, however it would be a poor design relative to the ballot form on the lefthand side of the figure.

The tabular form of the ballot consists of 3 columns and $\mathcal{C}$ rows, where $\mathcal{C}$ is the number of candidates in the election. The first column, which we denote by **U**, are Unique IDs which contains a unique ballot identifier and a choice identifier. In the example ballot of figure 5.1, the ballot number is 1234 and the suffixes identify each of the $\mathcal{C}$ markable positions on ballot 1234. So in this case, markable position 1234.01 would count for Bob. On a different ballot, say 1235, position 1235.01 might correspond to a different candidate.

The second column is the *Marks List* column, which we denote by **M**. In this column, the voter places a checkmark at exactly one spot, indicating the row corresponding to the candidate the voter wishes to vote for. The last column is the *Candidate Selection* column, which we will denote by **S**. This is a list of the candidates in a randomly permuted (per-ballot) order.

Eperio Tables.

An Eperio table is a data structure that encodes the ballot information. If you were to take every ballot in tabular form, concatenate them end-to-end, you would end up with the 'canonical' Eperio



U	M	S
1234.01	☐	Bob
1234.02	☐	Alice
1234.03	☐	Charlie

Figure 5.1: A Prêt à Voter ballot with 3 candidates. Each ballot has a randomly shuffled order of candidates. Left side: the ballot as received by the voter. Right side: an equivalent formulation of the same ballot information in tabular form.

table. This canonical table is never used directly, but many (e.g., 20) instances of it are created which are row-wise shuffles the table. In the original Eperio protocol, the **U** and **S** columns are individually encrypted for each instance of an Eperio table prior to the election to be used in the post-election audit.

Eperio Protocol.

Prior to the election, a set of trustees use a blackbox device (trusted for ballot secrecy but not integrity) to generate a canonical Eperio table for an election with $\mathcal{C}$ candidates and $\mathcal{V}$ voters. All randomness used by the blackbox is deterministically derived from seeds provided by the trustees. The canonical table will be $3 \times \mathcal{CV}$. The canonical table is provided to the printers for printing the ballots. As in almost all paper-based E2E voting systems, printing is assumed to be a trustworthy process (at least with respect to ballot secrecy — a print audit will establish the correctness of the printed ballots but cannot distinguish between a malicious printer or honest printers being provide the wrong information to print).

A set of ℓ Eperio tables are generated by applying a random permutation to the rows of the canonical table by the blackbox. ℓ is a security parameter where an attack that moves a vote from Alice to Bob will be detected (given adequate receipt checks and print audits) with probability $1 - 2^{-\ell}$. The **U** and **S** columns of each Eperio table is publicly committed prior to voting.

During voting, voters may request a ballot to be print audited (we defer to the paper the discussion of the print audit — we can handle more simply in our protocol). They then fill out their ballots for their selected candidates and have the mark position portion of their ballot recorded (they can keep

this lefthand side of the ballot as a receipt). After the election, the trustees input into the blackbox their random seeds and the scanned ballots (**U** and **M**). The blackbox reconstructs all the tables and asserts an **M** column for each Eperio table. These **M** columns and an assertion of the final tally is published.

After the results have been asserted, a random beacon is used to select an ℓ-bit string; one bit for each Eperio table. If the bit for a given table is 0, the blackbox (again reseeded by the trustees) reveals the **U** column and if it is 1, it reveals the **S** column (the **M** column for each is already public). For each **UM**-revealed table, voters can check their receipt and everyone can check for consistency across each table. For each **MS**-revealed table, anyone can check that it matches the asserted tally. The specific reasoning for each of the three possible audits can be found in Essex et al. (2010). For any particular committed Eperio table, if only one of these combinations is opened, privacy is preserved.

5.6 Our Protocol

Our observation is that the encryption in Eperio is used as a commitment scheme and can be changed to any type of commitments. The authors themselves make this observation suggesting that the perfectly-binding commitment scheme (based on encryption) could be replaced with Pedersen commitments for everlasting privacy. We observe here that the commitments could be replaced with a VSS-style commitment to provide unconditional integrity and everlasting privacy (but sacrificing universal verifiability). Our protocol is given in Figure 5.2.

