Magic Functions

In Memoriam
Bernard M. Dwork
1923 – 1998

Cynthia Dwork*   Moni Naor†   Omer Reingold‡   Larry Stockmeyer§

January 9, 2004

A preliminary version of these results appeared in *Proceedings of 40th IEEE Symposium on Foundations of Computer Science, 1999.*

*Microsoft Research - Silicon Valley Campus, 1065 L'Avenida, Mountain View, CA 94043. Email: dwork@microsoft.com. Most of this work was performed while this author was at the IBM Almaden Research Center. Research supported by BSF Grant 32-00032-1.

†Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: naor@wisdom.weizmann.ac.il. Research supported by a grant from the Israel Science Foundation administered by the Israeli Academy of Sciences.

‡Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: reingold@wisdom.weizmann.ac.il. Research supported by an Eshkol Fellowship of the Israeli Ministry of Science.

§Email: stock@acm.org. Work performed while this author was at the IBM Almaden Research Center.

Abstract

We prove that three apparently unrelated fundamental problems in distributed computing, cryptography, and complexity theory, are essentially the same problem. These three problems and brief descriptions of them follow. (1) *The selective decommitment problem.* An adversary is given commitments to a collection of messages, and the adversary can ask for some subset of the commitments to be opened. The question is whether seeing the decommitments to these open plaintexts allows the adversary to learn something unexpected about the plaintexts that are unopened. (2) *The power of 3-round weak zero-knowledge arguments.* The question is what can be proved in (a possibly weakened form of) zero-knowledge in a 3-round argument. In particular, is there a language outside of BPP that has a 3-round public-coin weak zero-knowledge argument? (3) *The Fiat-Shamir methodology.* This is a method for converting a 3-round public-coin argument (viewed as an identification scheme) to a 1-round signature scheme. The method requires what we call a "magic function" that the signer applies to the first-round message of the argument to obtain a second-round message (queries from the verifier). An open question here is whether every 3-round public-coin argument for a language outside of BPP has a magic function.

It follows easily from definitions that if a 3-round public-coin argument system is zero-knowledge in the standard (fairly strong) sense, then it has no magic function. We define a weakening of zero-knowledge such that zero-knowledge $\Rightarrow$ no-magic-function still holds. For this weakened form of zero-knowledge we give a partial converse: informally, if a 3-round public-coin argument system is not weakly zero-knowledge, then some form of magic is possible for this argument system. We obtain our definition of weak zero-knowledge by a sequence of weakenings of the standard definition, forming a hierarchy. Intermediate forms of zero-knowledge in this hierarchy are reasonable ones, and they may be useful in applications. Finally, we relate the selective decommitment problem to public-coin proof systems and arguments at an intermediate level of the hierarchy, and obtain several positive security results for selective decommitment.

1 Introduction

In this paper we show that three apparently unrelated problems are in fact very closely related. We sketch these problems at a high level.

Selective Decommitment The *selective decommitment* problem first arose in a slightly different form, *selective decryption*, in the context of Byzantine agreement, no later than 1985. In the distributed setting, processors use encryption to hide information from an adversary. If the adversary is dynamic, then the adversary can choose which processors to make faulty – and therefore which of a collection of ciphertexts to "open" – as a function of the ciphertexts that it sees. The adversary may therefore see the decryptions of selected plaintexts (messages), hence the term *selective decryption*. The problem is to prove security of the unopened ciphertexts. It is theoretically possible for the adversary to compute some unexpected function of the vector of the plaintexts. The crux of the problem is commitment of the encryptions, rather than decodability of the encryptions (cf. [8]). The *selective decommitment* problem is similar, except that instead of seeing encryptions of plaintexts the adversary is given *commitments* to the plaintexts. This problem is poorly understood even in *strong-receiver* commitments, which leak no information about the plaintext values *information-theoretically* (in such a case we rely on the computational boundedness of the committer to ensure that only one value can be revealed during opening of the commitment). Some positive security results for selective decommitment appear in Section 7.4.

3-Round Weak Zero-Knowledge Arguments The second problem lies in the intersection of zero-knowledge interactive proof systems and complexity theory: What can be proved in (a possibly weakened form of) zero-knowledge in a 3-round argument (interactive proof in which the prover is polynomial-time bounded)? It is known that no language outside of BPP has a 3-round *black box* zero-knowledge argument [21]. No analogous result is known for ordinary (non-black box) zero-knowledge.[1] We consider a hierarchy of successive natural weakenings of zero-knowledge, beginning with 3-round black box zero-knowledge, so the class of languages having 3-round public-coin arguments in the top level of the hierarchy is exactly BPP. Roughly speaking, in each successive weakening the simulator is given additional information so that its task is less broad (this is accomplished by a switch in the quantification order in the definition of zero-knowledge). Thus, an interactive argument that satisfies a given definition in the hierarchy also satisfies all weaker definitions in the hierarchy. Many of the levels of the hierarchy are "reasonable" definitions that may have cryptographic applications (we give one example later). Let GAP denote the statement: there is a gap in this hierarchy, or equivalently there exists a 3-round public-coin argument for some $L \notin$ BPP that is zero-knowledge according to the weakest definition in the hierarchy. The second problem is to prove or disprove the statement GAP. Since there are natural 3-round public-coin proofs and argument systems for any $L \in$ NP ([22]; see also Section 7.2.2 of this paper) whose zero-knowledge status is open, the answer to this question has immediate application.

The Fiat-Shamir Methodology This problem is cryptographic, and addresses a methodology suggested by Fiat and Shamir [18] to construct a (non-interactive) signature scheme from any 3-round (not necessarily zero-knowledge) public-coin identification scheme. The methodology requires

[1] However, Barak [3] has recently shown that every language in NP has a constant-round public-coin (non-black box) zero-knowledge argument.

what we will term magic functions. Roughly speaking, in an identification scheme the party to be identified, say Alice, is associated with a public identification key x_A, to which only she has the corresponding private key y_A. During the identification process, the verifier interacts with the party that is claiming to be Alice, to verify that this party knows the private key corresponding to the public key x_A. The identification process should not leak y_A: otherwise after Alice has identified herself some number of times she may be impersonated in future conversations. There are several natural 3-round identification schemes. Suppose that the three messages of the original interactive identification scheme are called α, β, and γ, respectively. Intuitively, in the Fiat-Shamir methodology the magic functions are used as follows to construct a (non-interactive) signature scheme. A certain magic function f is made public. The signer prepares the first message, α, of the 3-round identification scheme. The signer then applies the magic function f to (α, m), where m is the message to be signed, to obtain a "query" β (which is often a vector of individual queries). The signer then prepares γ, the reply to the query β when the first message is α. Finally the signer sends m and the entire conversation $\alpha\beta\gamma$ to the recipient, who uses the public magic function to check that $\beta = f(\alpha)$ and uses the test used by the verifier in the identification scheme to check the validity of the conversation $\alpha\beta\gamma$.[2] The third problem is to prove the existence or non-existence of magic functions. Informally, let MAGIC be the statement that every 3-round public-coin argument for a language not in BPP has a magic function.

We show that these three problems are deeply connected:

1. If there exists a commitment scheme resilient to selective decommitment, then every language in NP has a 3-round public-coin weak zero-knowledge argument; the exact type of security of the commitment scheme determines the level in our hierarchy of definitions of zero-knowledge at which we can place NP.

2. If any language $L \notin$ BPP has a 3-round public-coin argument system (P, V) satisfying even the weakest type of zero-knowledge in our hierarchy, then the Fiat-Shamir methodology fails completely on this argument system: GAP $\Rightarrow \neg$MAGIC. This result should be expected, given that the proof follows immediately from definitions.

3. If GAP is false, *i.e.*, if 3-round public-coin arguments satisfying even the lowest level of the hierarchy exist only for languages in BPP, then for every 3-round public-coin argument for a language not in BPP there is a magic function that at least weakly supports the Fiat-Shamir methodology. In spirit, this says $\neg$GAP $\Rightarrow$ MAGIC. This result is less expected. A major portion of the paper is devoted to proving this result.

Thus, in spirit, GAP $\Leftrightarrow \neg$MAGIC. There is an equivalent way to summarize these results intuitively in terms of the weakest form of zero-knowledge in the hierarchy, which we call "ultra-weak zero-knowledge." Let UWZK be the statement that there is a 3-round public-coin argument system for a language outside of BPP that is ultra-weak zero-knowledge. Obviously GAP $\Leftrightarrow$ UWZK, so UWZK $\Leftrightarrow \neg$MAGIC. Although both GAP $\Leftrightarrow \neg$MAGIC and UWZK $\Leftrightarrow \neg$MAGIC provide succinct intuitive descriptions of results 2 and 3 above, we actually prove that these equivalences hold for each individual 3-round public-coin argument system (P, V). We may write this UWZK$(P, V) \Leftrightarrow \neg$MAGIC$(P, V)$, where UWZK means that (P, V) is a 3-round public-coin argument system that enjoys the weakest form of zero-knowledge in the hierarchy, and MAGIC(P, V)

[2] Fiat and Shamir proposed this approach for removing interaction from a specific identification scheme of their invention [18].

means that (P, V) has a magic function. Given Barak's recent result that every language in NP has a constant-round public-coin zero-knowledge argument [3], it is plausible that GAP and UWZK are true; this would hold if "constant-round" can be replaced by "3-round" in his result. Assuming that BPP $\neq$ NP, this would settle the GAP, UWZK, and MAGIC questions, namely, GAP and UWZK are true and MAGIC is false. However, UWZK$(P, V) \Leftrightarrow \neg$MAGIC$(P, V)$ would still be far from trivial.

In the statements of GAP, UWZK, and MAGIC, we have restricted the 3-round arguments to ones where the verifier's coins are public because the Fiat-Shamir methodology is usually considered only for this case. We note that we actually consider a generalization of the Fiat-Shamir methodology, in which we do not require that in the modified (1-round) signature scheme the verification test is the same as the verification in the original (3-round) identification scheme. We show that GAP implies that even this generalization of the Fiat-Shamir methodology fails, and conversely. The strength of the generalized methodology makes the first direction quite strong; however, it makes the converse relatively weaker. As discussed in more detail below, there are weaker versions of the statements GAP, UWZK, and MAGIC where tests are restricted to those used in the original Fiat-Shamir methodology. For these weaker versions we again have GAP $\Leftrightarrow$ UWZK $\Leftrightarrow$ $\neg$MAGIC. We prefer to state our results for the case of more general tests.

Figure 1 summarizes our main results (bold arrows and check mark) and how they connect the three areas of 3-round weak zero-knowledge, selective decommitment, and magic functions. The bold double arrow indicates our result that a 3-round public-coin argument belongs to the lowest level of our hierarchy iff this argument does not have a magic function. The bold single arrow indicates our result that the security of selective decommitment implies that for any language in NP there is a 3-round public-coin interactive proof that satisfies a form of zero knowledge that is intermediate in our hierarchy. The check mark indicates that for various weakenings of the selective decommitment problem we have given a positive answer to the question of security of selective decommitment. These weakenings of selective decommitment indicated in the figure are of a different flavor than the weakenings of zero-knowledge in defining the hierarchy, and hence do not constitute a parallel hierarchy.

The rest of the paper is organized as follows. Related work is discussed in Section 2. In Section 3 we state several definitions and conventions that are used in the paper, and we also review some standard definitions of zero-knowledge interactive argument systems. In Section 4 we present our hierarchy of definitions of zero-knowledge. This is done first informally, and then formally. Between the two we give some intuition for the equivalence UWZK $\Leftrightarrow$ $\neg$MAGIC. Using Section 4 as a common base, the paper can then be divided into three parts: Sections 5 and 6 contain the results that are the formal statements of UWZK$(P, V) \Rightarrow \neg$MAGIC(P, V) and $\neg$UWZK$(P, V) \Rightarrow$ MAGIC(P, V), respectively; Section 7 contains our results on selective decommitment, including its connection with a form of zero-knowledge in the hierarchy; and Section 8 contains results on the relation between two versions of uniform zero-knowledge at various levels of our hierarchy. These three parts are mutually independent. For example, a reader who is interested primarily in the results on selective decommitment can skip directly from Section 4 to Section 7. In Section 9 we point out several open questions and directions for future research.

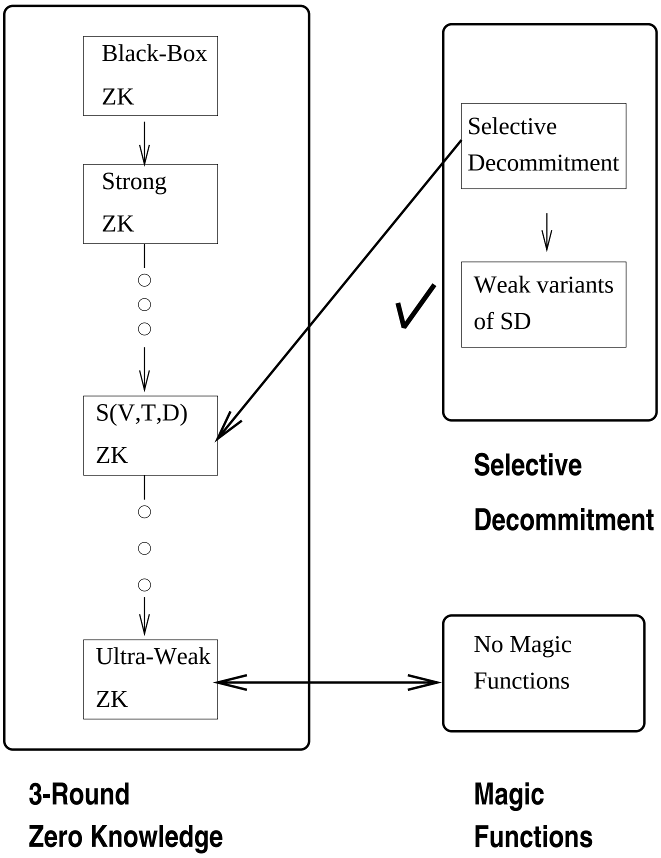

Figure 1: Summary of Results

2 Related Work

At least two other groups of researchers have independently observed variants of the statement "If there exist 3-round zero-knowledge proofs (under some relatively strong definition of zero-knowledge), then there are no magic functions" [32, 28]. Our results on this question are somewhat stronger, since they apply to a much weaker version of zero-knowledge, but the intuition is identical: since the magic function replaces the role of the verifier by generating the query message β, simulated conversations between the prover/signer and the magic function-verifier are indistinguishable from real conversations and hence are successful forgeries.

Chaum and Impagliazzo studied *correlation-intractable* functions and showed that if $\{f_k\}$ is correlation-intractable for all (arbitrary) relations R, in the sense that given k it is infeasible to find an x such that $R(x, f_k(x))$ is satisfied, then this collection of functions can be used to implement the Fiat-Shamir methodology [12]. Canetti, Goldreich, and Halevi proved a related statement [10]. They considered relations $R(x_1, \ldots, x_\ell, f_k(x_1), \ldots, f_k(x_\ell))$ where each of $x_1, \ldots, x_\ell$ is an n-bit string. Here, the goal is to find ℓ n-bit inputs x_i for which the above relation holds. They showed that if f_k takes inputs of length n then f_k is not ℓ-input ptime correlation intractable for any $\ell \geq \frac{|k|}{n}$. Intuitively, the relations that cause difficulty are those that depend on the choice of k. Thus, if there are enough input bits to name the key k then arranging correlations is easy. Bellare, Goldreich, and Impagliazzo, returning to single inputs, showed that if f_k is chosen randomly, and $|k|$ is sufficiently larger than n (specifically, $|k| \geq 4n$), then $\{f_k\}$ is 1-correlation intractable with respect to relations $R \in \mathrm{P}^{f_k}$ [5].

The concept of weakening the definition of zero-knowledge to achieve some degree of parallelizability and/or concurrency has appeared elsewhere, to wit, in Feige and Shamir's work on *witness indistinguishability* [17], and Dwork, Naor, and Sahai's work with *timing constraints* [15].

We note two interesting developments since our work first appeared.

1. Barak's constant-round public-coin zero-knowledge arguments for NP [3] have been mentioned above. The existence of a simulator for Barak's public-coin constant-round zero-knowledge arguments for all of NP implies, by the same reasoning as we employ to show that GAP $\Rightarrow \neg$MAGIC, that there is no interaction-removing function for these protocols.

2. In this paper, we have been presenting a tension between the possible existence of 3-round public-coin zero-knowledge proof/argument systems for languages outside of BPP and the question of the (non)existence of magic functions. In contrast, bypassing the resolution of this tension, Goldwasser and Tauman [27] exhibited a three-round public-coin identification scheme for which provably *no* magic function can exist (in the random oracle world these protocols *can* be converted into signature schemes [40, 1]).

3 Definitions and Conventions

In this section we introduce the types of computational objects that we will be using, and we review some standard definitions of zero-knowledge arguments. Additional discussion of zero-knowledge arguments is given in Section 4. All computational devices will be either polynomial-time (ptime) machines (*e.g.*, Turing machines) or polynomial-size (psize) families of circuits. Both types of devices can be either deterministic or probabilistic. Our default is that a device is a probabilistic ptime machine, unless otherwise noted. If a device is specified as a circuit family, the default is that it is probabilistic.

We let $\nu(n)$ denote a function that grows more slowly than the reciprocal of any polynomial, *i.e.*, for all $c > 0$ there is an n_0 such that $\nu(n) < 1/n^c$ for all $n \geq n_0$.

3.1 Interactive Argument Systems

A *interactive protocol* is a pair (P, V) of interactive probabilistic ptime machines, the *prover* P and the *verifier* V, where P has two inputs x and y, and V has one input x. Let $\mathrm{Accept}_{(P,V)}(x, y)$ be the event that V is in its accepting state at the end of the interactive computation. The protocol (P, V) is an *interactive argument for the language* L if the following two conditions hold (we let n denote the size of x):

1. *Completeness*: For all $x \in L$, there exists a y such that

$$\Pr[\mathrm{Accept}_{(P,V)}(x, y)] = 1 - \nu(n);$$

2. *Soundness*: For all probabilistic ptime machines P^*, for all $x \notin L$, for all y,

$$\Pr[\mathrm{Accept}_{(P^*,V)}(x,y)] < \nu(n).$$

We think of y as being a *witness* that proves that $x \in L$. For example, x might be a 3-colorable graph and y a 3-coloring of x. In general, we assume that there is a ptime-computable *witness relation* W, such that $x \in L$ iff there exists a y such that $W(x,y)$. The parameter n plays the role of a *security parameter*. In the case of the 3-coloring problem, for example, if n is the number of vertices, the assumption is that the time to find a 3-coloring in a 3-colorable graph (the time to find a y given x) grows faster than any polynomial in n.

3.2 Zero Knowledge

The intuition behind (P,V) being zero-knowledge is that a "cheating verifier" V^* interacting with P will not learn anything new about the witness y that V^* did not know before the interaction, provided that $W(x,y)$ (so $x \in L$). The definition is made more general by letting the verifier (both the good V and the cheating V^*) have some auxiliary information, denoted z, which could be, for example, a history of past interactions. (In this more general definition, "(x,y)" should be replaced by "(x,y,z)" in the completeness and soundness conditions above, and z should be universally quantified in both.) The sense in which V^* learns nothing new involves two more objects, the *simulator* S and the *test* T. Intuitively, V^* learns nothing new if the simulator, knowing x but not y, can produce a distribution of conversations that is indistinguishable, to every probabilistic ptime test T, from the distribution of conversations produced by V^* interacting with P. Let $(P,V^*)(x,y,z)$ denote the transcript of the conversation between P and V^* when P's input is (x,y) and V^*'s input is (x,z). (In this commonly used notation, we rely on the reader to remember that x and y are inputs to P, and x and z are inputs to V^*.) For the definition of zero knowledge in [25], the test T is allowed to be a psize family of deterministic circuits, and the definition takes the following form: For every (possibly cheating) verifier V^*, there exists a (probabilistic ptime machine) simulator S, such that for all deterministic psize circuit family tests T and all x,y,z such that $W(x,y)$,

$$|\Pr[T((P,V^*)(x,y,z))] - \Pr[T(S(x,z))]| < \nu(n) \qquad (1)$$

where the probability in the first (resp., second) term is over the random choices of P and V^* (resp., S). Here and subsequently, we write $\Pr[T(\dots)]$ as an abbreviation for $\Pr[T(\dots) = 1]$.