Verification. In our protocol, voters may engage in three checks. The first is a receipt check, which applies to any tables opened **UM**. External auditors may also check with these tables that no ballot is over-voted. The second check is a print audit, which applies to all rows in each table corresponding to a print audited ballot opened **UMS**. The final check is the correctness of the tally, checked with **MS**. Note all **UM** tables are shuffled but otherwise identical versions of the same data, and likewise with all **MS** tables. The basic integrity attack a malicious blackbox can conduct is changing the tally, which constitutes moving marks in the **M**. However it must guess which tables

will be opened **UM** and leave these unmodified (or the moved marks will be detectable via a receipt check), and guess exactly which tables will be opened **MS** to move the marks (or the tally will be unmodified, or inconsistent across tables). The probability of guessing correctly is $2^{-\ell}$ where ℓ is the number of tables. For $\ell = 20$ (a parameter used in Scantegrity for effectively the same purposes), the probability of guessing correct is less than a thousandth of a percentage. Importantly, this probability is independent of the adversary's computational power.

Discussion: Minimizing blackbox usage. The shareholders in our scheme are involved in three phases of the protocol: (1) preelection to use the blackbox to instantiate the election data, (2) after the election to use the blackbox to assert the mark column for each table, and (3) after the challenge to open up the data. In original Eperio, the blackbox must be used in all three steps. In our protocol, (3) can be accomplished by the shareholders directly without requiring the blackbox. In a variation of our protocol, we could also eliminate the blackbox from step (2). In step 2, the blackbox is required to permute a list of marks. The shareholders could do this directly if in step (1), the blackbox gave them each (in a specified order) a permutation to apply such that the composition of all these permutations is the permutation that was used. The issue is that this requires n-out-of-n shareholders in step (2) instead of k (however only k are required in step 3).[2]

5.7 Proof of Security (Sketch)

In our sketch of the security proof, we will reduce a breaking of either privacy or integrity to the breaking of one or more properties of the VSS scheme. We assume that the blackbox's computations are unobservable, and that the broadcast and private channels between shareholders are secure. In practice, these channels need not introduce extra cryptographic (and hence computational) assumptions, since they can be implemented as physical channels such as trusted couriers. In short, breaking either privacy or integrity will imply that strictly more than half of shareholders are malicious.

The reader can find in the Eperio paper [Essex et al. (2010)] the full reduction of security properties.

[2] Future work might explore the possibility of giving each shareholder a matrix that interpolates to the correct permutation matrix under the sequential composition of any k-out-of-n interpolations.

Pre-Casting

(1) Voters register with a local election authority. Issues of voter registration fraud are handled by the EA and are beyond the scope of this work.

(2) The EA publishes the number of candidates $\mathcal{C}$ and number of ballots to print (e.g., $2 \cdot \mathcal{V}$ where $\mathcal{V}$ is the voting age population and the scalar 2 allows for, on expectation, one print audit per voter). The EA sets security parameter ℓ.

(3) The blackbox uses local randomness to create the canonical Eperio table (which is provided to the printers) and ℓ permutations of it. It then uses VSS to commit the permuted tables to the shareholders, cell by cell. Each table's format and index is published. Upon completion, the shareholders purge the memory of the blackbox.

Vote Casting and Tallying

(1) Voters show up and register at the designated voting locations. For each voter, the EA will give the voter a paper ballot, such as the one in Figure 5.1, assuming they have not voted already.

(2) The voter may optionally choose to print audit the ballot. The scanner notes the serial number and its status as audited. The ballot is voided for voting purposes, and the voter is given the next ballot with the same option to audit or vote.

(3) Once the voter decides to vote, she marks her ballot and destroys the portion of the ballot containing the candidate ordering. The other portion, containing the serial number and marked position, is copied by the scanner and the original is kept by the voter as a privacy-preserving receipt.

(4) After the election, the scanners publish what they received: the **M** column of the canonical table.

(5) A quorum of at least k honest shareholders submit their shares of all tables to the blackbox, which reconstructs the canonical table (by sorting each Eperio table and checking for consistency). It also takes as input the scanner data. It outputs an asserted **M** column for each of the ℓ tables and an asserted final tally. The shareholders publish the output and purge the blackbox's memory.

Audit

(1) An unpredictable ℓ-bit value is publicly generated by a beacon (e.g., using stock prices [Clark and Hengartner (2010)]).

(2) For bit i of the beacon value, a quorum of at least k honest shareholders publish their shares of each cell in the **U** column in the i-th Eperio table if the bit is 0, and each cell in the **S** column if the bit is 1. For print audited ballots (only), they publish both the **U** and **S** cells.

(3) The shareholders securely delete all unused shares.

Figure 5.2: Our variant of Eperio using VSS.

5.7.1 Privacy

It was shown in Eperio that violating privacy reduces to a number of assumptions including breaking the *hiding* property of the commitment. Since we effectively only change the commitment scheme, we can ask ourselves: "If a cabal of malicious shareholders, auditors and voters collude, can they break the hiding property of the VSS-commitment?" Assuming, as always, that the number of malicious shareholders is a strict minority, the answer to the above question is no.

We do not pursue a full simulation-based proof but we comment that VSS-commitments have an additional property that should streamline such a proof, relative to the computational commitments used in Eperio. As a cut-and-choose protocol, Eperio faces a standard problem of simulateability: as the challenge space grows, the ability for the simulator to anticipate the correct challenge decreases exponentially (if it rewinds the verifier, it must do it an exponentially-increasing number of times which is not permissible). This can be side-stepped by, say, letting the simulator program the beacon value (by running it through a random oracle) or by repeating the protocol with one-bit challenges. In our case, a VSS-commitment is effectively a trapdoor commitment scheme for any majority of the shareholders. During the audit phase, the simulator can open a commitment in such a way that is perfectly consistent with any tally constraints imposed onto it.

Finally, we must also take care that each random choice (permutation in the tables) is truly random and not the result of a deterministic random generator (as in the original Eperio) or else the the permutations will not have a perfectly uniform distributed (which could be distinguished by an unbounded adversary). We modify Eperio along these lines — the shareholders do not contribute randomness, rather they remember shares of the randomness used (in the form of shuffled tables which can be resorted to recover the permutation).

5.7.2 Integrity

As in Eperio, the integrity of the election is reduced to a number of assumptions including the *binding* property of the commitment. We have replaced the commitment used by VSS, and in section 5.5.1 we have argued that VSS has properties which corresponds to the binding property of

a commitment scheme.

The auditing process remains the same. For each of the permuted Eperio tables, an auditor will ask the shareholders to open the commitments in such a way that corresponds to the three audits, as discussed in section 5.5.2. Assuming that the number of malicious shareholders are strictly less than half, the VSS binding property guarantees that they cannot change the commitment that has been successfully executed.

In fact, let us suppose that the malicious shareholders can arbitrarily control where the marks go in the permuted Eperio tables. However, since there is at least one honest shareholder, the malicious shareholders do not know how to consistently mark the votes. Therefore, with high probability increasing exponentially to one in the number of Eperio tables, either a voter will detect that his vote is inconsistent with his receipt when the **U** columns are opened during the auditing process, or an auditor will discover inconsistencies across different Eperio tables opened the same way. In either way, the malicious shareholders' cheating is detected.

5.8 Conclusion

We present a system, based on Eperio, that offers integrity and ballot secrecy against computationally unbounded adversaries, regardless of whether such an adversary is a voter, verifier, or election trustee. Further, our system enables voters to cast paper-based ballots, such as an optical scan ballot overlay as used in Scantegrity II or a permutation-style optical scan ballot as used in Prêt à Voter . Once the ballot is cast, the voter may leave and does not have to participate in tallying the election (in contrast to the other category of systems providing unconditional security: boardroom voting schemes).