This is a strong definition, in that (1) must hold for all x,y,z (satisfying $W(x,y)$). Goldreich [19] has defined a *uniform* version of zero knowledge where it is assumed that x,y,z are produced by a polynomial-time samplable distribution D. The probabilistic ptime machine D gets an input of the form 1^n, and it outputs a triple (x,y,z). Thus, it defines, for each n, a distribution $\{X_nY_nZ_n\}$ on (x,y,z). In general D can be a joint distribution, *i.e.*, the distributions X_n, Y_n and Z_n are in general dependent. We are only interested in distributions D such that $W(x,y)$ for every (x,y,z) that can be produced, where W is a witness relation for the underlying language. The definition of uniform zero knowledge is: For all V^*, there exists an S, such that for all T and D,

$$|\Pr[T((P,V^*)(X_n,Y_n,Z_n))] - \Pr[T(S(X_n,Z_n))]| < \nu(n) \qquad (2)$$

where the probability in the first (resp., second) term is over the random choices of P, V^* and

D (resp., S and D).[3] In this paper we will work almost entirely with uniform definitions of zero knowledge.

We call this version of uniform zero-knowledge the *aggregate probabilities* (AP) version, because in each term we aggregate the probability, over the input distribution D, of producing a conversation on which T outputs 1. Goldreich [19] has noted that this definition is equivalent to the *almost all individuals* (AAI) one, in which the two probabilities should be negligibly close for an overwhelming fraction (colloquially, "almost all") of the (x, y, z)'s. In Section 8 we give a proof of this equivalence, and we study whether a similar equivalence holds at other levels of our hierarchy.

In proving $\mathrm{UWZK}(P, V) \Leftrightarrow \neg\mathrm{MAGIC}(P, V)$, the zero-knowledge property will be used heavily. However, we do not need the completeness and soundness properties.[4]

An important convention concerning expressions such as (2) should be explained. Whenever the same distribution name, say X_n, appears more than once in the same $\Pr[\ldots]$ term, all occurrences represent the same value of x. For example, in some definitions we will allow the test T access to the input x, so we will be considering terms such as $\Pr[T(X_n, S(X_n, Z_n))]$. This means that we first choose a (x, y, z) according to the distribution D, and then substitute x for X_n in both occurrences of X_n in this term, and substitute z for Z_n. However, occurrences of X_n, Y_n, Z_n in two different $\Pr[\ldots]$ terms represent (x, y, z)'s chosen independently for each term.

When given input 1^n, the distribution D produces strings over some finite alphabet Σ. It is convenient to assume that for each D there is a polynomial $Q(n) \geq n$ such that, if (x, y, z) is produced by D on input 1^n, then $|x| = |y| = |z| = Q(n)$ (shorter strings can be padded to length $Q(n)$). With D clear from context, we let Σ_n abbreviate $\Sigma^{Q(n)}$. We say that x has *size* n if $x \in \Sigma_n$, and similarly for y, z. Sometimes we attach to Σ_n the name of a distribution as an aid to reading. For example, $A \subseteq \Sigma_n^X \Sigma_n^Y$ means that A is a set of (x, y) pairs.

When we refer to "fractions" and "densities" of sets, these are with respect to the distribution D. If $A' \subseteq A \subseteq \Sigma_n^X$, then we say that A' *has density* δ *in* A, if $\Pr[\, x \in A' \,|\, x \in A\,] = \delta$, where the probability is with respect to X_n. Equivalently, we may say that "A' is a δ fraction of A." The *density of* A' is the density of A' in Σ_n^X. Similarly, if Σ_n^X is replaced by $\Sigma_n^X \Sigma_n^Y$, the probability is with respect to $X_n Y_n$.

3.3 Signature Schemes

We use a uniform version of signature schemes, where the inputs to the scheme are given by a probability distribution. A signature scheme consists of two probabilistic ptime devices: a *signer* G and a *checker* T. Given a message m, the signer sends m to the checker together with a signature of m, and the checker decides to either accept or reject the signature by outputting 1 or 0, respectively. There are two probability distributions. The first distribution $\{X_n Y_n\}$ produces pairs (x, y) where, intuitively, x is public information that is shared by the signer and the receiver, and y is private information that is known only to the signer. The second distribution $\{M_n\}$ produces messages m to be signed. For convenience, we combine the two distributions into one, $D = \{X_n Y_n M_n\}$, where M_n and $X_n Y_n$ are independent.

Then (G, T, D) is a *signature scheme* if, for infinitely many n,

[3] It is necessary for the non-triviality of (2) that S does not have access to D.

[4] However, the completeness and soundness conditions are relevant in Section 7.

1. *Completeness* (the signer can almost always sign succesfully):

$$\Pr[T(X_n, M_n, G(X_n, Y_n, M_n))] \geq 1 - \nu(n); \tag{3}$$

2. *Soundness* (every forger almost always fails to sign successfully): For all probabilistic circuit families S,

$$\Pr[T(X_n, M_n, S(X_n, M_n))] < \nu(n). \tag{4}$$

We require completeness and soundness for only infinitely many n, because the existence of a signature scheme (via a magic function) will be shown equivalent to the negation of ultra-weak zero-knowledge, and the definition of ultra-weak zero-knowledge holds for all sufficiently large n. This is an unimportant technical detail, because later we prove this equivalence for each n individually. At times we refer to G as the "good signer" and S as the "simulator/forger." We call S the simulator/forger because in making a connection between zero-knowledge and the existence of magic functions, *simulation* in a zero-knowledge argument corresponds to *forging* in a signature scheme, and vice versa.

The "gold standard" for security of a signature scheme is *existential unforgeability under chosen message attack* [26]. Versions of our results for signature schemes enjoying this strong form of security are presented in Sections 5.2 and 6.2.

4 A Hierarchy of Definitions of Zero Knowledge

4.1 Informal Description of the Hierarchy

Goldreich and Krawczyk have shown that there is no 3-round *black box* zero knowledge interactive proof or argument for any language outside of BPP [21]. Roughly speaking, a simulation is "black box" if the simulator does not know the internal structure of the verifier, *i.e.,* how the verifier chooses its queries. This is formulated by requiring a single simulator that generates conversations polynomial-time indistinguishable from real prover-verifier conversations; the simulator is an oracle machine S that interacts with the verifier V^* by submitting oracle queries to obtain the verifier's replies. The quantification order is:

$$\exists S \, \forall V^* \, \ldots$$

In "ordinary" zero-knowledge [25, 23], the order of quantification is reversed:

$$\forall V^* \, \exists S \, \ldots$$

The question of the existence of 3-round ordinary zero-knowledge proofs or arguments is unresolved: although it may be the case that the simulator can make use of the "internals" of V^*, it is not known how to exploit this information.[5]

We further weaken the definition of zero-knowledge in several ways: first, we move to *uniform* zero-knowledge [19] as defined in Section 3, in which one assumes a polynomial-time samplable probability distribution on instances of the argument; the requirement is that for every (possibly cheating) verifier there exists a simulator that with high probability over instances (x, y, z)

[5]As noted above, Barak [3] has found a way to make use of the internals of V^* to resolve this question where "3-round" is replaced by "constant round."

(according to the given distribution), produces conversations that are indistinguishable from real prover-verifier conversations by any polynomial-time machine.

Indistinguishability has its own quantifier: S must have the property that for all tests T and all distributions D, with high probability the conversations produced by S are indistinguishable by T from conversations produced by (P, V^*). Thus the order of quantifiers for uniform zero-knowledge is $\forall V^* \exists S \forall T \forall D \ldots$ and the full uniform zero-knowledge requirement is

$$\forall V^* \exists S \forall T \forall D \,|\Pr[T((P,V^*)(X_n,Y_n,Z_n))] - \Pr[T(S(X_n,Z_n))]\,| < \nu(n).$$

The probability space in the term $\Pr[T((P,V^*)(X_n,Y_n,Z_n))]$ is over choices made by P, V^*, T, and the choice of (x,y,z) made by (X_n,Y_n,Z_n). The probability space in the term $\Pr[T(S(X_n,Z_n))]$ is over choices made by S and T, and the choice made by (X_n,Y_n,Z_n) (however, the simulator does not get access to the private input y to the prover).

Our second weakening of zero-knowledge is to permit the simulator to know the *test* T on which its simulations will be judged:

$$\forall V^* \forall T \exists S \forall D \ldots$$

Thus, knowing the (possibly cheating) verifier V^* and the distinguisher T, the simulator must produce conversations that for all D are indistinguishable (in the aggregate) by T to real conversations between the prover and V^*.

We further weaken the definition of zero-knowledge by permitting the simulator to know the *distribution* D on inputs:

$$\forall V^* \forall T \forall D \exists S \ \ldots$$

We call this $S(V,T,D)$ *zero-knowledge* because S may depend on V^*, T, and D. Intuitively, this definition says the following: "You name a type of information that you think is leaked by the protocol; I will exhibit a simulator that proves that this particular information is not leaked." This definition would be useful, for example, if the distribution D is fixed and we would be happy with a protocol that does not leak a specific type of information as defined by a specific test T; in such a situation, it would be overkill to require the protocol to leak no information regardless of what the distribution and test are. $S(V,T,D)$ is the type of zero-knowledge that is needed when a zero-knowledge protocol is executed as a subroutine inside a protocol that itself might not be zero-knowledge (see Section 9 for a concrete example). In particular, this type of zero-knowledge suffices when the security requirements of the outer protocol are such that it is easy to detect at the end of the protocol whether cheating occurred (*e.g.*, forging of a signature). We can think of the adversary attacking the outer protocol as being both the verifier and the tester. The test is whether it has succeeded in cheating. The distribution D can also be known, since it is defined by the outer protocol. It is therefore an interesting question whether a language outside of BPP has a 3-round interactive argument that is zero-knowledge under this definition. In Section 7 we give a definition of security for selective decommitment and prove that if there exists a commitment scheme secure against selective decommitment, then there exists a 3-round public-coin weak zero-knowledge argument (under this definition of ZK) for every language in NP.

For brevity in this informal discussion, we combine the final weakenings to obtain an "ultra-weak" definition of zero knowledge:

(W1) we restrict the verifiers to be deterministic, (W2) we only require simulations to pass those tests T that output 1 on real conversations with overwhelming probability,

(W3) we restrict the tests to be deterministic,

(W4) we permit the simulator to be a probabilistic polynomial-size family of circuits, and

(W5) the distribution Z_n is independent of $X_n Y_n$.

The conditions (W1) on V^* and (W2) on T hold in our formulation of the Fiat-Shamir methodology (see Section 5); in particular, V^* is restricted to be deterministic because V^* will play the role of a magic *function*. We need (W5) because we will view z as the message, which is chosen independently of x and y (see Section 5 for discussion of adversarially chosen messages and histories). The restrictions (W3) and (W4) are not crucial to the result that $\text{UWZK}(P, V) \Rightarrow \neg\text{MAGIC}(P, V)$, that is, if (P, V) is ultra-weak zero-knowledge then the Fiat-Shamir methodology cannot be applied to this (P, V). In particular, the move to circuit simulators/forgers may be inconsistent with the common view of the Fiat-Shamir methodology; however, this result also holds if the simulators are machines, provided that we make this change in both the definition and the formulation.

Because signature schemes also involve a message to be signed, we give (at the beginning of Section 5) a definition of ultra-weak zero knowledge involving messages. This is done by viewing the auxiliary input z as the message. Because we are working in the uniform framework of zero-knowledge, we assume that messages are chosen from a given distribution M on messages that is independent from the distribution used to choose the (public key, private key) = (x, y). Thus, there are independent ptime samplable distributions $D' = \{X_n Y_n\}$ and $M = \{M_n\}$ such that $D = \{X_n Y_n M_n\}$. Because Z_n is independent of $X_n Y_n$ in the definition of ultra-weak zero-knowledge, we take our formal definition of ultra-weak uniform zero-knowledge with messages to be identical to the definition of ultra-weak zero-knowledge, changing only Z_n to M_n to indicate that we are viewing z's as messages to be signed.

It is for this type of ultra-weak zero-knowledge, formalized in Definition 5.1 in Section 5, that we prove our result UWZK $\Leftrightarrow \neg$MAGIC. First we prove a *negative* result, UWZK $\Rightarrow \neg$MAGIC (presented in Section 5):

> If for some $L \notin$ BPP there exists a 3-round public-coin argument that is zero-knowledge in this ultra-weak sense, then the Fiat-Shamir methodology for building noninteractive signature schemes from 3-round arguments completely fails for at least one 3-round public-coin argument for a language not in BPP: if the signature scheme is complete (can sign essentially any message; see (3)) then there is a forger that succeeds with overwhelming probability (see (4) with "$< \nu(n)$" replaced by "$> 1 - \nu(n)$").

For ultra-weak zero-knowledge we can also prove a *hard core* theorem (presented in Section 6), $\neg$UWZK $\Rightarrow$ MAGIC. Intuitively, the theorem states:

> If only languages in BPP have 3-round public-coin interactive arguments that satisfy this ultra-weak form of zero-knowledge, then the Fiat-Shamir methodology can be applied to every 3-round public-coin argument for a language not in BPP, provided that the public key x and the message m are restricted to "hard cores" of public keys and messages, respectively, where the hard core of messages may depend on the particular public key used. A hard core generally does not contain all public keys (resp., messages) but it

contains some $1/\mathrm{poly}(n)$ fraction of the public keys (resp., messages). Another part of our result is that there exist deterministic psize circuits that decide membership in the hard cores; it follows that the hard cores are samplable by probabilistic psize circuits.

As noted in the Introduction, even if UWZK holds, this direction of our result can still be applied to individual argument systems: $\neg\mathrm{UWZK}(P, V) \Rightarrow \mathrm{MAGIC}(P, V)$.

A slightly more detailed description of the hard core result ($\neg\mathrm{UWZK}(P, V) \Rightarrow \mathrm{MAGIC}(P, V)$) and its relation to the Fiat-Shamir methodology follows. We view each (P, V^*) as signer using the Fiat-Shamir methodology as explained in the Introduction. That is, given x, y, m, the signer (P, V^*) simulates $P(x, y, m)$ to obtain the first message α; it then computes $\beta = V^*(x, m, \alpha)$; and it again simulates P to obtain its response γ to β. The signature is $\alpha\beta\gamma$.

The hard core result states that if (P, V) does not satisfy the ultra-weak definition, and the verifier V^*, the test T, and the distribution D witness this fact, then V^* is a "magic function" and $((P, V^*), T, D)$ is a signature scheme on certain "hard cores" of x's (public keys) and messages. That is, for each polynomial upper bound n^d on the size of the simulator, there is a hard core $\mathcal{C}$ containing some $1/\mathrm{poly}(n)$ fraction of the x's, and for each $x \in \mathcal{C}$ there is a hard core $\mathcal{M}(x)$ containing some $1/\mathrm{poly}(n)$ fraction of the messages, such that: (i) the good signer (P, V^*) can produce, for all but a negligible number of (x, y, m), a signature that passes the test T with all but negligible probability; and (ii) for any simulator/forger S of size n^d, for infinitely many n, and all but a small $1/\mathrm{poly}(n)$ fraction of the public keys $x \in \mathcal{C}$ and messages $m \in \mathcal{M}(x)$ in the hard cores, the probability that S forges a signature on m when the public key is x is $1/\mathrm{poly}(n)$ small. In other words, V^* provides a magic function for (P, V) on inputs x and m in the hard cores that can be used to remove interaction from the 3-round interactive argument (P, V^*), turning it into a 1-round signature scheme, $((P, V^*), T, D)$. Completeness of the signature scheme (almost all messages can be successfully signed) follows from (i), and a weak form of soundness (every forger of size n^d has only a small $1/\mathrm{poly}(n)$ chance of forging a signature, for almost all x and m in the hard cores) follows from (ii).

Our hard core result is actually more general. It relates the density of the hard cores to the simulation error of the "best" (having the smallest error) circuit simulator, where the density decrease as the error decreases. Assuming only that (P, V) is not ultra-weak zero-knowledge, the smallest simulation error can be as small as $1/n^c$ for some constant c, and consequently the density of the hard cores is $1/\mathrm{poly}(n)$. But, for example, if the smallest simulation error is $1/2$, then the density of the hard cores is at least 0.29.

In the proof of the hard core result, we use the weakenings (W3) that T is deterministic and (W4) that simulators/forgers are circuit families. Indeed, we give an example showing that a hard core result of the strength we want is not true if T can be probabilistic.

As discussed in the Introduction, the spirit of these results is that UWZK $\Leftrightarrow$ $\neg$MAGIC, and more generally that $\mathrm{UWZK}(P, V) \Leftrightarrow \neg\mathrm{MAGIC}(P, V)$ for all 3-round public-coin argument systems (P, V). In addition, we prove a version of $\mathrm{UWZK}(P, V) \Leftrightarrow \neg\mathrm{MAGIC}(P, V)$ for a model where the adversary may use an adaptive chosen message attack to produce a particular message and an attempted signature on that message.

Before giving the formal definition of the hierarchy in Section 4.3, first we illustrate the connection between weak zero-knowledge and $\neg$MAGIC by giving a trivial proof of the equivalence for a further weakening of ultra-weak zero-knowledge.

4.2 An Easy Proof of Weak Zero-Knowledge iff ¬MAGIC.

We use a "badly-weak" definition of uniform zero-knowledge and give a trivial proof that badly-weak zero-knowledge is equivalent to no magic function. Badly-weak zero-knowledge is a further weakening of ultra-weak zero-knowledge. In ultra-weak zero-knowledge, the simulator can fail (to produce conversations that are indistinguishable by T from real (P, V^*) conversations) with only negligible ($\nu(n)$) probability. (In fact, this is one "strength" of zero-knowledge that survived all the weakening between ordinary zero-knowledge and ultra-weak zero-knowledge.) In the definition of badly-weak zero-knowledge, the simulator may fail with probability as large as $1 - n^{-c}$ for some constant c. This makes the definition unusable in practice and therefore uninteresting in theory, although it is useful for illustration.

It is now virtually trivial to show that (P, V) is not badly-weak zero-knowledge iff there exists a magic function V^*, a distribution D, and a test T such that $((P, V^*), T, D)$ is a signature scheme. The negation of the definition of badly-weak zero knowledge is: There exists a deterministic V^*, a deterministic ptime T (that outputs 1 on real conversations with overwhelming probability), and a distribution $\{X_nY_nZ_n\}$ (with Z_n independent of X_nY_n), such that for all probabilistic psize circuit families S, there are infinitely many n such that

$$|\Pr[T(X_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Z_n, (S(X_n, Z_n))] \,| \geq 1 - \nu(n).$$

Using that T has property (W2) and renaming Z_n as M_n, this is equivalent to

$$\Pr[T(X_n, M_n, S(X_n, M_n))] \,| < \nu(n),$$

and this is exactly the soundness condition for a signature scheme. The completeness condition for $((P, V^*), T, D)$ is exactly the property (W2) of T.

Note that if we try to apply this trivial argument with ultra-weak zero-knowledge in place of badly-weak zero-knowledge, the direction $\mathrm{UWZK}(P, V) \Rightarrow \neg\mathrm{MAGIC}(P, V)$ still holds (trivially), but the other direction, from $\neg\mathrm{UWZK}(P, V)$ to a signature scheme, gives a "badly-weak" signature scheme where the forger can succeed with probability as large as $1 - n^{-c}$ for some constant c.

4.3 Formal Description of the Hierarchy

In this section we give formal definitions for the various definitions of zero-knowledge just discussed. Recall our default that all computational devices P, V, D, T, S are probabilistic polynomial-time machines, and that circuit families (when used) are probabilistic and polynomial-size. In the first two definitions in the hierarchy (which are previously known definitions), we allow the test T to be a deterministic circuit family; this is indicated by the quantifier $\forall_{\mathrm{dcf}}$. We think of each of the definitions as stating a property of a protocol (P, V), even though, like previous definitions of zero-knowledge, the definitions do not involve the "good verifier" V; they are statements about P only. Although our interest in this paper is in 3-round protocols, the definitions apply to multi-round protocols as well.

In each definition, we are interested in the zero-knowledge property holding only for those (x, y) pairs that satisfy a given ptime-computable relation W (see Section 3). In particular, when we write $\exists D$ or $\forall D$, we are quantifying over those D such that $W(x, y)$ for every (x, y, z) that can be produced by D. Similarly, when we write $\forall x, y$, it means "$\forall x, y$ such that $W(x, y)$".

We begin the hierarchy using circuit-family tests, because this was the choice made in the original paper [25] on zero-knowledge (see also [23, 38]). Recall that S is an oracle machine in this definition.

Definition 4.1 (Black Box Zero Knowledge (over W))
$\exists S\, \forall V^*\, \forall x, y, z\, \forall_{\mathrm{dcf}}\, T$

$$|\Pr[T((P, V^*)(x, y, z))] - \Pr[T(S^{V^*}(x, z))]| < \nu(n)$$

where the probabilities are over the coin flips of P, V^* and S.

In the next definition the simulator can depend on V^*, so we call it $S(V)$ zero-knowledge. This is the definition of zero-knowledge by Goldwasser, Micali, and Rackoff [25], as modified by Goldreich and Oren [23] to include histories z.

Definition 4.2 ($S(V)$ Zero-Knowledge (over W))
$\forall V^*\, \exists S\, \forall x, y, z\, \forall_{\mathrm{dcf}}\, T$

$$|\Pr[T((P, V^*)(x, y, z))] - \Pr[T(S(x, z))]| < \nu(n)$$

where the probabilities are over the coin flips of P, V^* and S.

In the next definition we combine two changes to the test: first we augment the test T by giving it as input the values x, y, z; second (anticipating our move to the uniform model of zero-knowledge) we let T be a probabilistic ptime machine. The "augmented" definitions that follow are important for proving the connection between selective decommitment and 3-round zero-knowledge.

Definition 4.3 (Augmented $S(V)$ Zero-Knowledge (over W))
$\forall V^*\, \exists S\, \forall x, y, z\, \forall T$

$$|\Pr[T(x, y, z, (P, V^*)(x, y, z))] - \Pr[T(x, y, z, S(x, z))]| < \nu(n)$$

where the probabilities are over the coin flips of P, V^*, T and S.

Lemma 4.4 *If (P, V) is zero-knowledge under Definition 4.2 then (P, V) is zero-knowledge under Definition 4.3.*

Proof. The proof is done in two steps. Let Definition 4.3_{circ} be identical to Definition 4.3, except that tests are still psize circuit families.

1. We first show that Definition 4.2 is equivalent to Definition 4.3_{circ}. One direction is immediate. For the other direction, since the random variables $(P, V^*)(x, y, z)$ and $S(x, z)$ are indistinguishable to all deterministic polynomial-size circuit tests T, they are indistinguishable in particular to any T that has x, y, z hard-wired in.

2. The second step is to show that Definition 4.3_{circ} implies Definition 4.3. We show the contrapositive. Assume that (P, V) does not satisfy Definition 4.3. Then
$\exists V^*\, \forall S\, \exists T\, \exists c\, \exists$ infinitely many $n\, \exists x, y, z$

$$|\Pr[T(x, y, z, (P, V^*)(x, y, z))] - \Pr[T(x, y, z, S(x, z))]| \geq 1/n^c.$$

For each such n, x, y, z there must be a particular choice ρ for T's string of random coins, such that this inequality holds. For each such n, x, y, z, the probabilistic ptime machine T and the string ρ can be combined into a deterministic psize circuit, and the existence of these circuits shows that (P, V) does not satisfy Definition 4.3_{circ}. □

We now move to the uniform model of Goldreich [19], where individual x, y, z are replaced by distributions.

Definition 4.5 (Augmented Uniform $S(V)$ Zero-Knowledge (over W))
$\forall V^* \, \exists S \, \forall T \, \forall D = \{X_n Y_n Z_n\}$

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < \nu(n)$$

where the probability in the first term is over the choice made by (X_n, Y_n, Z_n) and the coin flips of P, V^* and T; and the probability in the second term is over the (independent) choice made by (X_n, Y_n, Z_n) and the coin flips of S and T. The presence of the input X_n in the second term ensures that the simulator is being "tested" on its ability to simulate on the "right" inputs (recall that in keeping with the convention established in Section 3, all occurrences of X_n in the second term represent the same value for x).

Lemma 4.6 *If (P, V) is zero-knowledge under Definition 4.3 then (P, V) is zero-knowledge under Definition 4.5.*

Proof. The simulator S whose existence is ensured by compliance with Definition 4.3 also satisfies the conditions of Definition 4.5. □

Definition 4.7 (Augmented Uniform $S(V, T)$ Zero-Knowledge (over W))
$\forall V^* \, \forall T \, \exists S \, \forall D = \{X_n Y_n Z_n\}$

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

The proof of the following lemma and the next one are identical to the proof of Lemma 4.6: it is obvious that the same simulator works. So these simple proofs are omitted.

Lemma 4.8 *If (P, V) is zero-knowledge under Definition 4.5, then (P, V) is zero-knowledge under Definition 4.7.*

Definition 4.9 (Augmented Uniform $S(V, T, D)$ Zero-Knowledge (over W))
$\forall V^* \, \forall T \, \forall D = \{X_n Y_n Z_n\} \, \exists S$

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

Lemma 4.10 *If (P, V) is zero-knowledge under Definition 4.7, then (P, V) is zero-knowledge under Definition 4.9.*

In Section 7 we will prove that if a commitment scheme satisfies a certain natural definition of security then all of NP has a 3-round public-coin augmented uniform $S(V, T, D)$ zero-knowledge argument.

For the results on the Fiat-Shamir methodology we no longer need that the witness (the prover's private input) y is an input to the test T. Thus, in the next definition we deny the test T access to the witness, and we drop the appellation "augmented".

Definition 4.11 (Uniform $S(V,T,D)$ Zero-Knowledge (over W))
$\forall V^* \, \forall T \, \forall D = \{X_n Y_n Z_n\} \, \exists S$

$$|\Pr[T(X_n, Z_n, (P,V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

Lemma 4.12 *If (P,V) is zero-knowledge under Definition 4.9, then (P,V) is zero-knowledge under Definition 4.11.*

Proof. The simulator S whose existence is ensured by compliance with Definition 4.9 also satisfies the conditions of Definition 4.11. This is because if a test T cannot distinguish real from simulated conversations when given a witness, it certainly cannot distinguish without this information. □

For brevity, we combine several weakenings to obtain the next definition: making the verifier and test deterministic, permitting the simulator to be a circuit family, and restricting the tests T to be those that, with overwhelming probability, output 1 on real conversations. If $S = \{S_n\}$ is a circuit family, we let $S(x,z)$ abbreviate $S_n(x,z)$ where n is the size of x and z.

Definition 4.13 (Weak Uniform Zero-Knowledge (over W))
For all deterministic ptime verifiers V^*, for all deterministic ptime tests T such that

$$\Pr[T(X_n, Z_n, (P,V^*)(X_n, Y_n, Z_n))] = 1 - \nu(n),$$

$\forall D = \{X_n Y_n Z_n\}$*, there exists a family S of psize probabilistic circuits, such that*

$$|\Pr[T(X_n, Z_n, (P,V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

For the final weakening, we restrict distributions $D = \{X_n Y_n Z_n\}$ to those where Z_n is independent of $X_n Y_n$ for all n. The reason for this, explained in more details very shortly, is that we will view z as the message in a signature scheme, where the message is chosen independently of x and y.

Definition 4.14 (Ultra-Weak Uniform Zero-Knowledge (over W))
For all deterministic ptime verifiers V^*, for all deterministic ptime tests T such that

$$\Pr[T(X_n, Z_n, (P,V^*)(X_n, Y_n, Z_n))] = 1 - \nu(n),$$

$\forall D = \{X_n Y_n Z_n\}$ *with Z_n independent of $X_n Y_n$, there exists a family S of psize probabilistic circuits, such that*

$$|\Pr[T(X_n, Z_n, (P,V)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

This completes the description of the zero-knowledge hierarchy. In the next section we connect 3-round public-coin arguments for languages outside of BPP satisfying Definition 4.14 with the Fiat-Shamir methodology.

5 Fiat-Shamir Methodology

Because the object of the Fiat-Shamir methodology is to construct schemes for signing messages, we need a definition of zero knowledge that has a message input. In both Section 5 (UWZK $\Rightarrow$ $\neg$MAGIC) and Section 6 ($\neg$UWZK $\Rightarrow$ MAGIC), we consider two ways of introducing a message input into the definition of zero knowledge. The first is to assume that messages are chosen at random from a distribution $M = \{M_n\}$ that is independent of the distribution used to choose the public key x and the signer's private key y. This model where messages are chosen randomly, and results for it, are given in Section 5.1. In the following Section 6, this is the principal model used to state and prove our results.

In Section 5.2 we define a second model where the adversary (the simulator/forger S) is allowed to choose the message on which it tries to forge a signature, after performing a probabilistic ptime chosen message attack using the signature algorithm as an oracle. In Sections 5.2 and 6.2, we show that the proof of UWZK $\Leftrightarrow$ $\neg$MAGIC using the first model (random message) can easily be modified to prove a similar result for the second model (adaptively chosen message), provided of course that the same model is used in the definition of both UWZK and MAGIC.

5.1 Randomly Chosen Messages

Typically, x and an associated witness y are thought of as inputs to an interactive argument. For signature and identification schemes, however, we think of x and y as being a public key and private key, respectively, that (we hope) can be used for many executions of the protocol with different messages. A pair (x, y) is generated by a distribution D'. A signature scheme has an additional message input m, which is given to the signer and to the test T that checks the validity of a supposed signature. The message m is also given to the forger/simulator S who tries, without knowing the private key y, to forge a signature on m that the test T will accept as valid. In the uniform model, we assume that there is a polynomial-time samplable distribution M for generating messages. We assume that M and D' are independent. In Definition 4.14, the verifier's auxiliary input z is given to V^*, T, and S, and z is chosen independently of x and y. Therefore, we can take z to be the (independently chosen) message. (Because the prover and verifier will work together to form the signer, giving m to the verifier also gives m to the signer.) The following definition is thus equivalent to Definition 4.14, renaming Z_n as M_n. We write M_n and m in place of Z_n and z, respectively, as a reminder that M_n is independent of X_n and Y_n.

Definition 5.1 (Ultra-Weak Uniform Zero-Knowledge with Messages (over W)) *For all deterministic ptime verifiers V^*, for all deterministic ptime tests T such that*

$$\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] = 1 - \nu(n),$$

$\forall D = \{X_n Y_n M_n\}$ with M_n independent of $X_n Y_n$, there exists a family S of psize probabilistic circuits, such that

$$|\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] - \Pr[T(X_n, M_n, S(X_n, M_n))]| < \nu(n).$$

We will discuss below how additional information, depending on x and y, can be included in z along with the message m.

Our interpretation of the Fiat-Shamir methodology is as follows: for all 3-round public-coin identification schemes (P, V) there is a magic function V^* that can be used to remove interaction. Think of a verifier as being composed of two parts: one (V^*) generating queries, and the other (T) checking the validity of the replies to the queries. Viewed this way, the Fiat-Shamir methodology, applied to a particular (P, V), says (cf. the definition of a signature scheme in Section 3.3):

Definition 5.2 (Fiat-Shamir Methodology Applied to (P, V))
$\exists$ deterministic V^*, $\exists$ deterministic T, $\exists D = \{X_n Y_n M_n\}$, $\forall$ probabilistic psize circuit families S,

1. *Completeness ((P, V^*) can almost always sign successfully)*

$$\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] \geq 1 - \nu(n);$$

2. *Soundness (every forger almost always fails to sign successfully)*

$$\Pr[T(X_n, M_n, S(X_n, M_n))] < \nu(n).$$

Note that, although V^* is deterministic, P is not in general, so the signer can use randomization.

As noted in the Introduction, this is a slight generalization of the approach in [18]; in the original approach the test T is explicitly required to be a check that the "right" queries were answered, together with a check that the acceptance criteria of V^* are satisfied.

Our version of the general methodology then says[6]:

Conjecture 5.3 (Fiat-Shamir Methodology) *For all 3-round public-coin identification schemes (P, V), the methodology of Definition 5.2 can be applied to (P, V).*

The following theorem, from which it follows that $\mathrm{UWZK}(P, V) \Rightarrow \neg\mathrm{MAGIC}(P, V)$, is immediate from the definitions.

Theorem 5.4 *Let (P, V) be any argument satisfying Definition 5.1. Then for all possible interaction-removing functions V^*, if the resulting noninteractive signature scheme satisfies the completeness condition, then its soundness completely fails: with overwhelming probability over X_n, Y_n, M_n, a forger can succeed with all but negligible probability.*

We note that Theorem 5.4 holds if P, V^*, T, D, and S, are chosen to be device types other than the types used above, provided that the same choice is made in Definitions 4.14, 5.1, and 5.2. For example, we could take S and T to be probabilistic ptime machines. Theorem 5.4 also holds if we reintroduce additional information into z (for example, history information) that depends on x and y, provided that this is done consistently in Definitions 4.14, 5.1, and 5.2. Formally, there is a joint distribution $D' = \{X_n Y_n H_n\}$ and an independent $M = \{M_n\}$ such that $D = \{X_n\, Y_n\, (H_n \circ M_n)\}$. It is clear that this definition is equivalent to Definition 4.13. Typically, an h produced by H_n represents a history of the verifier's past interactions.

Regarding that we are considering a more general methodology in which the test can be more general than the test in the original Fiat-Shamir methodology, we can weaken Definition 4.14 by restricting T to those tests in the original methodology, and change Definitions 5.1 and 5.2 in the same way. Both directions of $\mathrm{UWZK}(P, V) \Leftrightarrow \neg\mathrm{MAGIC}(P, V)$ hold in this context.

[6] One can also consider stronger versions: *e.g.*, there exists a magic function that works for all identification schemes, and it works on all distributions D_n, all messages m *etc.*

5.2 Adversarially Chosen Messages

In this subsection we consider a more powerful simulator/forger that may use a chosen message attack to produce a particular message m_0 (different than any of the messages m chosen by the forger in the attack) and an attempted signature on m_0. In this case, the simulator/forger S has one input x and two outputs, a message $S(x)|_M$ and a (forged) signature $S(x)|_F$. We use the restriction notation to emphasize that these two outputs are obtained using the same sequence of coin flips. In addition, S can query the "good signer" $(P, V^*)(x, y, m)$ as an oracle: after x and y have been fixed, each query from S to the oracle is a message m, and S receives a signature $(P, V^*)(x, y, m)$ chosen according to the coin flips of P. For a given x and y, denote the two outputs of S as $S^{(P,V)(x,y,\cdot)}(x)|_M$ and $S^{(P,V)(x,y,\cdot)}(x)|_F$. These two outputs are obtained using the same sequence of coin flips of S and P. This is an *adaptive* chosen message attack because the forger can choose each query m after seeing the signatures of the good signer on all previously chosen messages.

A restriction on S (which is obviously needed) is that the message m_0 on which S tries to forge cannot be identical to any of the messages that S used to query the oracle; formally, this event occurs with probability zero. We call a circuit (machine) of this type, that is also psize (ptime) and probabilistic, a *chosen-message-attack circuit* (*machine*). The analogue of Definition 5.1 for this model is (recall that M_n is independent of X_nY_n):

Definition 5.5 (Ultra-Weak Uniform ZK with Adversarily-Chosen Messages (over W)) *For all deterministic ptime verifiers V^*, for all deterministic ptime tests T such that*

$$\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] = 1 - \nu(n),$$

$\forall D = \{X_nY_nM_n\}$*, there exists a family S of chosen-message-attack circuits, such that*

$$|\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] \\ - \Pr[T(X_n, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_M, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_F)]\,| < \nu(n).$$

By the convention described in Section 3, the order of events in calculating the second probability is: (i) choose an x, y according to X_nY_n, (ii) run (the probabilistic) $S^{(P,V^*)(x,y,\cdot)}(x)$ to find a message m_0 and a signature ξ_0, and (iii) compute $T(x, m_0, \xi_0)$.

Note that in Definition 5.5, M_n plays no role in the probability that S successfully forges on the message of its choice; we could replace that definition with the equivalent:

Definition 5.6 *For all deterministic ptime verifiers V^*, for all deterministic ptime tests T such that* $\Pr[T(X_n, M_n, (P, V)(X_n, Y_n, M_n))] = 1 - \nu(n)$, $\forall D = \{X_nY_nM_n\}$*, there exists a family S of chosen-message-attack circuits, such that*

$$\Pr[T(X_n, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_M, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_F)] \geq 1 - \nu(n).$$

In this model, the soundness condition in Definition 5.2 becomes:

$$\Pr[T(X_n, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_M, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_F)] < \nu(n). \tag{5}$$

The following analogue of Theorem 5.4 is again immediate from the definitions. Intuitively, the theorem says that if the argument system is chosen-message-attack simulatable with oracle calls to $(P, V^*)(x, y, \cdot)$, then the signature scheme obtained via the Fiat-Shamir methodology is vulnerable to a chosen message attack.

Theorem 5.7 *Let (P, V) be any argument satisfying Definition 5.5. Then for all possible interaction-removing functions V^*, if the resulting noninteractive signature scheme satisfies the completeness condition, then with overwhelming probability over X_nY_n, a chosen-message-attack S can succeed with all but negligible probability.*

In the next section we note that the hard core result $(\neg\mathrm{UWZK}(P, V) \Rightarrow \mathrm{MAGIC}(P, V))$ holds also in the case of chosen message attack.

6 Magic Functions

As stated in Theorem 5.4, if (P, V^*) satisfies Definition 5.1, our weakest definition of zero knowledge (with messages), then P does not have a magic function that can be used to transform (P, V^*) into a signature scheme that meets both the completeness and the soundness properties of the Fiat-Shamir methodology (in fact, if completeness holds, then soundness must fail in a drastic way). In Section 6.1 we prove a partial converse to this result. We show that if P does not satisfy Definition 5.1, and the deterministic V^*, the deterministic T, and the distribution $D = \{X_nY_nM_n\}$ witness this fact, then V^* is a "magic function" on some hard core $\mathcal{C}$ of x's and some hard core $\mathcal{M}(x)$ of messages, where the densities of $\mathcal{C}$ and $\mathcal{M}(x)$ are at least $1/\mathrm{poly}(n)$. This result could be used in theory to produce a secure signature scheme, with signer (P, V^*), in the following way (described informally). Suppose there is some unknown simulator/forger $S = \{S_n\}$, where we know only a constant d such that S_n is implemented as a circuit of size n^d for all n. To find a "likely secure" public key x, use X_nY_n to sample a (public key, private key) $= (x, y)$, and test whether $x \in \mathcal{C}$ (another part of our result is that there are deterministic polynomial-size circuits that test membership in $\mathcal{C}$ and $\mathcal{M}(x)$). Continue getting samples of (x, y) until an $x \in \mathcal{C}$ is found, and use this (x, y) as the (public key, private key) of the signature scheme. Because the density of $\mathcal{C}$ is $1/\mathrm{poly}(n)$, this will take expected time $\mathrm{poly}(n)$. Similarly, to find "likely secure" messages for this (x, y), use M_n to sample m's; every m with $m \in \mathcal{M}(x)$ is "likely secure" (we discuss below how to sign *all* messages instead of only a polynomial fraction of the m's). The sense in which x and m are "likely secure" is that, with probability $1/\mathrm{poly}(n)$ close to 1 (over the choice of x and m), the probability (over the coin tosses of S) that S can forge a signature on m that will pass the test T is at most some small $1/\mathrm{poly}(n)$ amount.

Let us define

$$\begin{aligned} \pi_T(n) &\stackrel{\text{def}}{=} \Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))], \\ \pi_T^S(n) &\stackrel{\text{def}}{=} \Pr[T(X_n, M_n, S(X_n, M_n))]. \end{aligned} \tag{6}$$

Then the assumption that P does not satisfy Definition 5.1, and V^*, T, D witness this fact, means that: (i) the test T almost always accepts valid signatures, formally,

$$\pi_T(n) = 1 - \nu(n), \tag{7}$$

and (ii) for every simulator/forger S there is at least a $1/n^c$ gap between the probabilities that S successfully signs and (P, V^*) successfully signs, formally, for all probabilistic polynomial-size circuit families S, there exists a constant c and infinitely many n such that

$$|\pi_T(n) - \pi_T^S(n)| \geq 1/n^c. \tag{8}$$

The sense in which V^* is a magic function for n on a hard core is (the definition is given more formally below): for all numbers a, b, d (the size of S in the following will be n^d), there exists a number q such that:

- there exists a hard core $\mathcal{C}$ of x's ($\mathcal{C} \subseteq \Sigma_n^X$) containing at least an n^{-q} fraction of Σ_n^X,
- for each $x \in \mathcal{C}$ there is a hard core $\mathcal{M}(x)$ of messages ($\mathcal{M}(x) \subseteq \Sigma_n^M$) containing at least an n^{-q} fraction of Σ_n^M, and
- every probabilistic circuit simulator S_n of size n^d forges very badly on all but a n^{-b} fraction of the $x \in \mathcal{C}$ and the $m \in \mathcal{M}(x)$, in the sense that

$$\Pr[T(x, m, S_n(x, m))] < 1/n^a. \tag{9}$$

Together with the assumption (7), the inequality (9) means that (with a negligible number of exceptions) the difference between the probability that $(P, V^*)(x, y, m)$ is a correct (according to T) signature of m and the probability that $S(x, m)$ is a correct signature of m is at least $1 - 1/n^a - \nu(n)$.

We refer to the probability in (6) as $\pi_T(n)$ (P, V^*, D will always be clear from context). In the case (7) that $\pi_T(n) = 1 - \nu(n)$, we think of T as being a validity check on signatures: $T(x, m, \xi) = 1$ if ξ is a valid (according to T) signature on m given public key x, or $T(x, m, \xi) = 0$ if ξ is invalid. In the more general case, $\pi_T(n)$ is arbitrary. The first case is the interesting one for signature schemes; our results for this case are in Section 6.1. We also give weaker results in Section 6.3 for the general case. We include results for this case, despite their relative weakness, because in the definition of distinguishability by T, for example (8), there usually is no restriction that the probability on the left in (8) must be close to one.