To be even-handed, we point out that our system introduces several drawbacks. We rely on private and broadcast channels which, in practice, require computational cryptography, thereby negating information-theoretic security. We have argued that these channels may be implemented physically as untappable channels and in fact, for elections such as the Scantegrity II municipal election at Takoma Park, MD, election officials did meet in person in the same room to set-up the election and

to compute the final tally. Like other paper ballot systems, the physical ballots are assumed to be unforgeable (therefore malicious voters cannot repudiate a correct audit) and we trust the EAs to not peek at the printed physical ballots before issuing them to voters (which would break privacy). Both of these issues could be mitigated to a large extent by using Scantegrity II ballots, however in Scantegrity II the scanner learns how the vote was cast (as it is a cryptographic overlay and not a replacement system).

Most importantly in terms of drawbacks, our system removes the ability for voters to independently verify the election results. They must trust that a majority of shareholders are honest. While we have no data on how many voters do a full cryptographic check of the election results in a typical E2E-verifiable election, we expect that many will already defer to someone else to check (whether by running their software without validating it or simply believing their assertions). That said, universal verification provides the agility to decide who you trust after the election and even do it yourself if you do not adequately trust anyone else who can perform the check. We are not advocating that unconditional security trumps universal verification, but we believe it is important to provide viable solutions for both sides of this trade-off. This way, readers can decide which is most appropriate for their election requirements.

Chapter 6

Conclusions

In this work, we defined contamination – a problem at the foundation of the theory of multi-prover interactive proofs, and proposed a solution – locality-explicit multi-prover interactive proofs.

We reconciled the theoretical foundations of multi-prover protocols and contamination, and found a new property of zero-knowledge simulators directly related to non-locality. We applied the techniques of this locality-explicit framework to construct practical protocols which uses physical distancing to enforce no-communication. We discovered a link between trust and contamination, and exploited the homomorphic properties of the multi-prover commitment scheme to build an information-theoretically secure voting protocol.

Defending against adversaries that are augmented by non-local resources remains a theoretic problem. However, it is of note that, at the time of this writing, multiple organizations and nation-states have either deployed or are in the process of deploying satellites for quantum cryptography, which would allow entanglement distribution to be realized. Thus, at least one obstacle barring the instantiation of (quantum) non-local attacks is being solved. If non-locality ever becomes a practical resource, then contamination might become a practical problem.

References

Adida, B. (2008). Helios: web-based open-audit voting. In *Usenix security symposium.*

Adlam, E., & Kent, A. (2015). Deterministic relativistic quantum bit commitment. *CoRR*, *abs/1504.00943*. Retrieved from http://arxiv.org/abs/1504.00943

Babai, L. (1985, May). Trading group theory for randomness. In *Proceedings of the seventeenth annual acm symposium on theory of computing* (pp. 421–429).

Babai, L., Fortnow, L., & Lund, C. (1992, December). Non-deterministic exponential time has two-prover interactive protocols. *Comput. Complex.*, *2*(4), 374–374. Retrieved from http://dx.doi.org/10.1007/BF01200430 doi: 10.1007/BF01200430

Barrett, J., Linden, N., Massar, S., Pironio, S., Popescu, S., & Roberts, D. (2005, Feb). Nonlocal correlations as an information-theoretic resource. *Phys. Rev. A*, *71*, 022101. Retrieved from https://link.aps.org/doi/10.1103/PhysRevA.71.022101 doi: 10.1103/PhysRevA.71.022101

Bell, J. S. (1964). On the Einstein-Podolsky-Rosen paradox. *Physics*, *1*, 195–200.

Bell, S., Benaloh, J., Byrne, M. D., Debeauvoir, D., Eakin, B., Kortum, P., ... Winn, M. (2013). Star-vote: A secure, transparent, auditable, and reliable voting system. *JETS*.

Bellare, M., Feige, U., & Kilian, J. (1995). On the role of shared randomness in two prover proof systems. In *Third israel symposium on theory of computing and systems, ISTCS 1995, tel aviv, israel, january 4-6, 1995, proceedings* (pp. 199–208). IEEE Computer Society. Retrieved from https://doi.org/10.1109/ISTCS.1995.377031 doi: 10.1109/ISTCS.1995.377031

Benaloh, J. (2006). Simple verifiable elections. In *Evt.*

Benaloh (*né* Cohen), J. D., & Fisher, M. J. (1985). A robust and verifiable cryptographically secure election scheme. In *Sfcs*.

Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., & Rogaway, P. (1990). Everything provable is provable in zero-knowledge. In *Proceedings of the 8th annual international cryptology conference on advances in cryptology* (pp. 37–56). London, UK, UK: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=646753.704888

Ben-Or, M., Goldwasser, S., Kilian, J., & Wigderson, A. (1988). Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the twentieth annual acm symposium on theory of computing* (pp. 113–131). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/62212.62223 doi: 10.1145/62212.62223

Brassard, G., Broadbent, A., & Tapp, A. (2003). Multi-party pseudo-telepathy. In F. Dehne, J.-R. Sack, & M. Smid (Eds.), *Algorithms and data structures* (pp. 1–11). Berlin, Heidelberg: Springer Berlin Heidelberg.

Brassard, G., & Crépeau, C. (1986). Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for SAT and beyond. In 27^{th} *symp. of found. of computer sci.* (pp. 188–195). IEEE.

Brassard, G., & Crépeau, C. (1987). Zero-knowledge simulation of boolean circuits (extended abstract). In A. M. Odlyzko (Ed.), *Advances in cryptology: Proceedings of crypto '86* (Vol. 263, pp. 223–233). Springer-Verlag.

Brassard, G., Crépeau, C., Mayers, D., & Salvail, L. (1998, June). *Defeating classical bit commitments with a quantum computer.* arXiv:quant-ph/9806031.

Broadbent, A., & Tapp, A. (2008). Information-theoretically secure voting without an honest majority. In *Wote.*

Burton, C., Culnane, C., & Schneider, S. (2016). Verifiable electronic voting in practice: the use of vvote in the victorian state election. *IEEE Security and Privacy*.

Carback, R. T., Chaum, D., Clark, J., Conway, J., Essex, A., Hernson, P. S., ... Vora, P. L. (2010). Scantegrity II election at Takoma Park. In *Usenix security symposium.*

Carter, J., & Wegman, M. N. (1979). Universal classes of hash functions. *Journal of Computer*

and System Sciences, *18*(2), 143 - 154. Retrieved from http://www.sciencedirect.com/science/article/pii/0022000079900448 doi: https://doi.org/10.1016/0022-0000(79)90044-8

Chailloux, A., & Leverrier, A. (2017). Relativistic (or 2-prover 1-round) zero-knowledge protocol for np secure against quantum adversaries. In J.-S. Coron & J. B. Nielsen (Eds.), *Advances in cryptology – eurocrypt 2017: 36th annual international conference on the theory and applications of cryptographic techniques, paris, france, april 30 – may 4, 2017, proceedings, part iii* (pp. 369–396). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-56617-7_13 doi: 10.1007/978-3-319-56617-7_13

Chaum, D. (1988). Elections with unconditionally-secure ballots and disruption equivalent to breaking rsa. In *Eurocrypt*.

Chaum, D. (2004). Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, *2*(1), 38–47.

Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L., ... Sherman, A. T. (2008). Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *Evt*.

Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A. T., & Vora, P. (2008, May/June). Scantegrity: End-to-end voter verifiable optical-scan voting. *IEEE Security and Privacy*, *6*(3), 40–46.

Chaum, D., Fiat, A., & Naor, M. (1990). Untraceable electronic cash. In *Proceedings on advances in cryptology* (pp. 319–327). Berlin, Heidelberg: Springer-Verlag.

Chaum, D., Ryan, P. Y. A., & Schneider, S. (2005). A practical voter-verifiable election scheme. In *Esorics*.

Chiesa, A., Forbes, M. A., Gur, T., & Spooner, N. (2018). Spatial isolation implies zero knowledge even in a quantum world. *Electronic Colloquium on Computational Complexity (ECCC)*, *25*, 44.

Church, A. (1936). A note on the entscheidungsproblem. *Journal of Symbolic Logic*, *1*(1), 40–41. doi: 10.2307/2269326

Clark, J., & Hengartner, U. (2010). On the use of financial data as a random beacon. In *Evt/wote*.

Clauser, J. F., Horne, M. A., Shimony, A., & Holt, R. A. (1969, Oct). Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, *23*, 880–884. doi: 10.1103/PhysRevLett.23.880

Cleve, R., Hoyer, P., Toner, B., & Watrous, J. (2004). Consequences and limits of nonlocal strategies. In *Proceedings of the 19th ieee annual conference on computational complexity* (pp. 236–249). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dx.doi.org/10.1109/CCC.2004.9 doi: 10.1109/CCC.2004.9

Cramer, R., Franklin, M., Schoenmakers, B., & Yung, M. (1996). Multi-authority secret-ballot elections with linear work. In *Eurocrypt.*

Cramer, R., Gennaro, R., & Schoenmakers, B. (1997). A secure and optimally efficient multi-authority election scheme. In *Eurocrypt.*

Crépeau, C., Salvail, L., Simard, J.-R., & Tapp, A. (2011). Two provers in isolation. In *Advances in cryptology – asiacrypt 2011: 17th international conference on the theory and application of cryptology and information security, seoul, south korea, december 4-8, 2011. proceedings* (pp. 407–430). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-25385-0_22

Crépeau, C., & Yang, N. (2016). Multi-prover interactive proofs: Unsound foundations. *Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology*, 485-493.

Crepeau, C., & Yang, N. (2017). Multi-prover interactive proofs: Unsound foundations. In R. C.-W. Phan & M. Yung (Eds.), *Paradigms in cryptology – mycrypt 2016. malicious and exploratory cryptology: Second international conference, mycrypt 2016, kuala lumpur, malaysia, december 1-2, 2016, revised selected papers* (pp. 485–493). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-61273-7_25 doi: 10.1007/978-3-319-61273-7_25

Demirel, D., van de Graaf, J., & dos Santos Araujo, R. S. (2012). Improving helios with everlasting privacy towards the public. In *Evt/wote.*

Dwork, C., Feige, U., Kilian, J., Naor, M., & Safra, S. (1992). Low communication 2-prover zero-knowledge proofs for NP. In *Advances in cryptology - CRYPTO '92, 12th annual international cryptology conference, santa barbara, california, usa, august 16-20, 1992, proceedings*

(pp. 215–227). Retrieved from https://doi.org/10.1007/3-540-48071-4_15 doi: 10.1007/3-540-48071-4_15

Essex, A., Clark, J., Hengartner, U., & Adams, C. (2010). Eperio: Mitigating technical complexity in cryptographic election verification. In *Evt/wote*.

Fehr, S., & Fillinger, M. (2015). Multi-prover commitments against non-signaling attacks. In *Advances in cryptology – crypto 2015: 35th annual cryptology conference, santa barbara, ca, usa, august 16-20, 2015, proceedings, part ii* (pp. 403–421). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-662-48000-7_20

Feige, U., & Kilian, J. (1994). Two prover protocols: low error at affordable rates. In F. T. Leighton & M. T. Goodrich (Eds.), *Proceedings of the twenty-sixth annual ACM symposium on theory of computing, 23-25 may 1994, montréal, québec, canada* (pp. 172–183). ACM. doi: 10 .1145/195058.195128

Feige, U., & Kilian, J. (2000). Two-prover protocols - low error at affordable rates. *SIAM J. Comput.*, *30*(1), 324–346. Retrieved from https://doi.org/10.1137/S0097539797325375 doi: 10.1137/S0097539797325375

Feige, U., & Lovász, L. (1992). Two-prover one-round proof systems: Their power and their problems (extended abstract). In *Proceedings of the twenty-fourth annual acm symposium on theory of computing* (pp. 733–744). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/129712.129783 doi: 10.1145/129712.129783

Fortnow, L., Rompel, J., & Sipser, M. (1994, November). On the power of multi-prover interactive protocols. *Theor. Comput. Sci.*, *134*(2), 545–557. Retrieved from http://dx.doi.org/10.1016/0304-3975(94)90251-8 doi: 10.1016/0304-3975(94)90251-8

Gallegos-Garcia, G., Iovino, V., Rial, A., Ronne, P. B., & Ryan, P. Y. A. (2016). *(universal) unconditional verifiability in e-voting without trusted parties* (Tech. Rep.). IACR Eprint Report 2016/975.