The assumptions (7) and (8) directly imply that every probabilistic psize simulator must fail weakly on some weak fraction of the (x, m)'s; *i.e.*, if $c' > c$ then the probability that S successfully signs m (*i.e.*, $\Pr[T(x, m, S(x, m))]$) is at most $1 - 1/n^{c'}$ on some fraction $1/n^{c'}$ of the (x, m)'s. In contrast, our hard core result concentrates the hardness on some 1/poly dense set $\mathcal{C}$. On this set, every simulator of size n^d must fail strongly (succeed with probability at most $1/n^a$) on some strong $(1 - 1/n^b)$ fraction of $\mathcal{C}$. We can make "strong" be as strong as we want by increasing a, b, and d, but this decreases the density of $\mathcal{C}$.

We described informally how, if P does not satisfy Definition 5.1, and the deterministic V^*, the deterministic T, and the distribution $D = \{X_n Y_n M_n\}$ witness this fact, then V^*, the hard core $\mathcal{C}$ of x's, and the hard core $\mathcal{M}(x)$ of messages could be used in theory to produce a secure signature scheme, with signer (P, V^*). We now describe, still informally, how the signature scheme can be modified to provide signatures on *all* messages m, not just those in $\mathcal{M}(x)$.

The general idea is to

1. Define M_n to be strings of length, say, $10n$, rather than just of length n, and
2. Consider a valid signature of $m \in \{0, 1\}^n$ to be a signature on any $10n$-bit string with prefix m. The set $\mathcal{M}(x)$ is now a subset of $\{0, 1\}^{10n}$ (this needs some refinement).

Changing the definition of M_n in this way does not affect our proofs, so $\mathcal{M}(x)$ still has polynomial density.

What can we hope to gain by this modification? Previously, the message m could be out of the core, which would prevent us from signing it. Now we have many more ways of signing m: it is sufficient that for some suffix i we will have that $m \circ i$ is in the core so that signing m will not be impossible ("$\circ$" denotes concatenation). We could even be more optimistic and hope that for every m the fraction of i such that $m \circ i$ is in the core is roughly the overall density of the core. In this case, finding such a useful suffix i for an arbitrary message m is fast enough (i.e., the time it takes is roughly proportional to the inverse of the density of the core). Unfortunately, this straightforward padding idea cannot guarantee such strong properties. For every message m we could have that either all suffixes are useful or none of them is (which implies that the number of messages which can be signed is no larger than before). In fact, this is true for any deterministic way of associating the messages in $\{0,1\}^n$ with subsets of $\{0,1\}^{10n}$. We overcome this obstacle by using a randomized mapping based on 2-universal (also known as pair-wise independent) permutations [11]. We start by giving the relevant definitions.

A family $\mathcal{F}$ of permutations $f : R \to R$ is 2-universal if for all $z \neq w \in R$ and for all $a \neq b \in R$, $\Pr_{f \in \mathcal{F}}[f(z) = a \wedge f(w) = b] = 1/(|R| \cdot (|R| - 1))$.[7] A family $\mathcal{H}$ of functions $h : D \to R$ is universal if for all $z \neq w \in D$, $\Pr_{h \in \mathcal{H}}[h(z) = h(w)] \leq 1/|R|$. The property of these functions that we use is given by the Leftover Hash Lemma [30, 29]:

Lemma 6.1 *Let k, ℓ and e be three positive integers such that $3e < \ell < k$. Let $X \subseteq \{0,1\}^k$ be a set of at least 2^ℓ elements and x uniformly distributed in X. Let $\mathcal{H}$ be a family of universal hash functions, $h : \{0,1\}^k \mapsto \{0,1\}^{\ell-3e}$ for all $h \in \mathcal{H}$. Then, for all but a 2^{-e} fraction of $h \in \mathcal{H}$, the uniform distribution over $\{0,1\}^{\ell-3e}$ and $h(x)$ are of statistical distance of at most 2^{-e}.*

We can now return to our modification of the signature scheme aimed at enabling signatures of *all messages*. The public key x is augmented by a description of a permutation f chosen uniformly from a 2-universal family $\mathcal{F}$ of permutations on the set of $10n$-bit strings. The procedure for signing a message m becomes: sample $i \in \{0,1\}^{9n}$ until an i is found for which $f(m \circ i) \in \mathcal{M}(x)$. Call such an i *useful*. Let s be the result of applying the signing function, using signing key y, to the string $f(m \circ i)$. The signature on m is the triple (m, i, s).

We claim that, with overwhelming probability over the choice of function $f \in \mathcal{F}$, the expected time to find a useful i is polynomial in n. That is, for every n-bit message m, we are interested in the expected (over choice of f) fraction of i's for which $f(m \circ i) \in \mathcal{M}(x)$; we would like this fraction to be close to the density of $\mathcal{M}(x)$ *for every* message m.

First, let $g = f^{-1}$; that is, for all $a \in \{0,1\}^{10n}$, $g(a) = f^{-1}(a)$. (Note that the family of permutations g defined in this way is also a 2-universal family of permutations.) For each such g, define $h : \{0,1\}^{10n} \to \{0,1\}^n$ by letting $h(a)$ be the first n bits of $g(a)$ for all $a \in \{0,1\}^{10n}$. It is easy to verify that the family $\mathcal{H}$ of functions h defined in this way is universal. Because the same h could arise from several different f's, the probability distribution on $\mathcal{H}$ is not necessarily uniform.

Note that, since each $a \in \mathcal{M}(x)$ for which $h(a) = m$ corresponds to a unique $i \in \{0,1\}^{9n}$ for which $f(m \circ i) \in \mathcal{M}(x)$,

$$|\{i \,|\, f(m \circ i) \in \mathcal{M}(x)\}| = |\{a \in \mathcal{M}(x) | h(a) = m\}|$$

so it suffices to determine the size of $\{a \in \mathcal{M}(x) | h(a) = m\}$, or, equivalently, its density. For this, we apply the Leftover Hash Lemma with $k = 10n$, $e = 2n$, and $\ell = 7n$ and with $X = \mathcal{M}(x)$ (so that the lemma compares the uniform distribution on $\{0,1\}^{\ell-3e} = \{0,1\}^n$ and $h(a)$, $a \in_R \mathcal{M}(x)$).

[7] The family of all linear permutations is a canonical example for 2-universal permutations.

To apply the lemma we need $|\mathcal{M}(x)| \geq 2^\ell$, and, since $\mathcal{M}(x) \subseteq \{0,1\}^{10n}$ has polynomial density, we have $|\mathcal{M}(x)| > 2^{9n} > 2^\ell = 2^{7n}$, the condition is satisfied. For $m \in \{0,1\}^n$, the lemma implies that

$$\Pr_{h \in \mathcal{H}}[|\{a \in \mathcal{M}(x) \,|\, h(a) = m\}| < |\mathcal{M}(x)|(2^{-n} - 2^{-2n})] < 2^{-2n}.$$

Note that $|\mathcal{M}(x)|(2^{-n}-2^{-2n}) \geq \frac{|\mathcal{M}(x)|}{2^{n+1}} = 2^{9n}/poly(n)$. We therefore have for any particular message m that $\{i \,|\, f(m,i) \in \mathcal{M}(x)\}$ is of polynomial density in $\{0,1\}^{9n}$ with probability at least $(1-2^{-2n})$. Since the number of possible messages m is 2^n (i.e., much smaller than $1/2^{-2n}$) we can conclude that with overwhelming probability (i.e., $(1 - 2^n \cdot 2^{-2n}) = (1 - 2^{-n})$) this holds for every message m. Therefore, the modified signature schemes allows efficient signing of all messages.

In Section 6.2 we give a version of the hard core result for the model of Section 5.2, where the simulator gets to choose the message on which it tries to forge a signature. The proof is actually simpler in this case: the hard core $\mathcal{C}$ is a subset of $\Sigma_n^X \Sigma_n^Y$ (rather than just of Σ_n^X); the requirement for the hard core is that a chosen-message attacker should not be able to forge on a message that she chooses by using an (adaptive) chosen message attack. Since the simulator/forger chooses the message, the proof does not need a separate hard core of messages: hardness must work for any choice of message by the forger. However, the hard core result in Section 6.1 does not imply the one in Section 6.2, nor vice versa. For the result in Section 6.2 we strengthen both the assumption and the conclusion of the result in terms of the more powerful chosen-message-attack simulators.

In Section 6.3 we prove a related result for the case where (7) is not assumed to hold (the "general π_T" case), but we still assume (8) that every simulator must fail by at least $1/n^c$ in the aggregate when attempting to match $\pi_T(n)$. The meaning (9) of S_n forging very badly becomes

$$|\Pr[T(x, m, (P', V^*)(x, y, m))] - \Pr[T(x, m, S_n(x, m))]| > 1/2 - 1/n^a. \tag{10}$$

A weakness of this result, compared to the result in Section 6.1 where (7) is assumed, is that we establish (10) for a prover P' different than the original prover P (although V^*, T, D and M can remain the same) (so we get the existence of *some* kind of magic, but not exactly the magic conjectured by Fiat and Shamir). This change in prover is necessary in certain cases, because we give an example P, V^*, T, D (M can be null) where it is impossible to obtain a hard core result of the strength we want without changing P (even if we can change V^*, T and D). So the hard core result of Section 6.1 does not follow from that of Section 6.3. The new prover P' is closely related to the original prover P: Informally, given (x, y, m), the new prover P' simulates $(P, V^*)(x, y, m)$ for a polynomial number of random choices of P's coins (recall that V^* is deterministic) until it finds conversations meeting certain conditions; P' then probabilistically chooses one of these conversations to use. (The term $1/2 - 1/n^a$ in (10) can be replaced by $1 - 1/n^a$, if the prover P' is given one bit of advice for each n.)

For the hard core results in subsections 6.1–6.3, we also show that membership in $\mathcal{C}$ and in $\mathcal{M}(x)$ (given a particular x) can be decided by deterministic psize circuits. By combining these circuits with the distribution D, it follows that $\mathcal{C}$ and $\mathcal{M}(x)$ are samplable by probabilistic psize circuits. Because D produces (x, y) pairs, the circuit that decides whether $x \in \mathcal{C}$ can be used to filter out the (x, y) with $x \notin \mathcal{C}$.

The hard core works only on some infinite set of n's. This is the best we can do, assuming only the negation of Definition 5.1, because this negation is consistent with a simulator that forges perfectly (probability 1 on all (x, m)) on all but some infinite set I of n's. Clearly, we cannot have a hard core that works for $n \notin I$. If we further weaken Definition 5.1 by changing a.e. n to i.o. n, then we would get a.e. versions of the hard core results, but implications of Definition 5.1 to the Fiat-Shamir methodology would be weaker. To address this, we prove more general versions of the hard core results, that include both the a.e. and i.o. cases, and many others. In these results, the density of the hard cores $\mathcal{C}$ and $\mathcal{M}(x)$ with security parameter n depends on how well the "best" simulator can forge, *i.e.*, how close the absolute value in (8) is to zero, when the security parameter is n. Roughly, better simulation (absolute value closer to zero) means smaller density of $\mathcal{C}$ and $\mathcal{M}(x)$, and vice versa.

The proofs of our hard core results use a basic strategy that was used previously by Impagliazzo [31]. However, the hard core results in [31] are done in the framework of circuit complexity rather than interactive proof systems, and the basic strategy is modified in different ways in [31] and in this paper. In particular, our results do not follow as corollaries of the results in [31].

6.1 T is a Validity Test

Recall that Equation (7) says that $\pi_T(n)$ is very close to 1, while (8) says that there is a polynomial gap between $\pi_T(n)$ and $\pi_T^S(n)$. We take (7) as an assumption in this subsection. Note that (7) and (8) together imply that for all S there exists a c and infinitely many n such that

$$\Pr[T(X_n, M_n, S(X_n, M_n,))] \leq 1 - n^{-c}. \tag{11}$$

Given that (7) holds, we will use (11) in place of (8) because it is more succinct and to the point. The probability in (11) depends on D, T, S, and n, but not on P or V^*. However, the assumption (7) depends on properties of P and V^*.

We first state the result in a simple form, where the assumption is that P, V^*, T, D do not satisfy the negation of Definition 5.1 in the sense stated at the beginning of Section 6; the proof of this result is delayed because it follows as an easy corollary of a more general result that we state and prove later. Recall (from Section 3) that "fractions" and "densities" are with respect to the distribution D.

Theorem 6.2 *Let T be a deterministic ptime machine, and let $D = \{X_n Y_n M_n\}$ be a probabilistic ptime distribution. Assume that for all probabilistic psize circuit families S, there exists a constant $c > 0$ and infinitely many n such that $\Pr[T(X_n, M_n, S(X_n, M_n))] \leq 1 - n^{-c}$. Then for all numbers a, b, and d, there exists numbers p and q and an infinite set I of n's such that for each $n \in I$:*

1. *there exists a set $\mathcal{C} \subseteq \Sigma_n^X$ having density at least n^{-q} in Σ_n^X,*
2. *for all $x \in \mathcal{C}$, there exists a set $\mathcal{M}(x) \subseteq \Sigma_n^M$ having density at least n^{-q} in Σ_n^M, and*
3. *for all probabilistic circuits S_n of size n^d, there exists a set $\mathcal{H} = \mathcal{H}_{S_n} \subseteq \mathcal{C}$ having density at least $1 - n^{-b}$ in $\mathcal{C}$, such that for all $x \in \mathcal{H}$ there exists a set $\mathcal{H}'(x) \subseteq \mathcal{M}(x)$ having density at least $1 - n^{-b}$ in $\mathcal{M}(x)$, such that for all $x \in \mathcal{H}$ and all $m \in \mathcal{H}'(x)$,*

$$\Pr[T(x, m, S_n(x, m))] < n^{-a}.$$

Moreover, there are deterministic circuits $\widehat{C}(\cdot)$ and $\widehat{B}(\cdot, \cdot)$ of size n^p such that, for all x and m, $x \in \mathcal{C}$ iff $\widehat{C}(x) = 1$, and $m \in \mathcal{M}(x)$ iff $\widehat{B}(x, m) = 1$. It follows that $\mathcal{C}$ and $\mathcal{M}$ are samplable by probabilistic psize circuits.

Remarks

1. *Deterministic* T. In this hard core result, and the ones to follow, to obtain results of the strength we want it is essential that T be deterministic, as the following (contrived) example demonstrates. The example requires a reasonable assumption on the limits of psize circuit families. In the example, we can take Z_n and M_n to be null for simplicity. Let P be the prover that, on input $(x, y) \in W$, sends $\alpha = y$ as its first message. Define T by $T(x, \alpha\beta\gamma) = 1$ if $(x, \alpha) \in W$, and $\Pr[T(x, \alpha\beta\gamma)] = 1 - 1/n^k$ if $(x, \alpha) \notin W$. Our assumption on circuits is that every probabilistic psize circuit family S, given only x, has some difficulty finding a y such that $(x, y) \in W$; that is, for all but a $1/n^{k'}$ fraction of x's, we have $\Pr[W(x, S(x))] < n^{-k'}$, where $k' > k$. The verifier V^* can be arbitrary. This choice of (P, V^*, T, D) satisfies (7) and (11) (for any $c > k'$), but a hard core result as described above does not hold because $\Pr[T(x, S(x))] \geq 1 - 1/n^k$ for all x and S. So the "forges very badly" statement, $\Pr[T(x, S(x))] < 1/n^a$, does not hold even for one x and one S.

That S can be a circuit family is essential to the proof method. But other than these two requirements (T must be deterministic and S can be circuits) T, D, and M can be (independently) chosen to be either ptime machines or psize circuits (the same choice must be made in both the assumption and the conclusion).

2. *Dependence of the density of the hard core on the polynomial size of the simulator.* One might hope to find a hard core of density n^{-q} for some q that does not depend on the size of the simulator. This is impossible, given only the assumption in Theorem 6.2. This assumption allows the constant c to grow to arbitrarily large values as the size of S increases. When $c > q$, we can no longer have a hard core of density n^{-q}.

3. *Difficulty in finding separate hard cores for x's and messages.* In Theorem 6.2, the hard core $\mathcal{M}(x)$ of messages depends on x. Given a, b, d, one might hope to find a hard core $\mathcal{C}$ of x's and a hard core $\mathcal{M}$ of m's, both of $1/\text{poly}(n)$ density, such that, for every simulator S_n (of size n^d) there is a substantial fraction $H_{\mathcal{C}}$ of $\mathcal{C}$ and a substantial fraction $H_{\mathcal{M}}$ of $\mathcal{M}$ such that S_n forges very badly on all $(x, m) \in H_{\mathcal{C}} \times H_{\mathcal{M}}$. Imagine a bipartite graph B with bipartition Σ_n^X and Σ_n^M, and connect x and m by an edge iff S_n forges very badly on (x, m). If we could find $\mathcal{C}$ and $\mathcal{M}$ as above, then there would be a complete bipartite graph in B having almost a n^{-q} fraction of the vertices on each side. Given only the assumption in the theorem, and without further information on the structure of this bipartite graph, B could have edge density only n^{-c}. But even with edge density 1/2, a simple probabilistic argument shows that, for all N, there is a bipartite graph with N vertices on each side that contains no $k \times k$ complete bipartite subgraph if k is larger than about $2 \log N$. In our case, N is the number of x's and m's (for a fixed size n). Because the graph must have a complete bipartite subgraph with vertex density almost n^{-q}, it follows that $n^{-q} = O(k/N)$. So $q = (\log N - \log\log N - O(1))/\log n$. In the typical case that the number N of x's and m's is not bounded by a polynomial in n, there is no constant upper bound on q. Although this does not prove that separate hard cores are impossible, it does show that more information about the bipartite graph must be used in the proof.

4. *Histories.* For simplicity, we state and prove the hard core results without additional information, such as a history, included in the auxiliary input z given to the signer, test, and simulator. We mention two ways that "history" information can be reintroduced without changing the proofs in any significant way. In both cases, there is a joint distribution $D' = \{X_n Y_n H_n\}$ and an independent $M = \{M_n\}$, and $D = \{X_n \, Y_n \, (H_n \circ M_n)\}$. The first way is to assume that h is a ptime function of x and y; that is, there is a ptime computable function f such that $h = f(x, y)$ for every (x, y, h)

that can be produced by D'. The only way the results change in this case is that the hard core $\mathcal{C}$ is now a set of (x, y) pairs, and the hard cores of messages are now $\mathcal{M}(x, y)$ instead of $\mathcal{M}(x)$. The reason why y is now relevant to the hard core $\mathcal{C}$ is that the ability of a given S to forge depends on the auxiliary input h given to it, and h depends on y.

A second way is to allow $D' = \{X_n Y_n H_n\}$ to be an arbitrary ptime-samplable distribution. For example, for each (x, y), the component h could be the result of a chosen message attack on the signature scheme with (public key, private key) $= (x, y)$, where the attack is done by a probabilistic adversary. The way the results change in this case is that the hard core $\mathcal{C}$ is now a set of (x, y, h) triples, and the hard cores of messages are $\mathcal{M}(x, y, h)$. This is problematic when applied to signature schemes if we have only a weak assumption (as in Theorem 6.2) on the ability of simulators to forge in the aggregate. In this case, the density of $\mathcal{C}$ can be n^{-q} small, and this is consistent with the situation that, for all (x, y), only a small n^{-q} fraction of the attacks h are "safe" for (x, y). However, for signature schemes of practical value, we would want a stronger restriction on the ability of simulators to forge, as in the soundness condition in Definition 5.2. As we will see below, if the probability $1 - n^{-c}$ in the assumption in Theorem 6.2 is changed to n^{-c}, then the density of the hard core $\mathcal{C}$ becomes $1 - n^{-q}$, where q increases as c increases. It follows that for at least a $1 - n^{-q/2}$ fraction of the (x, y)'s there is a $1 - n^{-q/2}$ fraction of the h's (depending on (x, y)) such that $(x, y, h) \in \mathcal{C}$. Thus, for most of the (x, y), most of the attacks h are safe for (x, y).

We now state the more general result from which Theorem 6.2 follows. First we define a measure $E(d, n)$ of the aggregate error by which the "best" circuit simulator of size n^d fails to forge signatures when the security parameter is n. Let $\mathcal{F}(d, n)$ be the set of probabilistic circuits of size n^d that take as input an element of $\Sigma_n^X \Sigma_n^M$; note that $\mathcal{F}(d, n)$ is finite. Define $E(d, n)$ by:

$$\max_{S_n \in \mathcal{F}(d,n)} \Pr[T(X_n, M_n, S_n(X_n, M_n))] = 1 - E(d, n) \tag{12}$$

if the probability on the left is at most $1 - 2e^{-n}$; otherwise (the probability is $> 1 - 2e^{-n}$), define $E(d, n) = 0$. The second condition says that an error less than $2e^{-n}$ is so small that we take the error to be zero. This ensures that $E(d, n) \geq 2e^{-n}$ if $E(d, n) \neq 0$, which is a useful technical property in the proof. (The case $E(d, n) = 0$ is not relevant, because then the theorem is trivially true.) The function E also depends on T, D, and M, but because these will be clear from context they are not included in the notation.