Garay, J., Givens, C., Ostrovsky, R., & Raykov, P. (2014). Broadcast (and round) efficient verifiable secret sharing. In *Icits*.

Goldreich, O. (2006). *Foundations of cryptography: Volume 1*. New York, NY, USA: Cambridge University Press.

Goldreich, O., Micali, S., & Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, *38*(3).

Goldwasser, S., Micali, S., & Rackoff, C. (1989, February). The knowledge complexity of interactive proof-systems. *SIAM. J. Computing*, *18*(1), 186–208.

Grilo, A. B., Slofstra, W., & Yuen, H. (2019). Perfect zero knowledge for quantum multiprover interactive proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, *26*, 86.

Hao, F., & Zieliński, P. (2009). A 2-round anonymous veto protocol. In *Security protocols.*

Henry, K., Stinson, D. R., & Sui, J. (2009). The effectiveness of receipt-based attacks on threeballot. *IEEE TIFS*, *4*(4), 699–707.

Impagliazzo, R., & Yung, M. (1988). Direct minimum-knowledge computations. In C. Pomerance (Ed.), *Advances in cryptology: Proceedings of crypto '87* (Vol. 293, pp. 40–51). Springer-Verlag.

Ito, T., & Vidick, T. (2012). A multi-prover interactive proof for nexp sound against entangled provers. In *Proceedings of the 2012 ieee 53rd annual symposium on foundations of computer science* (pp. 243–252). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dx.doi.org/10.1109/FOCS.2012.11 doi: 10.1109/FOCS.2012.11

Kalai, Y. T., Raz, R., & Rothblum, R. D. (2014). How to delegate computations: The power of no-signaling proofs. In *Proceedings of the forty-sixth annual acm symposium on theory of computing* (pp. 485–494). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2591796.2591809 doi: 10.1145/2591796.2591809

Kempe, J., Kobayashi, H., Matsumoto, K., Toner, B., & Vidick, T. (2011). Entangled games are hard to approximate. *SIAM Journal on Computing*, *40*(3), 848-877. doi: 10.1137/090751293

Kempe, J., & Vidick, T. (2011). Parallel repetition of entangled games. In *Proceedings of 43rd acm symposium on theory of computing (stoc)* (p. 353-362).

Kent, A. (1999, Aug). Unconditionally secure bit commitment. *Phys. Rev. Lett.*, *83*, 1447–1450. doi: 10.1103/PhysRevLett.83.1447

Kiayias, A., & Yung, M. (2002). Self-tallying elections and perfect ballot secrecy. In *Pkc.*

Kiayias, A., Zacharias, T., & Zhang, B. (2015). *End-to-end verifiable elections in the standard model* (Tech. Rep. No. 2015/346). IACR Eprint Report.

Kilian, J. (1990a). Strong separation models of multi prover interactive proofs. In *Dimacs workshop on cryptography.*

Kilian, J. (1990b). *Uses of randomness in algorithms and protocols*. MIT Press.

Kilian, J. (2018, July). *Personal e-mail communication.*

Lapidot, D., & Shamir, A. (1991). Fully parallelized multi prover protocols for nexp-time (extended abstract). In *32nd annual symposium on foundations of computer science, san juan, puerto rico, 1-4 october 1991* (pp. 13–18). Retrieved from https://doi.org/10.1109/SFCS.1991.185342 doi: 10.1109/SFCS.1991.185342

Lapidot, D., & Shamir, A. (1995). A one-round, two-prover, zero-knowledge protocol for NP. *Combinatorica*, *15*(2), 204–214. doi: 10.1007/BF01200756

Lapidot, D., & Shamir, A. (1997). Fully parallelized multi-prover protocols for nexp-time. *J. Comput. Syst. Sci.*, *54*(2), 215–220. Retrieved from https://doi.org/10.1006/jcss.1997.1238 doi: 10.1006/jcss.1997.1238

Lunghi, T., Kaniewski, J., Bussières, F., Houlmann, R., Tomamichel, M., Wehner, S., & Zbinden, H. (2015, Jul). Practical relativistic bit commitment. *Phys. Rev. Lett.*, *115*, 030502. Retrieved from https://link.aps.org/doi/10.1103/PhysRevLett.115.030502 doi: 10.1103/PhysRevLett.115.030502