The differences between Theorem 6.2 and Theorem 6.3 below are: in Theorem 6.3 there is no assumption about T, D, and M; and the densities of $\mathcal{C}$ and $\mathcal{M}(x)$ depend on $E(d', n)$, for a d' depending on a, b, and d. An intuitive description of the meaning of the theorem immediately follows the formal statement. All square-roots in this paper are positive square-roots.

Theorem 6.3 *Let T be a deterministic ptime machine, and let $D = \{X_n Y_n M_n\}$ be a probabilistic ptime distribution. There is a number k (depending only on T) and a number ℓ such that the following holds for all numbers a, b, and d, and all sufficiently large n. Let $\varepsilon = E(a + 2b + d + k, n) - e^{-n}$, let g be a number such that $0 < g \leq 1/4$, and let $\sigma = 1 - \sqrt{1 - \varepsilon} - g$.*

1. *There exists a set $\mathcal{C} \subseteq \Sigma_n^X$ having density at least σ in Σ_n^X,*

2. *for all $x \in \mathcal{C}$, there exists a set $\mathcal{M}(x) \subseteq \Sigma_n^M$ having density at least σ in Σ_n^M, and*

3. for all probabilistic circuits S_n of size n^d, there exists a set $\mathcal{H} \subseteq \mathcal{C}$ having density at least $1-n^{-b}$ in $\mathcal{C}$, such that for all $x \in \mathcal{H}$ there exists a set $\mathcal{H}'(x) \subseteq \mathcal{M}(x)$ having density at least $1-n^{-b}$ in $\mathcal{M}(x)$, such that for all $x \in \mathcal{H}$ and all $m \in \mathcal{H}'(x)$,

$$\Pr[T(x, m, S_n(x, m))] < n^{-a}.$$

Moreover, there is a deterministic circuit $\widehat{C}(\cdot)$ of size $O((n^{a+b} + n/g)^\ell)$ and a deterministic circuit $\widehat{B}(\cdot,\cdot)$ of size $O(n^{\ell(a+b)})$ such that for all x and m, $x \in \mathcal{C}$ iff $\widehat{C}(x) = 1$ and $m \in \mathcal{M}(x)$ iff $\widehat{B}(x, m) = 1$.

Before proving this result, we explain how the quantities ε, g and σ are used in two cases, and we prove Theorem 6.2 as a corollary. (The extreme generality of Theorem 6.3, in particular, the absence of any assumptions about T, D, and M, immediately also yields several analogues to Theorem 6.2.) The quantity ε is essentially the simulation error of the best simulator of size $n^{d'}$ where $d' = a + 2b + d + k$. If ε is small, say, $\varepsilon = n^{-r}$, then we can take $g = (1 - \sqrt{1-\varepsilon})/2$ and $\sigma = (1 - \sqrt{1-\varepsilon})/2$. So σ and g are at least $n^{-r}/4$. This says that the densities of hard cores $\mathcal{C}$ and $\mathcal{M}(x)$ are at least linear in the simulation error of the best circuit of size $n^{d'}$, and the circuits $\widehat{C}$ and $\widehat{B}$ are of polynomial size. On the other hand, if ε is large, say, $\varepsilon = 1 - n^{-r}$, then we can take $g = (\sqrt{1-\varepsilon})/n = n^{-(r/2+1)}$, and $\sigma = 1 - n^{-r/2} - n^{-(r/2+1)}$. So the densities of the hard cores are very large, and again the deciding circuits are of polynomial size.

Theorem 6.2, proved next, assumes that $\pi_T^S(n) \leq 1 - n^{-c}$, so ε may be small. An analogous result for the case where ε and σ are large holds if the assumption is changed to $\Pr[T(X_n, M_n, S(X_n, M_n))] \leq n^{-c}$.

Proof of Theorem 6.2.
Let T, D, M, a, b, d be given. Let $d' = a + 2b + d + k$, and $\varepsilon_n = E(d', n) - e^{-n}$.

We first show that it suffices to establish, for some constant r and infinitely many n, that $\varepsilon_n \geq n^{-r}$. Fix one of these infinitely many n. If $\varepsilon = \varepsilon_n > 1/n$, then take $g = 1/(4n)$ in Theorem 6.3. Then $\sigma = (1 - \sqrt{1-\varepsilon}) - g > \varepsilon/2 - g > 1/(2n)$. If $n^{-r} \leq \varepsilon_n \leq n^{-1}$, then we take $g = \sigma = (1 - \sqrt{1-\varepsilon})/2 \geq \varepsilon/4 \geq n^{-r}/4$. In either case, for numbers q and p depending on r, the densities of $\mathcal{C}$ and $\mathcal{M}(x)$ ensured by Theorem 6.3 are at least $\sigma \geq n^{-q}$, and the size poly(n/g) of the decider circuits is n^p.

We now establish that $\varepsilon_n \geq n^{-r}$, for some constant r and infinitely many n. For each n, let $S^*_{d',n}$ be a probabilistic circuit of size $n^{d'}$ such that

$$\Pr[T(X_n, M_n, S^*_{d',n}(X_n, M_n))] = 1 - E(d', n)$$

i.e., $S^*_{d',n}$ is a circuit that achieves the max in the definition (12) of $E(d', n)$. Let $S_{d'}$ be the circuit family $\{ S^*_{d',n} \mid n \geq 1 \}$. Applying the assumption of Theorem 6.2 to this family, there is a constant c and infinitely many n such that $\Pr[T(X_n, M_n, S^*_{d',n}(X_n, M_n))] \leq 1 - n^{-c}$. For these infinitely many n, we have $\varepsilon_n = E(d', n) - e^{-n} \geq n^{-c} - e^{-n}$. $\square$

Before beginning the proof of Theorem 6.3, we give a lemma that will be used several times. Let K be a probabilistic circuit with input u (in applications, u might represent an x or an (x, y) pair). Let $\Pr[K(u)]$ denote the probability that K outputs 1 on input u. Let τ_1, τ_2 be numbers with $0 \leq \tau_1 < \tau_2 \leq 1$. The deterministic circuit N *separates τ_1 and τ_2 for K* if for all u: if $\Pr[K(u)] \leq \tau_1$, then $N(u) = 0$; and if $\Pr[K(u)] \geq \tau_2$, then $N(u) = 1$. The proof of the following lemma is well-known (see, for example, the proof that BPP $\subseteq$ P/poly in [2]). We state the result in a simple form, which is sufficient for our purposes.

Lemma 6.4 *There is a number ℓ with the following property. For every probabilistic circuit K of size s and all numbers τ_1 and τ_2 with $0 \le \tau_1 < \tau_2 \le 1$, there is a deterministic circuit N of size $O((s/(\tau_2 - \tau_1))^\ell)$ such that N separates τ_1 and τ_2 for K.* □

Proof of Theorem 6.3.
At a high level the proof proceeds in two steps for a fixed n: first we show the existence of a hard core $\mathcal{B} \subseteq \Sigma_n^X \Sigma_n^M$, and a deterministic psize circuit $\widehat{B}$ that decides membership in $\mathcal{B}$. We then define $\mathcal{C}$ by putting x in $\mathcal{C}$ if $(x, m) \in \mathcal{B}$ for a sufficiently large fraction of the m's, and we show the existence of a deterministic circuit $\widehat{C}$ that decides membership in $\mathcal{C}$, and bound the size of $\widehat{C}$.

Fix a sufficiently large n. The meaning of "sufficiently large" is that certain inequalities in the proof may fail to hold for small n, and these small n are excluded. To prove the existence of the hard core $\mathcal{B}$ of pairs (x, m), we use a basic method used in [31], although we modify the method in order to get a psize circuit decider for $\mathcal{B}$.

In the basic method, the construction of $\mathcal{B}$ is done in steps. At each step, we have a collection $\mathcal{U}$ of simulator circuits of size n^d, and two sets G and H that partition $\Sigma_n^X \Sigma_n^M$ (*i.e.*, $H = \overline{G}$ with respect to $\Sigma_n^X \Sigma_n^M$). Intuitively, the set G contains the (x, m) that are good for some simulator in the collection; that is, some simulator in $\mathcal{U}$, when given (x, m), forges a signature on m with probability at least n^{-a}. On the complement H of G, no simulator in $\mathcal{U}$ forges this well. At a given step, we ask if there is some simulator U of size n^d such that some n^{-2b} fraction F of H is good for U. If so, we add U to $\mathcal{U}$, add F to G, remove F from H, and continue. If there is no such U, then H is the hard core. If the construction has not stopped after a certain polynomial number of steps, then we argue that the simulators in $\mathcal{U}$ can be combined into a "super-simulator" that contradicts the definition of the function E. Some modifications to this construction are needed so that we can use Lemma 6.4 to show the existence of a psize circuit decider for $\mathcal{B}$. The details follow.

The outcome of the proof is that the theorem holds with the density $1 - n^{-b}$ replaced by $1 - 3n^{-b}$ in item 3. This can be handled by starting the proof with a slightly larger $b' > b$. For simplicity, we carry out the proof using the original b, and absorb $(b' - b)$ into the constant k. Let $d' = a + 2b + d + k$. We can assume that $E(d', n) \ge 2e^{-n}$, because the theorem is trivially true if $E(d', n) = 0$, for then $\sigma < 0$ and $\mathcal{C}$ can be empty.

If $A' \subseteq A \subseteq \Sigma_n^X \Sigma_n^M$, let $\delta(A'|A)$ denote the density of A' in A under D_n and M_n, *i.e.*, the conditional probability that $(x, m) \in A'$ given that $(x, m) \in A$. Let $\delta(A')$ abbreviate $\delta(A' \mid \Sigma_n^X \Sigma_n^M)$. Define $\delta(A'|A)$ and $\delta(A')$ similarly if $A' \subseteq A \subseteq \Sigma_n^X$ or Σ_n^M.

The construction is described as a procedure FHC (Find Hard Core). In this procedure, G_i and H_i denote the sets described above after step i. There is a variable t in this procedure that is used as a threshold. For a simulator U, define $\mathrm{forge}_U(x, m)$ to be the probability that U forges on (x, m) (with respect to the test T):

$$\mathrm{forge}_U(x, m) = \Pr[T(x, m, U(x, m))].$$

Define $G_U(t)$ to be the set of (x, m) on which U forges with probability at least t:

$$G_U(t) = \{\, (x, m) \mid \mathrm{forge}_U(x, m) \ge t \,\}.$$

FHC (Find Hard Core)

1. Initialize $G_0 = \emptyset$, $H_0 = \Sigma_n^X \Sigma_n^M$, $\mathcal{U} = \emptyset$, $i = 0$, and $t = n^{-a}$.

2. If $\delta(H_i) < \varepsilon$ $(= E(d', n) - e^{-n})$ then stop. As shown below, if the procedure stops here, then we have a contradiction to the definition of $E(d', n)$.

3. If there exists a simulator U of size n^d such that

$$\delta(G_U(t) \cap H_i \,|\, H_i) \geq n^{-2b} \tag{13}$$

(in other words, U forges with probability at least t on some n^{-2b} fraction of the current hard set H_i), then let U be a simulator satisfying (13), and set $G_{i+1} \leftarrow G_i \cup G_U(t)$, $H_{i+1} \leftarrow H_i - G_U(t)$, and $\mathcal{U} \leftarrow \mathcal{U} \cup \{U\}$. Set $i \leftarrow i + 1$, and return to step 2.

4. At this point we know that there does not exist a U of size n^d satisfying (13). (*Comment*: If we did not want to show the existence of psize circuit deciders for the hard cores, we could stop here and use $\mathcal{B} = H_i$.) If $i = 0$ then set $H'_i = H_i$ and go to step 4b to stop. If $i > 0$, let $t' = t - \frac{1}{2}n^{-a-2b-1}$, and let $G'_i = \bigcup_{1 \leq j \leq i} G_{U_j}(t')$. Because $G_i = \bigcup_{1 \leq j \leq i} G_{U_j}(t)$ and $t' < t$ (so t' is a threshold that is easier for simulators to satisfy), it follows that $G_i \subseteq G'_i$. Let $H'_i = \overline{G'_i}$, so $H'_i \subseteq H_i$. There are now two cases, depending on whether or not the difference $H_i - H'_i$ is at least a n^{-2b} fraction of the current hard set H_i, and whether the density of H'_i satisfies the stopping condition in step 2.

   (a) If $\delta((H_i - H'_i) \,|\, H_i) \geq n^{-2b}$, then set $G_i \leftarrow G'_i$, $H_i \leftarrow H'_i$, and $t \leftarrow t'$, and return to step 2. If $\delta(H'_i) < \varepsilon$, then set $H_i \leftarrow H'_i$ and return to step 2 (in this case, the procedure will stop immediately when it returns to step 2; this is done for technical convenience in the proof).

   (b) Otherwise, stop. As shown below, we have found a hard core.

Claim 6.5 *1. FHC must stop after at most n^{2b+1} iterations.*

2. When FHC stops, the variable t satisfies $\frac{1}{2}n^{-a} \leq t \leq n^{-a}$.

3. If FHC stops at step 4b, then at this point:

   (a) $\delta(H'_i) \geq \varepsilon$,

   (b) $\delta(H'_i|H_i) \geq 1 - n^{-2b}$, and

   (c) for every probabilistic circuit simulator U of size n^d, there is a set $F \subseteq H_i$ with $\delta(F|H_i) \geq 1 - n^{-2b}$ such that $\text{forge}_U(x, m) < n^{-a}$ *for all $(x, m) \in F$.*

Proof. 1. At each iteration at least a n^{-2b} fraction is removed from H_i to obtain H_{i+1}. Thus, after n^{2b+1} iterations, $\delta(H_i) \leq (1 - n^{-2b})^{n^{2b+1}} < e^{-n} \leq E(d', n) - e^{-n}$, because we have assumed $E(d', n) \geq 2e^{-n}$. When the density of H_i falls to this point, FHC will stop at step 2.

2. The variable t is initially n^{-a}, and t never increases. The only place where t decreases is in step 4a. Note that when t is decreased by $\frac{1}{2}n^{-a-2b-1}$ in this step, either the set $H_i - H'_i$ removed from H_i is at least a n^{-2b} fraction of H_i, or FHC immediately stops when it returns to step 2. By an argument similar to part 1 above, t is decreased at most n^{2b+1} times. Therefore we always have $t \geq n^{-a} - \frac{1}{2}n^{-a-2b-1}n^{2b+1} = \frac{1}{2}n^{-a}$.

3(a). For FHC to reach step 4b, it must first have failed the test $\delta(H'_i) < \varepsilon$ in step 4a, assuming $i > 0$. If $i = 0$, then $H'_i = H_i = \Sigma_n^X \Sigma_n^M$, so $\delta(H'_i) = 1$.

3(b). For FHC to reach step 4b, it must first have failed the test $\delta((H_i - H_i') \mid H_i) \geq n^{-2b}$ in step 4a, assuming $i > 0$. Therefore, $\delta(H_i'|H_i) \geq 1 - n^{-2b}$. If $i = 0$, then $H_i' = H_i$.

3(c). For FHC to reach step 4b, it must first have failed the test in step 3; that is, for all U of size n^d, $\delta(G_U(t) \cap H_i \mid H_i) < n^{-2b}$. Taking $F = H_i - G_U(t)$ we have $\delta(F|H_i) \geq 1 - n^{-2b}$ and $\text{forge}_U(x, m) < t \leq n^{-a}$ for all $(x, m) \in F$. $\square$

The proof is now reduced to showing: (I) that FHC cannot stop at step 2 because this leads to a contradiction, and (II) when FHC stops at step 4b we can find the hard cores $\mathcal{C}$ and $\mathcal{M}(x)$ and circuits that decide membership for them.

We first show (I). Assume that FHC stops at step 2, and let i be the step number at which it stops. By Claim 6.5(1), $i \leq n^{2b+1}$. Consider the following "super-simulator" U^*.

Definition of U^*: Input (x, m).
For each of the i different $U \in \mathcal{U}$, run $U(x, m)$ for $2n^{a+1}$ trials using a new random choice for U's coins at each trial. If any of these $2in^{a+1}$ trials produces a conversation ξ such that $T(x, m, \xi) = 1$, then output one such ξ. Otherwise, output a conversation produced by an arbitrary one of the trials.

Because each $U \in \mathcal{U}$ has size n^d, and $|\mathcal{U}| \leq n^{2b+1}$, it is easy to see that U^* can be implemented by a circuit of size at most $n^{a+2b+d+k} = n^{d'}$, where k depends only on T. We now place a lower bound on the aggregate $\Pr[T(X_n, M_n, U^*(X_n, M_n))]$. For a given (x, m), if $(x, m) \in G_i$ then there exists a $U \in \mathcal{U}$ such that $\Pr[T(x, m, U(x, m))] \geq t \geq \frac{1}{2}n^{-a}$ (see Claim 6.5(2)). Because U^* runs U for $2n^{a+1}$ trials, it follows that, with probability $> 1 - e^{-n}$, U^* finds a ξ such that $T(x, m, \xi) = 1$. By the stopping condition 2, the (x, m) not belonging to G_i (*i.e.*, belonging to H_i) form a set of density at most $\varepsilon = E(d', n) - e^{-n}$. It follows that for an (x, m) randomly chosen (by D_n), U^* finds a ξ with $T(x, m, \xi) = 1$ with probability $> (1 - e^{-n}) - (E(d', n) - e^{-n}) = 1 - E(d', n)$. Because U^* has size $n^{d'}$, this contradicts the definition of $E(d', n)$.

The final part (II) of the proof is to show that if FHC stops at step 4b, then we can find $\mathcal{C}$ and $\mathcal{M}(x)$ and psize circuit deciders for them. We first consider the interesting case that $i > 0$ when it stops. We begin by defining the circuit $\widehat{B}$. We then define the hard core $\mathcal{B}$, mentioned in the first paragraph of the proof, by $\mathcal{B} = \{ (x, m) \mid \widehat{B}(x, m) = 1 \}$. Let i be the step number and let t and t' be the values of these variables when FHC stops (at step 4b), and recall that $t' = t - \frac{1}{2}n^{-a-2b-1}$. At this point, $\Sigma_n^X \Sigma_n^M$ is partitioned into three sets: H_i', $H_i - H_i'$, and $\overline{H_i}$ (recall $H_i' \subseteq H_i$). We define the circuit $\widehat{B}$ so that $\widehat{B}(x, m) = 0$ for all $(x, m) \in \overline{H_i}$ (which is the same as all $(x, m) \notin G_i$), and $\widehat{B}(x, m) = 1$ for all $(x, m) \in H_i'$. This ensures that, if $\mathcal{B}$ is the set of (x, m) such that $\widehat{B}(x, m) = 1$, then $H_i' \subseteq \mathcal{B} \subseteq H_i$.

Note that for all $(x, m) \in G_i$ there exists a $U \in \mathcal{U}$ such that $\text{forge}_U(x, m) \geq t$; and for all $(x, m) \in H_i'$ and all $U \in \mathcal{U}$ we have $\text{forge}_U(x, m) < t'$. For $U \in \mathcal{U}$, let K_U be a probabilistic psize circuit that outputs 1 with probability $\text{forge}_U(x, m)$ (note that all devices appearing in the definition of forge are either ptime or psize). By Lemma 6.4, for each $U \in \mathcal{U}$, there is a deterministic psize circuit $\widehat{B}_U(x, m)$ that separates t' and t for K_U. By the preceding discussion, if $\widehat{B}$ is defined as the NOR of $\widehat{B}_U$ over $U \in \mathcal{U}$, then $\widehat{B}$ has the desired properties.

The following claim states properties of $\mathcal{B}$ that are used later. The only fact used about $\mathcal{B}$ is that $H_i' \subseteq \mathcal{B} \subseteq H_i$.

Claim 6.6 *1.* $\delta(\mathcal{B}) \geq \varepsilon$.

2. *For every probabilistic circuit simulator U of size n^d, there is a set $F \subseteq \mathcal{B}$ such that $\delta(F|\mathcal{B}) \geq 1 - 2n^{-2b}$ and $\mathrm{forge}_U(x, m) < n^{-a}$ for all $(x, m) \in F$.*

Proof. Part 1 is immediate from Claim 6.5(3a) and $H_i' \subseteq \mathcal{B}$.