Malkhi, D., Margo, O., & Pavlov, E. (2002). E-voting without 'cryptography'. In *Financial cryptography.*

Mannan, M., Kim, B. H., Ganjali, A., & Lie, D. (2011). Unicorn: two-factor attestation for data security. In *Ccs.*

Moran, T., & Naor, M. (2006). Receipt-free universally-verifiable voting with everlasting privacy. In *Crypto.*

Moran, T., & Naor, M. (2007). Split-ballot voting: Everlasting privacy with distributed trust. In *Ccs.*

Neff, C. A. (2001). A verifiable secret shuffle and its application to e-voting. In *Ccs.*

Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information: 10th anniversary edition*. Cambridge University Press. doi: 10.1017/CBO9780511976667

Pedersen, T. P. (1992). Non-interactive and information-theoretic secure verifiable secret sharing. In

J. Feigenbaum (Ed.), *Advances in cryptology — crypto '91* (pp. 129–140). Berlin, Heidelberg: Springer Berlin Heidelberg.

Popescu, S., & Rohrlich, D. (1994). Quantum nonlocality as an axiom. *Foundations of Physics*, *24*(3), 379–385. Retrieved from http://dx.doi.org/10.1007/BF02058098 doi: 10.1007/BF02058098

Popescu, S., & Rohrlich, D. (1998). Causality and nonlocality as axioms for quantum mechanics. In G. Hunter, S. Jeffers, & J.-P. Vigier (Eds.), *Causality and locality in modern physics* (pp. 383–389). Dordrecht: Springer Netherlands.

Popoveniuc, S., & Hosp, B. (2006). An introduction to punchscan. In *Wote.*

Rabin, T., & Ben-Or, M. (1989). Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual acm symposium on theory of computing* (pp. 73–85). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/73007.73014 doi: 10.1145/73007.73014

Raz, R. (1998). A parallel repetition theorem. *SIAM Journal on Computing*, *27*(3), 763-803. doi: 10.1137/S0097539795280895

Riva, B., & Ta-Shma, A. (2007). Bare-handed electronic voting with pre-processing. In *Proceedings of the usenix workshop on accurate electronic voting technology* (pp. 15–15). Berkeley, CA, USA: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm?id=1323111.1323126

Rivest, R. L., & Smith, W. D. (2007). Three voting protocols: threeballot, VAV, and twin. In *Evt.*

Schoenmakers, B. (1999). A simple publicly verifiable secret sharing scheme and its applications to electronic voting. In *Crypto.*

Schoenmakers, B. (2000, July). Fully auditable electronic secret-ballot elections. *Xootic Magazine*.

Shamir, A. (1979). How to share a secret. *Commun. ACM*, *22*(11), 612–613. doi: 10.1145/359168.359176

Shamir, A. (1992, October). IP = PSPACE. *J. ACM*, *39*(4), 869–877. Retrieved from http://doi.acm.org/10.1145/146585.146609 doi: 10.1145/146585.146609

Stadler, M. (1996). Publicly verifiable secret sharing. In *Eurocrypt.*

Turing, A. M. (1937). Computability and lambda-definability. *Journal of Symbolic Logic*, *2*(4),

153–163. doi: 10.2307/2268280

Verbanis, E., Martin, A., Houlmann, R., Boso, G., Bussières, F., & Zbinden, H. (2016, Sep). 24-hour relativistic bit commitment. *Phys. Rev. Lett.*, *117*, 140506. doi: 10.1103/PhysRevLett .117.140506

Wilde, M. M. (2013). *Quantum information theory*. Cambridge University Press. doi: 10.1017/ CBO9781139525343

Yang, N., & Clark, J. (2017). Practical governmental voting with unconditional integrity and privacy. In M. Brenner et al. (Eds.), *Financial cryptography and data security* (pp. 434–449). Cham: Springer International Publishing.

Zagórski, F., Carback, R., Chaum, D., Clark, J., Essex, A., & Vora, P. L. (2013). Remotegrity: Design and use of an end-to-end verifiable remote voting system. In *Acns*.