To prove part 2, first note that Claim 6.5(3b) and $H_i' \subseteq \mathcal{B} \subseteq H_i$ imply that $\delta(\mathcal{B}|H_i) \geq 1 - n^{-2b}$. For an arbitrary U of size n^d, let $F \subseteq H_i$ be given by Claim 6.5(3c). Because F has density at least $1 - n^{-2b}$ in H_i, and $\mathcal{B}$ has density at least $1 - n^{-2b}$ in H_i, it follows that $F \cap \mathcal{B}$ has density at least $1 - 2n^{-2b}$ in $\mathcal{B}$. $\square$

In the rest of the proof of Theorem 6.3 we will need only Claim 6.6 and the existence of a psize circuit decider $\widehat{B}$ for $\mathcal{B}$. For each $x \in \Sigma_n^X$, let $\mathcal{M}(x) = \{\, m \mid (x, m) \in \mathcal{B} \,\}$ and let $\mu(x)$ be the density of $\mathcal{M}(x)$ in Σ_n^M. Let $\delta_{\mathcal{B}} = \delta(\mathcal{B})$. Below, we define $\mathcal{C}$ by placing x in $\mathcal{C}$ iff $\mu(x)$ is large enough.

Recall the definition of σ from the statement of the theorem: $\sigma = 1 - \sqrt{1-\varepsilon} - g$, where $0 < g \leq 1/4$. We can assume that $\sigma \geq 0$, because otherwise the theorem holds trivially ($\mathcal{C}$ can be empty). By combining the psize circuit $\widehat{B}$ that decides $\mathcal{B}$ and the probabilistic ptime machine that samples M_n, there is a probabilistic psize circuit K such that, on input x, K outputs 1 with probability $\mu(x)$. Using Lemma 6.4, let $\widehat{C}$ be a circuit that takes input x and separates $\tau_1 = \sigma$ and $\tau_2 = \sigma + g$ for K. Note that $0 \leq \tau_1$ by assumption, and $\tau_2 = 1 - \sqrt{1-\varepsilon} \leq 1$ because $\varepsilon \leq 1$. Because $g = \tau_2 - \tau_1$, the size of $\widehat{C}$ is polynomial in n/g, as required. Define the hard core $\mathcal{C} \subseteq \Sigma_n^X$ by $\mathcal{C} = \{\, x \mid \widehat{C}(x) = 1 \,\}$.

It remains to show that $\mathcal{C}$ and all $\mathcal{M}(x)$ for $x \in \mathcal{C}$ have the properties 1, 2 and 3 in the theorem.

1. We show that $\delta(\mathcal{C}) \geq \sigma$. For any numbers p and q with $p, q > 0$ and $p + q - pq \leq \delta_{\mathcal{B}}$, we claim that $\delta(\{\, x \mid \mu(x) \geq q \,\}) \geq p$ (*i.e.*, at least a fraction p of the x's are related by $\mathcal{B}$ to at least a fraction q of the m' s). If this claim were not true, then $\delta_{\mathcal{B}}$ would be less than $(1-p)q + p = p + q - pq \leq \delta_{\mathcal{B}}$, a contradiction. If we set $p = q = \sigma + g = 1 - \sqrt{1-\varepsilon}$, then $p + q - pq = \varepsilon$. If we decrease p to σ, the value of $p + q - pq$ does not increase. Letting $p = \sigma$ and $q = \sigma + g$, we have $p + q - pq \leq \varepsilon \leq \delta_{\mathcal{B}}$ (see Claim 6.6(1)), and the conclusion that at least a fraction p of the x's have $\mu(x) \geq q$. ¿From $\mu(x) \geq q \ (= \tau_2)$ we have $\widehat{C}(x) = 1$, so $x \in \mathcal{C}$. It follows that $\delta(\mathcal{C}) \geq p = \sigma$.

2. We show that $\delta(\mathcal{M}(x)) \geq \sigma$ for all $x \in \mathcal{C}$. If $x \in \mathcal{C}$ then $\widehat{C}(x) = 1$, so $\delta(\mathcal{M}(x)) = \mu(x) \geq \tau_1 = \sigma$ follows from the definitions of K and $\widehat{C}$.

3. We show that for all probabilistic circuits S_n of size n^d, there exists a set $\mathcal{H} \subseteq \mathcal{C}$ having density at least $1 - 3n^{-b}$ in $\mathcal{C}$, such that for all $x \in \mathcal{H}$ there exists a set $\mathcal{H}'(x) \subseteq \mathcal{M}(x)$ having density at least $1 - 3n^{-b}$ in $\mathcal{M}(x)$, such that for all $x \in \mathcal{H}$ and all $m \in \mathcal{H}'(x)$, $\Pr[T(x, m, S_n(x, m))] < n^{-a}$.

Let $\mathcal{C}_{\mathrm{ext}} = \{\, (x, m) \in \mathcal{B} \mid x \in \mathcal{C} \,\}$, that is, the set $\mathcal{C}$ with each x extended with its $\mathcal{B}$-related messages. We first prove a lower bound on $\delta(\mathcal{C}_{\mathrm{ext}}|\mathcal{B})$. Recall that $\delta(\overline{\mathcal{C}}) \leq 1 - \sigma$ (because we have shown that $\delta(\mathcal{C}) \geq \sigma$), and for all $x \in \overline{\mathcal{C}}$, we have $\mu(x) < \tau_2 = \sigma + g$. Using $\delta_{\mathcal{B}} \geq \varepsilon$ and $\varepsilon/2 \leq 1 - \sqrt{1-\varepsilon} \leq \varepsilon$ for all $0 \leq \varepsilon \leq 1$, it follows that

$$\begin{aligned}
\delta(\mathcal{C}_{\mathrm{ext}} \,|\, \mathcal{B}) \geq 1 - \frac{(1-\sigma)(\sigma+g)}{\delta_{\mathcal{B}}} \quad &\geq \quad 1 - \frac{(\sqrt{1-\varepsilon}+g)(1-\sqrt{1-\varepsilon})}{\varepsilon} \\
&\geq \quad 1 + \frac{1-\varepsilon-\sqrt{1-\varepsilon}}{\varepsilon} - \frac{g(1-\sqrt{1-\varepsilon})}{\varepsilon} \\
&\geq \quad 1/2 \, - \, g.
\end{aligned}$$

The assumption $g \leq 1/4$ then gives $\delta(\mathcal{C}_{\mathrm{ext}}|\mathcal{B}) \geq 1/4$.

Given $U = S_n$, we know from Claim 6.6(2) that there is a set $F \subseteq \mathcal{B}$ such that $\delta(F|\mathcal{B}) \geq 1 - 2n^{-2b}$ and $\text{forge}_U(x,m) < n^{-a}$ for all $(x,m) \in F$. Because $\delta(\mathcal{C}_{\text{ext}}|\mathcal{B}) \geq 1/4$, it follows that $\delta(F \cap \mathcal{C}_{\text{ext}} \,|\, \mathcal{C}_{\text{ext}}) \geq 1 - 8n^{-2b}$. Recall that by definition, $\mathcal{M}(x) = \{\, m \mid (x,m) \in \mathcal{B} \,\}$. For each $x \in \mathcal{C}$, let $\mathcal{H}'(x) = \{\, m \in \mathcal{M}(x) \mid (x,m) \in F \,\}$. Let $p' = q' = 1 - 3n^{-b}$. We now argue similarly to above (for p and q) that there exists an $\mathcal{H}$ having density at least p' in $\mathcal{C}$, such that for all $x \in \mathcal{H}$, the set $\mathcal{H}'(x)$ has density at least q' in $\mathcal{M}(x)$. If such a set $\mathcal{H}$ does not exist, then $\delta(F \cap \mathcal{C}_{\text{ext}} \,|\, \mathcal{C}_{\text{ext}})$ would be less than $(1-p')q' + p' = 1 - 9n^{-2b}$, which contradicts the fact that $\delta(F \cap \mathcal{C}_{\text{ext}} \,|\, \mathcal{C}_{\text{ext}}) \geq 1 - 8n^{-2b}$. This completes the proof that $\mathcal{C}$ and $\mathcal{M}(x)$ have property 3 in the theorem (with n^{-b} replaced by $3n^{-b}$).

There is only the case $i = 0$ left. If $i = 0$ when the procedure stops at step 4b, then take $\mathcal{B} = \Sigma_n^X \Sigma_n^M$, $\mathcal{C} = \Sigma_n^X$, and $\mathcal{M}(x) = \Sigma_n^M$. So the circuits $\widehat{C}$ and $\widehat{B}$ are trivial. □

Remark 6.7 *Define the average m-density of $\mathcal{C}$ to be $\sum_{x \in \mathcal{C}} p(x)\mu(x)$, where $p(x)$ is the probability of x under D_n. It follows from the statement of Theorem 6.3 that the average m-density of $\mathcal{C}$ is at least σ^2; this is fine if σ is close to one. In the case that σ and ε are close to zero, a better (than σ^2) lower bound on the average m-density of $\mathcal{C}$ is $\varepsilon(1/2 - g)$. After noting that the average m-density is exactly $\delta(\mathcal{C}_{\text{ext}})$, this lower bound is immediate from the bounds $\delta(\mathcal{C}_{\text{ext}}|\mathcal{B}) \geq 1/2 - g$ and $\delta(\mathcal{B}) \geq \varepsilon$. The average m-density is also the probability that an (x,m), chosen randomly according to X_n and M_n, satisfies $x \in \mathcal{C}$ and $m \in \mathcal{M}(x)$. From this, it is easy to place an upper bound on the average m-density. Precisely, fix an arbitrary T, D, and n. If the sets $\mathcal{C}$ and $\mathcal{M}(x)$ for $x \in \mathcal{C}$ satisfy the properties stated in Theorem 6.3 for this T, D, n, and if these sets have average m-density $\overline{\mu}$, then $\overline{\mu} = O(E(d,n))$. This can be compared to the lower bound $\overline{\mu} = \Omega(E(d',n))$, the principal difference being that $d' > d$.*

6.2 Adversarially Chosen Messages

In section 5.2 we defined a uniform model of security for signature schemes, where the adversary may use a chosen message attack to find a message m on which it tries to forge. There is an analogous version of the hard core result for this model. For simplicity, we state the analogous version of Theorem 6.2; the analogous version of Theorem 6.3 can be obtained similarly. The next theorem is similar to Theorem 6.2 except that both the assumption and the conclusion are strengthened to the case of chosen-message-attack simulators, and there is no separate hard core of messages.

Theorem 6.8 *Let P be a probabilistic ptime machine, let V^* and T be deterministic ptime machines, and let $D = \{X_n Y_n\}$ be a probabilistic ptime distribution.*

Assume that for all chosen-message-attack circuit families S, there exists a constant $c > 0$ and infinitely many n such that $\Pr[T(X_n, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_M, S^{(P,V^*)(X_n,Y_n,\cdot)}(X_n)|_F)] \leq 1 - n^{-c}$. Then for all numbers a, b, and d, there exist numbers p and q and an infinite set I of n's such that for each $n \in I$:

1. *there exists a set $\mathcal{C} \subseteq \Sigma_n^X \Sigma_n^Y$ having density at least n^{-q} in $\Sigma_n^X \Sigma_n^Y$,*
2. *for all chosen-message-attack circuits S_n size n^d, there exists a set $\mathcal{H} \subseteq \mathcal{C}$ having density at least $1 - n^{-b}$ in $\mathcal{C}$, such that for all $(x,y) \in \mathcal{H}$*

$$\Pr[T(x, S^{(P,V^*)(x,y,\cdot)}(x)|_M, S^{(P,V^*)(x,y,\cdot)}(x)|_F)] < n^{-a}.$$

Moreover, there is a deterministic circuit $\widehat{C}(\cdot,\cdot)$ of size n^p such that for all x and y, $(x,y) \in \mathcal{C}$ iff $\widehat{C}(x,y) = 1$.

Proof. The proof is a small modification to part of the proof of Theorem 6.3, combined with the proof of Theorem 6.2 as a corollary of Theorem 6.3. The new proof is simpler because there are not separate hard cores of messages. Instead of first constructing the hard core $\mathcal{B}$ and then deriving $\mathcal{C}$ and $\mathcal{M}(x)$ from $\mathcal{B}$, now $\mathcal{C}$ is constructed directly in virtually the same way that $\mathcal{B}$ was constructed. In the four-step procedure, we begin in step 1 with $H_0 = \Sigma_n^X \Sigma_n^Y$. In step 2, the stopping condition is $\delta(H_i) < n^{-q}$ for some sufficiently large constant q. In steps 3 and 4, we redefine $G_U(t)$, where U is now a chosen-message-attack circuit, by

$$G_U(t) = \{\, (x,y) \mid \Pr[T(x, U^{(P,V^*)(x,y,\,\cdot\,)}(x)|_M, U^{(P,V^*)(x,y,\,\cdot\,)}(x)|_F)] \geq t \,\}.$$

It is proved as before that the procedure cannot stop at step 2 (otherwise there would be a super-simulator, now a chosen-message-attack circuit, that contradicts the assumption of the theorem). When the procedure stops at step 4b, we define the circuit $\widehat{C}$ and the hard core $\mathcal{C}$ where $H_i' \subseteq \mathcal{C} \subseteq H_i$ in virtually the same way that $\widehat{B}$ and $\mathcal{B}$ were defined. One minor difference is that the elements of $\mathcal{C}$ (resp., $\mathcal{B}$) and the inputs to $\widehat{C}$ (resp., $\widehat{B}$) are (x,y) pairs (resp., (x,m) pairs). Another difference is that $\widehat{C}$ is defined in terms of deterministic separators for K_U, where $K_U(x,y)$ outputs 1 with probability

$$\Pr[T(x, U^{(P,V^*)(x,y,\,\cdot\,)}(x)|_M, U^{(P,V^*)(x,y,\,\cdot\,)}(x)|_F)].$$

Because the input to K_U is (x,y), K_U can simulate the oracle $(P,V^*)(x,y,\,\cdot\,)$. Thus, K_U is a psize probabilistic circuit. $\square$

6.3 General T

In this section we consider the case where $\pi_T(n) = \Pr[T(X_n, M_n, (P,V^*)(X_n, Y_n, M_n))]$ can have any value, not necessarily close to 1, and possibly a different value for each n. The assumption of the hard core result in this case is (informally) that for every simulator S there is a $c > 0$ such that S cannot approximate $\pi_T(n)$ to within n^{-c}. The intuitive meaning of the main theorem here is that, if P is a prover whose conversations (with some V^*) cannot be very well simulated in the aggregate, then there exists a P' (closely related to P in a way made precise below) whose converstions (with the same V^*) have a hard core for simulation.

We first give an example showing that, in this case, we must change the prover to obtain a hard core result of the strength we want. Let D be a distribution such that it is hard for a psize simulator, when given x, to find a y such that $W(x,y)$. The message distribution M can be null, and V^* can be arbitrary. The test T is defined by $T(x, \alpha\beta\gamma) = 1$ iff $W(x,\alpha)$. On input (x,y), the prover P sends $\alpha = y$ with probability n^{-c} and sends $\alpha = 0$ otherwise, where we assume that $\neg W(x,0)$ for all x; during the third round, P sends $\gamma = 0$. It is clear that $\pi_T(n) = n^{-c}$, and $\Pr[T(X_n, S(X_n))]$ is close to zero by assumption. So the assumption of the hard core result holds. Let T', D', V' be arbitrary. Define the simulator S by: On input x, S simulates V' on input $\alpha = 0$ to obtain a β; then S outputs the conversation $0\beta 0$. If P is the prover, this is correct with probability $1 - n^{-c}$. So for all (x,y), $|\Pr[T'(x, (P,V')(x,y))] - \Pr[T'(x, S(x))]| \leq n^{-c}$. Thus, there is no subset of $\Sigma_n^X \Sigma_n^Y$ having non-zero density (wrt D') on which the distance between $\Pr[T'(x, (P,V')(x,y))]$ and $\Pr[T'(x, S(x))]$ is large (e.g., bounded below by a positive constant) as we want of a hard core result.

Because the original prover P is changed to another prover P', the hard core result becomes stronger as stronger restrictions are placed on P'. Roughly, in the following definition (which is a little more general than we need), the new prover P' is restricted to receiving a polynomial number of samples of $T[x, m, (P, V^*)(x, y, m)]$, i.e., bits $b_1, b_2, ..., b_\ell$, where the randomization is over the random choices of P. Then P' can do a probabilistic ptime computation on these bits and request, by specifying i, the conversations ξ_i that (P, V^*) produced during the i-th sample. The new prover is not given x, y, or m explicitly, nor is it given ξ_j if $j \neq i$.

Definition 6.9 *Let P be a probabilistic interactive device that takes input (x, y), let V^* be a deterministic interactive device that takes input (x, m), and let T be a deterministic device that takes input (x, m, ξ) and outputs either 0 or 1. The device P' is a* poly-(P, V^*, T)-repeater *if P', using P, V^*, T as oracles, operates as follows on inputs x, y, and m. First, for a polynomial n^k number of independent samples $\xi_i = (P, V^*)(x, y, m)$ for $1 \leq i \leq n^k$, P' receives $t_i = T(x, m, \xi_i)$ for all i. P' then performs a probabilistic polynomial-time computation on $(t_1, \ldots, t_{n^k})$ to choose an i_0 with $1 \leq i_0 \leq n^k$. (P' cannot use x, y, m or the ξ_i in this computation.) Then P' is given $\xi_{i_0} = \alpha_0\beta_0\gamma_0$. To carry out the protocol, P' sends α_0 as the first message. After β_0 is received (as it must be because V^* is deterministic), P' sends γ_0 as the third message.*

When (P', V^*) is used as a signer, it is obvious how (P', V^*) would operate to produce a signature $\alpha_0\beta_0\gamma_0$, given (x, y, m).

We next state and prove an analogue of Theorem 6.3, that is, $\neg\mathrm{UWZK}(P, V) \rightarrow$ MAGIC. Main differences in the new theorem are that $E(d, n)$ and ε are defined differently, the hard core $\mathcal{C}$ is a set of (x, y) pairs, and the conclusion is somewhat weaker.

First we must define a version of the "error function" $E(d, n)$ for the general π_T case. To simplify the statement of the new theorem, we assume that there is a constant c such that any error smaller than n^{-c} is treated as zero. We can choose c as large as we like, but this increases the polynomial number of samples used by the new prover P'. For a given P, V^*, T, and D, define the function $E'_c(d, n)$ by:

$$E'_c(d, n) = \min_{S_n \in \mathcal{F}(d,n)} |\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] - \Pr[T(X_n, M_n, S_n(X_n, M_n))]|$$

if the quantity on the right is at least n^{-c}; otherwise (the quantity is $< n^{-c}$) define $E'_c(d, n) = 0$.

The following notation will be used in the next proof, and similar notation will be used in Section 8. For a given P, P', V^*, D:

$$\pi_T(n) = \Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] \quad \pi_T(x, y, m) = \Pr[T(x, m, (P, V^*)(x, y, m))]$$

$$\pi'_T(n) = \Pr[T(X_n, M_n, (P', V^*)(X_n, Y_n, M_n))] \quad \pi'_T(x, y, m) = \Pr[T(x, m, (P', V^*)(x, y, m))]$$

$$\pi^S_T(n) = \Pr[T(X_n, M_n, S(X_n, M_n))] \qquad \pi^S_T(x, y, m) = \Pr[T(x, m, S(x, m))].$$

Theorem 6.10 *Let P be a probabilistic ptime machine, let V^* and T be deterministic ptime machines, and let $D = \{X_n Y_n M_n\}$ be a probabilistic ptime distribution. There is a number ℓ such that, for all numbers a, b, c, and d, there is a number d' (depending on a, b, d and T, but not on c) such that the following holds for all sufficiently large n. Let $\varepsilon = E'_c(d')/5$, let g be a number such that $0 < g \leq 1/4$, and let $\sigma = 1 - \sqrt{1 - \varepsilon} - g$.*

1. There exists a poly-(P, V^*, T)-repeater P';

2. there exists a set $\mathcal{C} \subseteq \Sigma_n^X \Sigma_n^Y$ having density at least σ in $\Sigma_n^X \Sigma_n^Y$;

3. for all $(x, y) \in \mathcal{C}$, there exists a set $\mathcal{M}(x, y) \subseteq \Sigma_n^M$ having density at least σ in Σ_n^M; such that

4. for all probabilistic circuits S_n of size n^d, there exists a set $\mathcal{H} \subseteq \mathcal{C}$ having density at least $1 - n^{-b}$ in $\mathcal{C}$, such that for all $(x, y) \in \mathcal{H}$ there exists a set $\mathcal{H}'(x, y) \subseteq \mathcal{M}(x, y)$ having density at least $1 - n^{-b}$ in $\mathcal{M}(x, y)$, such that for all $(x, y) \in \mathcal{H}$ and all $m \in \mathcal{H}'(x, y)$,

$$|\Pr[T(x, m, (P', V^*)(x, y, m))] - \Pr[T(x, m, S_n(x, m))] > 1/2 - n^{-a}. \tag{14}$$

Moreover: (i) there is a deterministic circuit $\widehat{C}(\cdot, \cdot)$ of size $O((n^c + n^{a+b} + n/g)^\ell)$ and a deterministic circuit $\widehat{B}(\cdot, \cdot, \cdot)$ of size $O((n^c + n^{a+b})^\ell)$ such that for all x, y, m, $(x, y) \in \mathcal{C}$ iff $\widehat{C}(x, y) = 1$ and $m \in \mathcal{M}(x, y)$ iff $\widehat{B}(x, y, m) = 1$; and (ii) if P' is given one bit of advice for each n, then the quantity $1/2 - n^{-a}$ in (14) can be replaced by $1 - n^{-a}$.

The following analogue of Theorem 6.2 can be obtained as a corollary of Theorem 6.10 in the same way that Theorem 6.2 was obtained from Theorem 6.3. The intuitive meaning of the theorem is the following. Let Definition 5.0 be a definition of zero-knowledge that is identical to (ultra-weak) Definition 5.1, except that the test T is not restricted by $\pi_T(n) = 1 - \nu(n)$. Intuitively, the theorem says that if P does not satisfy Definition 5.0 and V^*, T, D witness this fact, then this system, with P replaced by some poly-(P, V^*, T)-repeater P', has a hard core for simulation.

Theorem 6.11 *Let P be a probabilistic ptime machine, let V^* and T be deterministic ptime machines, and let $D = \{X_n Y_n M_n\}$ be a probabilistic ptime distribution. Assume that for all probabilistic psize circuit families S, there exists a constant $c > 0$ and infinitely many n such that*

$$|\Pr[T(X_n, M_n, (P, V^*)(X_n, Y_n, M_n))] - \Pr[T(X_n, M_n, S(X_n, M_n))]| \geq n^{-c}.$$

Then for all numbers a, b, and d, there exist numbers p and q and an infinite set I of n's such that for each $n \in I$, statements 1, 2, 3, and 4 of Theorem 6.10 hold with σ replaced by n^{-q} in 2 and 3. Moreover: (i) there are deterministic circuits $\widehat{C}(\cdot, \cdot)$ and $\widehat{B}(\cdot, \cdot, \cdot)$ of size n^p, such that for all x, y, m, $(x, y) \in \mathcal{C}$ iff $\widehat{C}(x, y) = 1$ and $m \in \mathcal{M}(x, y)$ iff $\widehat{B}(x, y, m) = 1$; and (ii) if P' is given one bit of advice for each n, then the quantity $1/2 - n^{-a}$ in (14) can be replaced by $1 - n^{-a}$.

Proof of Theorem 6.10.
The proof is based on the proof of Theorem 6.3, and familiarity with that proof is assumed. This time, we will find a hard core for (P', V^*) by contradiction to the definition of error $E'_c(d', n)$ for the original P. The number d' is chosen so that the size of a certain "super-simulator" (described below) is less than $n^{d'}$, where the size depends only on a, b, d, and T. If $E'_c(d', n) = 0$ for a particular n, then the proof is immediate for that n because then $\sigma \leq 0$ and $\mathcal{C}$ can be empty. Therefore, by definition of E'_c we can assume that $E'_c(d', n) \geq n^{-c}$.

The proof of Theorem 6.3 must be modified because in that proof there was a concrete notion of a simulator S "forging very badly" on (x, y, m), namely, that $\pi_T^S(x, y, m)$ is far from 1. In this proof, S is trying to make this probability close to $\pi_T(x, y, m)$, and this latter probability, rather than being very close to 1, can have any value in $[0, 1]$, and the value can depend on (x, y, m).

In order to get some handle on the value of $\pi_T(x,y,m)$, for a fixed n the (x,y,m) are divided into three sets, A_0, A_1, and A_2, where A_0 (resp., A_1, A_2) is (approximately) the set of (x,y,m) such that $\pi_T(x,y,m)$ is less than n^{-p} (resp., greater than $1-n^{-p}$, between n^{-p} and $1-n^{-p}$), where p is such that $n^{-p} \le E'_c(d',n)/10$. Because $E'_c(d',n) \ge n^{-c}$, this can always be achieved with $p \le c+1$. Because we will need psize circuit deciders for these three sets, we allow a little uncertainty around the boundaries n^{-p} and $1-n^{-p}$. Let $K(x,y,m)$ be a probabilistic circuit that produces a random sample of $\pi_T(x,y,m)$. Using Lemma 6.4, let $\widehat{C}_0$ (resp., $\widehat{C}_1$) be a deterministic psize circuit that separates $n^{-p}/2$ and n^{-p} (resp., separates $1-n^{-p}$ and $1-n^{-p}/2$) for K. Define

$$\begin{aligned} A_0 &= \{ (x,y,m) \mid \widehat{C}_0(x,y,m) = \widehat{C}_1(x,y,m) = 0 \} \\ A_1 &= \{ (x,y,m) \mid \widehat{C}_0(x,y,m) = \widehat{C}_1(x,y,m) = 1 \} \\ A_2 &= \{ (x,y,m) \mid \widehat{C}_0(x,y,m) = 1 \text{ and } \widehat{C}_1(x,y,m) = 0 \}. \end{aligned}$$

The proof of the following claim is immediate from the definitions of the sets A_0, A_1, A_2, and the circuits $\widehat{C}_0$ and $\widehat{C}_1$. Part 1 also uses $n^{-p} < 1/2$, from which it follows that there does not exist an (x,y,m) with $\widehat{C}_0(x,y,m) = 0$ and $\widehat{C}_1(x,y,m) = 1$.

Claim 6.12 *1. A_0, A_1, A_2 partition $\Sigma_n^X \Sigma_n^Y \Sigma_n^M$.*

2. If $(x,y,m) \in A_0$, then $\pi_T(x,y,m) \le n^{-p}$.

3. If $(x,y,m) \in A_1$, then $1-n^{-p} \le \pi_T(x,y,m)$.

4. If $(x,y,m) \in A_2$, then $n^{-p}/2 \le \pi_T(x,y,m) \le 1-n^{-p}/2$.

We next define the poly-(P,V^*,T)-repeater P'. The idea is that P' is trying to find conversations ξ^0, ξ^1 such that $T(x,m,\xi^z) = \xi^z$ for $z = 0,1$, by sampling n^{p+1} conversations $(P,V^*)(x,y,m)$. If all sampled conversations ξ have the same $T(x,m,\xi)$, then P' uses any one of them.

Definition of P':
Receive n^{p+1} samples $t_i = T(x,m,\xi_i)$ where $\xi_i = (P,V^*)(x,y,m)$. If all t_i have the same value, then request and use ξ_1. If z, z' are such that $t_z = 0$ and $t_{z'} = 1$, then choose $r = z$ or $r = z'$ with probability $1/2$ each, and request and use ξ_r.

The following claim, giving properties of P', is immediate from Claim 6.12 and the definition of P'.

Claim 6.13 *1. If $(x,y,m) \in A_0$, then $\pi'_T(x,y,m) \le 1/2 + \nu(n)$.*

2. If $(x,y,m) \in A_1$, then $\pi'_T(x,y,m) \ge 1/2 - \nu(n)$.

3. If $(x,y,m) \in A_2$, then $\pi'_T(x,y,m) = 1/2 \pm \nu(n)$.

We next describe the "super-procedure" used to find the hard core. Let FHC_0 be the four-step procedure FHC in the proof of Theorem 6.3. The subscript 0 here means that the procedure finds a hard core such that every simulator of size n^d is "stuck near 0" on a large fraction of the hard core. Because we are now considering sets of (x,y,m) triples (rather than (x,m) pairs), $G_U(t)$ is slightly redefined as

$$G_U(t) = \{ (x,y,m) \mid \pi_T^U(x,y,m) \ge t \}.$$

The procedure FHC_1 is identical to FHC_0 except that it finds a hard core on which every simulator is "stuck near 1". This is done by replacing $G_U(t)$ by

$$G'_U(t) = \{\, (x, y, m) \mid \pi^U_T(x, y, m) \le 1 - t \,\}.$$

The super-procedure has four super-steps, S1–S4. At each super-step, it uses either FHC_0 or FHC_1 as a subprocedure, starting initially with a different H_0. In each case the super-procedure is trying to find a hard core within H_0 such that (i) the density of the hard core (in $X_n Y_n M_n$) is sufficiently large, and (ii) every simulator is either "stuck near 0" (for super-steps 1 and 3) or "stuck near 1" (for super-steps 2 and 4). In each case, H_0 is chosen so that every simulator is stuck at a value that is almost 1/2 away from the true value $\pi'_T(x, y, m)$. For example, when $H_0 = A_1$ as in super-step 1, we know from Claim 6.13 that $\pi'_T(x, y, m) \ge 1/2 - \nu(n)$ for all (x, y, m) in H_0. If every simulator is stuck near zero (specifically $\pi^U_T(x, y, m) \le n^{-a}$) in a hard core that is contained in H_0, then every simulator is at least $1/2 - n^{-a} - \nu(n)$ away from the true value in the hard core. The super-procedure also maintains a set of simulators $\mathcal{U}^{(j)}$ for $j = 1, 2, 3, 4$; whenever the subprocedure (FHC_0 or FHC_1) in super-step Sj adds a simulator U to $\mathcal{U}$ in its Step 3, the super-procedure adds U to $\mathcal{U}^{(j)}$. If any one of the four super-steps finds a hard core, then the super-procedure has found a hard core. If none of the four super-steps finds a hard core, we show that the simulators in the sets $\mathcal{U}^{(j)}$ can be combined to form a super-simulator that contradicts the definition of E'_c. When the subprocedure in superstep Sj ($1 \le j \le 4$) stops at iteration i, either in step 2 or step 4b, let $G^{(j)}$ (resp., $H^{(j)}$, $H^{(j)'}$) denote G_i (resp., H_i, H'_i). The four super-steps are similar. We describe the first super-step in more detail than the following ones.

S1. Run FHC_0 starting with $H_0 = A_1$ (we are trying to find a hard core inside A_1, where by definition $\pi'_T(x, y, m) \ge 1/2 - \nu(n)$, on which simulators are "stuck near 0" and hence are far from $\pi'_T(x, y, m)$). Note that in Steps 2 and 4 of FHC, $\delta(H_i)$ is now interpreted as $\delta(H_i|\Sigma^X_n \Sigma^Y_n \Sigma^M_n)$ (This applies to all the other super-steps as well.)

   (a) FHC_0 stops at step 4b. Note that Claim 6.5 holds with part 3(c) changed to: for every probabilistic circuit simulator U of size n^d, there is a set $F \subseteq H^{(1)}$ with $\delta(F|H^{(1)}) \ge 1 - n^{-2b}$ such that $\pi^U_T(x, y, m) < n^{-a}$ for all $(x, y, m) \in F$. Define the circuit $\widehat{B}$ from $H^{(1)}$ and $H^{(1)'}$ as before, and let $\widehat{B}^{(1)}$ be the conjunction of this circuit with $\widehat{C}_0 = \widehat{C}_1 = 1$ (this essentially restricts $\widehat{B}$ to those (x, y, m) in the starting set A_1). Define $\mathcal{B}^{(1)} = \{\, (x, y, m) \mid \widehat{B}^{(1)}(x, y, m) = 1 \,\}$, so $H^{(1)'} \subseteq \mathcal{B}^{(1)} \subseteq H^{(1)}$. We have $\delta(\mathcal{B}^{(1)}) \ge \varepsilon$ as before, because $\delta(\mathcal{B}^{(1)}) \ge \delta(H^{(1)'})$, and $\delta(H^{(1)'}) \ge \varepsilon$ by step 4a of FHC_0. It can be seen that the rest of part (ii) of the proof of Theorem 6.3 (defining the hard cores $\mathcal{C}$ and $\mathcal{M}(x, y)$ in terms of $\mathcal{B} = \mathcal{B}^{(1)}$) can be carried out as before.

   (b) FHC_0 stops at step 2. Note that $\delta(H^{(1)}) < \varepsilon$, and

$$\forall (x, y, m) \in G^{(1)}\ \exists U \in \mathcal{U}^{(1)} : \pi^U_T(x, y, m) \ge n^{-a}/2. \tag{15}$$

   Continue to S2.

S2. Run FHC_1 starting with $H_0 = A_0$. (This super-step is symmetric to S1. We are trying to find a hard core inside A_0, where by definition $\pi'_T(x, y, m) \le 1/2 + \nu(n)$, on which simulators are "stuck near 1" and hence are far from $\pi'_T(x, y, m)$).

(a) FHC_1 stops at step 4b. Note that Claim 6.5 holds with part 3(c) changed to: for every probabilistic circuit simulator U of size n^d, there is a set $F \subseteq H^{(2)}$ with $\delta(F|H^{(2)}) \geq 1-n^{-2b}$ such that $\pi_T^U(x,y,m) > 1-n^{-a}$ for all $(x,y,m) \in F$. Define the circuit $\widehat{B}$ from $H^{(2)}$ and $H^{(2)\prime}$ as before, and let $\widehat{B}^{(2)}$ be the conjunction of this circuit with $\widehat{C}_0 = \widehat{C}_1 = 0$ (restricting $\widehat{B}$ to the starting set A_0). Define $\mathcal{B}^{(2)}$ and the hard cores $\mathcal{C}$ and $\mathcal{M}(x,y)$ as in S1.

(b) FHC_1 stops at step 2. Note that $\delta(H^{(2)}) < \varepsilon$, and

$$\forall (x,y,m) \in G^{(2)}\ \exists U \in \mathcal{U}^{(2)} : \pi_T^U(x,y,m) \leq 1-n^{-a}/2. \tag{16}$$

Continue to S3.

S3. Run FHC_0 starting with $H_0 = A_2$. (In this super-step and the next one, we are trying to find a hard core on which each simulator is either "stuck near 0" or "stuck near 1". This time the hard core is a subset of A_2 on which $\pi_T'(x,y,m) = 1/2 \pm \nu(n)$, so again the simulator is far from $\pi_T'(x,y,m)$.)

(a) FHC_0 stops at step 4b. Note that Claim 6.5 holds with part 3(c) changed to: for every probabilistic circuit simulator U of size n^d, there is a set $F \subseteq H^{(3)}$ with $\delta(F|H^{(3)}) \geq 1-n^{-2b}$ such that $\pi_T^U(x,y,m) < n^{-a}$ for all $(x,y,m) \in F$. Define the circuit $\widehat{B}$ from $H^{(3)}$ and $H^{(3)\prime}$ as before, and let $\widehat{B}^{(3)}$ be the conjunction of this circuit with $\widehat{C}_0 = 1$ and $\widehat{C}_1 = 0$ (restricting $\widehat{B}$ to the starting set A_2). Define $\mathcal{B}^{(3)}$ and the hard cores $\mathcal{C}$ and $\mathcal{M}(x,y)$ as in S1.

(b) FHC_0 stops at step 2. Note that $\delta(H^{(3)}) < \varepsilon$, and

$$\forall (x,y,m) \in G^{(3)}\ \exists U \in \mathcal{U}^{(3)} : \pi_T^U(x,y,m) \geq n^{-a}/2. \tag{17}$$

Continue to S4.

S4. Run FHC_1 starting with $H_0 = G^{(3)}$.

(a) FHC_1 stops at step 4b. Note that Claim 6.5 holds with part 3(c) changed to: for every probabilistic circuit simulator U of size n^d, there is a set $F \subseteq H^{(4)}$ with $\delta(F|H^{(4)}) \geq 1-n^{-2b}$ such that $\pi_T^U(x,y,m) > 1-n^{-a}$ for all $(x,y,m) \in F$. Define the circuit $\widehat{B}$ from $H^{(4)}$ and $H^{(4)\prime}$ as before, and let $\widehat{B}^{(4)}$ be the conjunction of this circuit with $\widehat{C}_0 = 1$ and $\widehat{C}_1 = 0$. Define $\mathcal{B}^{(4)}$ and the hard cores $\mathcal{C}$ and $\mathcal{M}(x,y)$ as in S1.

(b) FHC_1 stops at step 2. Note that $\delta(H^{(4)}) < \varepsilon$, and because this super-step started with $H_0 = G^{(3)}$ (see (17))

$$\forall (x,y,m) \in G^{(4)}\ \exists U \in \mathcal{U}^{(3)}\ \exists U' \in \mathcal{U}^{(4)} : \\ \pi_T^U(x,y,m) \geq n^{-a}/2 \quad \text{and} \quad \pi_T^{U'}(x,y,m) \leq 1-n^{-a}/2. \tag{18}$$

As we now show, if a hard core was not found at any of the four super-steps, then there exists a super-simulator that contradicts the definition of E_c'. Let $\mathcal{U} = \bigcup_{1\leq j\leq 4} \mathcal{U}^{(j)}$. By Claim 6.5(1), $\mathcal{U}$ contains at most $4(2b+1)$ simulators, each of size n^d. For a given (x,m), for $z = 0,1$, a conversation ξ is a *z-conversation* if $T(x,m,\xi) = z$. We actually define a class of super-simulators $U^*(\rho)$, where

ρ is an n-bit fraction with $0 \le \rho \le 1$. Roughly speaking, if for input (x, m) the super-simulator $U^*(\rho)$ can find both a 1-conversation and a 0-conversation, then it flips a ρ-biased coin to decide which of these two conversations to output. Eventually, we will choose ρ so that $\pi_T^{U^*(\rho)}(n)$ closely approximates $\pi_T(n)$.

Definition of $U^*(\rho)$: Input (x, m).
For each $U \in \mathcal{U}$, run $U(x, m)$ for $2n^{a+1}$ trials. If only 0-conversations (resp., only 1-conversations) are found, then output an arbitrary one of these 0-conversations (resp., 1-conversations). If both a 0-conversation ξ_0 and a 1-conversation ξ_1 are found, then output ξ_1 with probability ρ, or ξ_0 with probability $1 - \rho$.

It is clear that there is a d', depending only on a, b, d and T, such that the size of $U^*(\rho)$ is at most $n^{d'}$.

Note that $G^{(1)}$ and $H^{(1)}$ partition A_1, $G^{(2)}$ and $H^{(2)}$ partition A_0, and $G^{(4)}$, $H^{(3)}$, and $H^{(4)}$ partition A_2. By Claim 6.12(1), it follows that $G^{(1)}$, $G^{(2)}$, $G^{(4)}$, $H^{(1)}$, $H^{(2)}$, $H^{(3)}$, $H^{(4)}$ partition $\Sigma_n^X \Sigma_n^Y \Sigma_n^M$. As noted in the super-procedure, $\delta(H^{(j)}) < \varepsilon$ for $j = 1, 2, 3, 4$, and a property of $G^{(1)}$, $G^{(2)}$, $G^{(4)}$ is given in (15), (16), (18), respectively. These properties will be used to show the existence of a ρ such that $|\pi_T(n) - \pi_T^{U^*(\rho)}(n)| < E'_c(d', n)$, contradicting the definition of E'_c.

Let $\delta_j = \delta(G^{(j)})$ for $j = 1, 2, 4$. Recall $G^{(1)} \subseteq A_1$, $G^{(2)} \subseteq A_0$, and $G^{(4)} \subseteq A_2$, and recall the fact (noted in the previous paragraph) that $\Sigma_n^X \Sigma_n^Y \Sigma_n^M - (G^{(1)} \cup G^{(2)} \cup G^{(4)})$ is a set of density at most 4ε. It is then immediate from Claim 6.12 that

$$\delta_1(1 - n^{-p}) \le \pi_T(n) \le \delta_1 + \delta_2 n^{-p} + \delta_4 + 4\varepsilon. \tag{19}$$

We will also need bounds on $\pi_T^{U^*(0)}(n)$ and $\pi_T^{U^*(1)}(n)$. By definition of $U^*(0)$, $\pi_T^{U^*(0)}(x, y, m) = 1$ iff $U^*(0)$ does not find a 0-conversation. By (16) and (18), it almost certainly finds a 0-conversation if $(x, y, m) \in G^{(2)} \cup G^{(4)}$. It follows that

$$\pi_T^{U^*(0)}(n) \le \delta_1 + 4\varepsilon + \nu(n). \tag{20}$$

Similarly, $\pi_T^{U^*(1)}(x, y, m) = 1$ iff $U^*(1)$ finds a 1-conversation, which happens almost certainly (by (15) and (18)) if $(x, y, m) \in G^{(1)} \cup G^{(4)}$. Therefore

$$\pi_T^{U^*(1)}(n) \ge \delta_1 + \delta_4 - \nu(n). \tag{21}$$

If $\pi_T^{U^*(0)}(n) \le \pi_T(n) \le \pi_T^{U^*(1)}(n)$, then there is an n-bit fraction ρ with $0 \le \rho \le 1$ such that $\pi_T^{U^*(\rho)}(n) = \pi_T(n) \pm \nu(n)$. So $|\pi_T(n) - \pi_T^{U^*(\rho)}(n)| < \nu(n) < n^{-c} \le E'_c(d', n)$, a contradiction. If $\pi_T(n) < \pi_T^{U^*(0)}(n)$, then by (19) and (20),

$$\delta_1(1 - n^{-p}) \le \pi_T(n) < \pi_T^{U^*(0)}(n) \le \delta_1 + 4\varepsilon + \nu(n).$$

We have chosen p so that $n^{-p} \le E'_c(d', n)/10$, and $4\varepsilon = (4/5)E'_c(d', n)$. So $|\pi_T(n) - \pi_T^{U^*(0)}(n)| \le n^{-p} + 4\varepsilon + \nu(n) < E'_c(d', n)$. Similarly, if $\pi_T(n) > \pi_T^{U^*(1)}(n)$, then by (19) and (21),

$$\delta_1 + \delta_4 - \nu(n) \le \pi_T^{U^*(1)}(n) < \pi_T(n) \le \delta_1 + \delta_2 n^{-p} + \delta_4 + 4\varepsilon.$$

So $|\pi_T(n) - \pi_T^{U^*(1)}(n)| \le n^{-p} + 4\varepsilon + \nu(n) < E'_c(d', n)$. This completes the proof in the case that P' is a machine that is not given (non-uniform) advice.

If P' can receive one bit $v = v(n)$ of advice for each n, it will use the advice bit to tell it whether to look for a 0-conversation or a 1-conversation.

Definition of $P'(v)$:
Receive n^{p+1} samples $t_i = T(x, m, \xi_i)$ where $\xi_i = (P, V^*)(x, y, m)$. If there exists an i with $t_i = v$, then request and use ξ_i. Otherwise, request and use ξ_1.

The "right" advice is obtained as follows. For each n: if FHC_0 stops at its step 4b in super-step S1 or S3 (simulators are stuck near 0), then set $v(n) = 1$; if FHC_1 stops at its step 4b in super-step S2 or S4 (simulators are stuck near 1), then set $v(n) = 0$. $\square$

It should be noted that it is important for the non-triviality of Theorem 6.10 that the size d' of the super-simulator U^* does not depend on c. For if $d' = d'(c)$, then $\varepsilon = E'_c(d'(c), n)$. In the case that for every c, some simulator S of size $n^{d'(c)}$ has $|\pi_T(n) - \pi_T^S(n)| < n^{-c}$, then ε would be identically zero. In the proof, this means that U^* cannot use $\widehat{C}_0$ or $\widehat{C}_1$ because the size of these two circuits grows like n^p, and p grows with c.

7 Selective Decommitment

The selective decommitment problem originated from the *selective decryption* problem in the context of distributed computing, specifically, tolerating a dynamic adversary in Byzantine agreement. Before formally defining commitment and selective decommitment, we briefly describe a version of the selective decryption problem and its roots. Let n be a security parameter. Let E be a public-key cryptosystem enjoying some strong notion of security, such as semantic security under chosen plaintext attack, and let Π be a polynomial-time samplable distribution on m-tuples of messages, where m is polynomial in n. Consider the following interaction between a *sender* B and an *adversary* A (by our default convention, A and B are probabilistic ptime machines):

1. The sender B draws an m-tuple $\bar{p} = (p_1, \ldots, p_m)$ according to Π and creates a vector $\bar{c} = (c_1, \ldots, c_m)$ where each c_i is defined by $c_i = E(p_i, r_i)$; r_i is the (independent) string of random bits used in creating the ciphertext c_i.

2. Given the ciphertext vector $\bar{c}$, the adversary A selects $m/2$ indices $I = \{i_1, \ldots, i_{m/2}\}$ in the set $\{1, 2, \ldots, m\}$.

3. For each $i_j \in I$, B sends to A the pair (p_{i_j}, r_{i_j}).

The problem is to prove (or disprove) that the remaining, unopened, encryptions are "secure".

The connection to distributed computing is as follows. In running a distributed protocol, say, a distributed coin-flipping protocol, in a model in which all processors have a public encryption key, B is one of the processors, E is its public key, and the m messages are the messages sent to B in a given round of the protocol, with plaintext message p_i being encrypted and sent to B by processor i. A *dynamic* adversary is allowed to corrupt parties based on the message traffic; an *intrusive* adversary can see the internal state of any corrupted party. In this setting, the adversary, seeing the ciphertext vector $\bar{c}$, chooses some $m/2$ processors to corrupt. By viewing the internal states of the corrupted processors the adversary can learn the plaintexts and random strings used by these corrupted processors to create the ciphertexts. Intuitively, we want to be sure that the adversary learns no more than the $m/2$ values of the corrupted parties.

Since E is a secure encryption scheme one would have hoped that the remaining ciphertexts are still "secure" under a reasonable definition of security. After all, how can seeing the ciphertexts help

A to "intelligently" choose I? Once I is chosen, how can seeing the encryptions and decryptions of the chosen elements give A any additional information about $\bar{p}$ beyond that obtained from the plaintexts in I themselves? To strengthen this intuition even further we can let the encryption of each plaintext p_i be performed with an independent encryption key pk_i (in addition to an independent random string r_i). Unfortunately, even when each plaintext is encrypted in a truly independent way, proving that the unopened plaintexts are secure turns out to be a real problem: It is theoretically possible for A to compute some unexpected function of the entire plaintext vector $\bar{p}$ or even of a single unopened (unselected) plaintext p_i.

That E is an encryption scheme (*i.e.*, E has a corresponding decryption key) obscures the real issue. The crux of the problem seems to lie in the fact that (usually) $E(p_i, r_i)$ provides a *commitment* to the plaintext value p_i. For standard encryption schemes, B can only open c_i in one way:

$$\forall (p', r') \neq (p_i, r_i) : \; E(p', r') \neq E(p_i, r_i)\,.$$

Indeed, [8] defined and constructed a *non-committing* encryption scheme, and used it to bypass the problem of selective decryption in order to construct distributed protocols that are secure against a dynamic adversary. The general question of whether even encryption schemes that are non-malleable against chosen ciphertext attacks in the post-processing mode (NM-CCA-post) are secure against selective decryption is still open. Moreover, the extreme complexity of the non-committing encryption of [8](no other is known), in comparison with the simplicity of the NM-CCA-post secure Cramer-Shoup cryptosystem [13], make this question of great practical interest.

Remark 7.1 *Another interesting scenario is when A can choose the subset I* adaptively *(that is, for each j, choose i_j, see the pair (p_{i_j}, r_{i_j}) and only then choose i_{j+1}). However, this scenario seems much more difficult than the (sufficiently difficult) "one-shot" case. Moreover, this scenario has no (apparent) application to 3-round interactive proofs (although it does have application to distributed computing).*

We now formally describe what we mean by a commitment scheme. In this paper, we focus on schemes with single-round commit phase and single-round reveal phase. For more general (*i.e.*, interactive) definitions of commitment schemes see *e.g.* [35, 36, 14]. A function $\mathcal{C}$ that takes as input a pair p, r (p is the plaintext and r is the random string) and a security parameter 1^ℓ, is a commitment scheme if it has the following properties:

1. *Efficiently Computable.* $\mathcal{C}(\cdot, \cdot, 1^\ell)$ runs in time polynomial in ℓ.

2. *Binding.* No computationally bounded algorithm can find two pairs (p, r) and (p', r'), where $p \neq p'$ and $\mathcal{C}(p, r, 1^\ell) = \mathcal{C}(p', r', 1^\ell)$. That is, for all constants c, for all sufficiently large values of ℓ, for all probabilistic polynomial time algorithms $\mathcal{A}$ running in time at most ℓ^c, the probability that $\mathcal{A}$ can find (p, r) and (p', r'), where $p \neq p'$ and $\mathcal{C}(p, r, 1^\ell) = \mathcal{C}(p', r', 1^\ell)$, is less than ℓ^{-c}, where the probability is taken over r, r', and the random choices of made by $\mathcal{A}$. In the *strong-sender* variant, it is not only hard to come up with a "collision" (p, r) and (p', r'), but such collisions *do not exist.*

3. *(Semantic) Security.* For any p, p' the distributions $\mathcal{C}(p, r, 1^\ell)$ and $\mathcal{C}(p', r, 1^\ell)$ (where r is uniformly distributed) are computationally indistinguishable. That is, for all constants c, for all sufficiently large ℓ, and for all probabilistic polynomial time tests $\mathcal{A}$,

$$|\Pr[\mathcal{A}(\mathcal{C}(p, r, 1^\ell))] - \Pr[\mathcal{A}(\mathcal{C}(p', r', 1^\ell))]| < \ell^{-c}$$

where the probabilities are over the choices of r, r' and the random choices of $\mathcal{A}$. In the *strong-receiver* variant (also known as information-theoretic commitment), the commitments leak no information about the plaintext values information-theoretically: the distributions $\mathcal{C}(p, r, 1^\ell)$ and $\mathcal{C}(p', r', 1^\ell)$ are identical, or at least statistically close.

Selective decommitment is almost identical to selective decryption, with the difference that the encryption scheme E is replaced with a commitment scheme $\mathcal{C}$. We remark that, prior to the current work, no solution to the selective decommitment problem was known[8] even for the case of *strong-receiver* commitments. Even when each commitment yields no information about the plaintexts information-theoretically, security against selective decommitment seems difficult to prove (and, to our knowledge, may not even hold). In Section 7.4 we discuss the limitations of standard definitions and proof techniques in showing security for the selective decommitment problem. We also give some preliminary positive results.

The rest of the section is organized as follows: In Section 7.1, we give a natural definition of security for selective decommitment. Under this definition, we prove in Section 7.2 that if there exists a commitment scheme secure against selective decommitment, then every language in NP has a public-coin 3-round argument (proof where the prover is polynomial-time bounded) that satisfies Definition 4.9. It follows from Theorem 5.4 (and the fact that Definition 5.1 is a weakening of Definition 4.9) that for such proofs the Fiat-Shamir methodology fails completely. In Section 7.3, we give a weaker definition of (semantic) security for selective decommitment and show its relation to the existence of some sort of semantically secure 3-round arguments. Finally, we give our positive results for selective decommitment in Section 7.4.

7.1 Definition of Security

We now formally define the *selective decommitment* problem for a commitment scheme $\mathcal{C}$. As described above (for selective decryption), this problem concerns the following interactive protocol (B, A) between a *sender* B and an *adversary* A. The private input of the sender B is an m-tuple of plaintexts $\bar{p} = (p_1, \ldots, p_m)$. We also allow A to have a private input z that represents its auxiliary information. The quantities $\bar{p}, z$ are distributed according to $D_n = \Pi_n Z_n$. The protocol has three rounds:

1. B creates a sequence of independent commitments $\bar{c} = (c_1, \ldots, c_m)$ where each c_i is defined by $c_i = \mathcal{C}(p_i, r_i)$; r_i is the (independent) string of random bits used in creating the commitment c_i. B then sends $\bar{c}$ to A.

2. Given the commitments vector $\bar{c}$, the adversary A selects *a legal subset*[9] of the commitments: $I = \{i_1, \ldots, i_k\} \subseteq \{1, 2, \ldots, m\}$. A then sends I to B.

3. For each $i_j \in I$, B sends to A the pair (p_{i_j}, r_{i_j}).

The intuition for the next definition is that for every adversary selector A trying to learn something about the plaintext p, there should exist another machine M that does as well as A, *without seeing the commitments*. Thus, without seeing the commitments, M selects a legal subset

[8]Several researchers have been concerned with problems of this nature during the last decade (e.g., [7, 8, 9, 20, 37]).

[9]In the description above a legal subset is any subset of size $m/2$. However, each application may call for its own definition of a "legal subset". We therefore let this definition be implicit in our description.

I' of values, and receives as a response the corresponding plaintexts. To formulate this intuition we consider the following interactive protocol $(B', M)(\bar{p}, z)$ between a *modified sender* B' and the machine M. The inputs $\bar{p}, z$ are distributed according to $D_n = \Pi_n Z_n$ as above. The protocol now has only two rounds:

1. M selects a legal subset of plaintexts: $I' = \{i'_1, \ldots, i'_k\} \subseteq \{1, 2, \ldots, m\}$ and sends I' to B'.
2. For each $i'_j \in I'$, B' sends to M the plaintext $p_{i'_j}$.

It remains to define what it means that A or M "learns something" about the plaintext p. Dolev, Dwork, and Naor [14] introduced the concept of *semantic security with respect to relations* for the case of encryption, and showed that, under several types of attack, semantic security with respect to relations is equivalent to "traditional" semantic security (as defined by Goldwasser and Micali [24]; see Section 7.1.1 below). Intuitively, a cryptosystem enjoys semantic security with respect to relations if for all probabilistic polynomial-time computable relations R, seeing $E(p)$ "does not help" in generating p' such that $R(p, p')$ is satisfied[10]. Our definition of security for selective decommitment, presented next, is analogous to the notion in [14][11].

Definition 7.2 (Semantic Security wrt Relations for Selective Decommitment) $\forall R \forall A_1 A_2 \forall D = \{\Pi_n Z_n\} \exists M_1 M_2$

$$|\Pr[R(\Pi_n, Z_n, A_2((B, A_1)(\Pi_n, Z_n), Z_n))] - \Pr[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n), Z_n))]| < \nu(n).$$

Remark 7.3 *In Definition 7.2 we have made the following choices:*

- *The adversary A is split into two parts: A_1 that interacts with B, and A_2 that uses the transcript of this interaction (including the final state of A_1 and all of its tape contents) and tries to satisfy the relation R. The machine M is split in a similar way. This choice is merely a matter of presentation.*
- *One of the inputs to the relation R is the auxiliary input z (that is accessible to both A and M).*

We keep consistency with these choices in subsequent definitions.

7.1.1 Informal Review of Notions of Security

In this section, we briefly review several notions of security and the connections among them. We focus on the uniform case.

Goldwasser and Micali's definition of semantic security for cryptosystems says essentially that for all *not necessarily efficiently computable* functions F, seeing $E(p)$ "does not help" in computing $F(p)$ [24]. This definition is known to be equivalent to *indistinguishability of encryptions*,

[10] The intuition for restricting R to be polynomial-time computable is simple: Suppose $R(p, y \circ E)$ holds iff y is a double-encryption of p under the public probabilistic encryption algorithm E. Then, given $y' \in_R E(p)$ we simply choose $y \in_R E(y')$ and output y. Clearly, $R(p, y \circ E)$ is satisfied; equally clearly, without access to $E(p)$ it is impossible to do so well at generating a double-encryption of p.

[11] In the definition of [14], the adversaries A and M may choose different distributions on which to be tested; in the zero-knowledge setting, the simulator works with the same distribution on theorem instances as in the real conversations. Because we are interested in applying selective decommitment to zero-knowledge, the definition tests both adversaries on the same distribution.

which says, roughly, that it is hard to find a pair of plaintexts p, p' such that encryptions of p are polynomial-time distinguishable from encryptions of p' [34, 19]. As it turns out, the difficulty of computing the function F has no role in the proof – thus, indistinguishability is equivalent to semantic security with respect to polynomial-time computable functions. Indistinguishability of encryptions is also known to be equivalent to semantic security with respect to *probabilistic polynomial-time* computable *relations* [14]; equivalence to semantic security with respect to ptime functions follows via the previous result.

No such equivalences are known for selective decommitment. As we will see, in this case semantic security with respect to relations is actually equivalent to *simulateability* of interactions, yielding the connection between selective decommitment and interactive proof systems (see Section 7.2). Section 7.3 contains a definition of semantic security for selective decommitment with respect to *polynomial-time* computable functions. We show that in some scenarios we can actually achieve semantic security with respect to polynomial-time computable functions; however, the polynomial-time restriction cannot be relaxed in our proofs. Indeed, we conjecture that, *in the case of selective decommitment*, semantic security with respect to polynomial-time computable relations is stronger than semantic security with respect to polynomial-time computable functions. Resolution of this conjecture would be extremely interesting, especially in light of the connection between security of selective decommitment and security of certain natural 3-round public-coin interactive proofs (see Section 7.2.2 below).

7.2 Connection to $S(V, T, D)$ Zero-Knowledge

In this section we show that if there exists a commitment scheme that satisfies Definition 7.2 (semantic security with respect to relations for selective decommitment), then every language in NP has a 3-round argument that is $S(V, T, D)$ Zero-Knowledge (Definition 4.9). To simplify this proof we first introduce (in Section 7.2.1) a new definition of security for arguments and prove its equivalence to Definition 4.9.

7.2.1 Another Characterization of $S(V, T, D)$ Zero-Knowledge

Coming to prove that commitment schemes that satisfy Definition 7.2 imply $S(V, T, D)$ zero-knowledge arguments, we seem to face the following problem: While Definition 7.2 is phrased in terms of semantic security, the different definitions of zero-knowledge are all based on the (seemingly more demanding) notion of simulatability. As it turns out, this is not really a problem: In both settings (selective decommitment and interactive protocols), semantic security with respect to relations is just another form of simulatability. We formalize this claim by introducing a definition of semantic security with respect to relations for interactive protocols and showing its equivalence to Definition 4.9 of $S(V, T, D)$ zero-knowledge. We note that in the same way one can give a definition of security for selective decommitment based on simulatability and show its equivalence to Definition 7.2.

Definition 7.4 (Semantic Security wrt Relations for IPs (over W)) *An interactive protocol (P, V) enjoys semantic security with respect to relations (over W) if $\forall R \, \forall V_1^* V_2^* \, \forall D = \{X_n Y_n Z_n\}$ (with $X_n Y_n$ ranging over W) $\exists M$*

$$|\Pr[R(X_n, Y_n, Z_n, V_2^*((P, V_1^*)(X_n, Y_n, Z_n), X_n, Z_n))] - \Pr[R(X_n, Y_n, Z_n, M(X_n, Z_n))]| < \nu(n).$$

Note that the expression "with respect to relations" in the above definition refers to the relation R, and not to the relation W.

Theorem 7.5 *An interactive protocol (P, V) satisfies Definition 4.9 (over W) iff (P, V) satisfies Definition 7.4 (over W).*

Theorem 7.5 is obtained as an immediate corollary of Lemmas 7.6 and 7.7 proved below.

The connection between simulatability and semantic security with respect to relations is rather immediate; the basic idea is that the test T is identified with the relation R, and the simulator S is identified with the machine M. Nevertheless, obtaining an equivalence between these notions requires a careful choice of the exact variant of simulatability. Two important properties of Definition 4.9 that are vital to the proof of Theorem 7.5 are:

1. The simulator S "knows" the test T (that the simulated conversation has to pass). This is used for showing that Definition 7.4 implies Definition 4.9.

2. The test T has access to the witness y (and not only to the public input x and auxiliary information z). This is used for showing the converse.

3. The simulator S "knows" the distribution D. This is needed because, in Definition 7.4, the distribution D comes before the simulator M in the quantification order.

Lemma 7.6 *If (P, V) satisfies Definition 7.4 (over W), then (P, V) satisfies Definition 4.9 (over W).*

Proof. Suppose the lemma false. Then there exists an interactive proof system (P, V) over W satisfying Definition 7.4 but not satisfying Definition 4.9, so in particular we have (by negating Definition 4.9): $\exists V^* \, \exists T \, \exists D = \{X_n Y_n Z_n\}$ (with $X_n Y_n$ ranging over W) $\forall S \, \exists c$ for infinitely many n

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| \geq \frac{1}{n^c}.$$

The test T immediately defines a relation that *should be hard* to satisfy: ξ is related to x, y, z under T iff it is a transcript that T accepts when the inputs to the protocol are x, y and z. More precisely, define $R \equiv T$, $V_1^* \equiv V^*$ and $V_2^*(\xi, x, z) = \xi$. From Definition 7.4 it follows that there exists a probabilistic polynomial-time machine M such that

$$|\Pr[R(X_n, Y_n, Z_n, V_2^*(((P, V_1^*)(X_n, Y_n, Z_n), X_n, Z_n))] - \Pr[R(X_n, Y_n, Z_n, M(X_n, Z_n))]| < \nu(n).$$

Defining $S \equiv M$, this is equivalent to:

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

This contradicts the assumption that (P, V) does not satisfy Definition 4.9. $\square$

Lemma 7.7 *If (P, V) satisfies Definition 4.9 (over W), then (P, V) satisfies Definition 7.4 (over W).*

Proof. Suppose the lemma false. Then there exists an interactive proof system (P, V) satisfying Definition 4.9 but not satisfying Definition 7.4, so in particular we have (by negating Definition 7.4): $\exists R\,\exists V_1^* V_2^*\,\exists D = \{X_n Y_n Z_n\}$ (with $X_n Y_n$ ranging over W) $\forall M\,\exists c$ for infinitely many n

$$|\Pr[R(X_n, Y_n, Z_n, V_2^*((P, V_1^*)(X_n, Y_n, Z_n), X_n, Z_n))] - \Pr[R(X_n, Y_n, Z_n, M(X_n, Z_n))]| \geq \frac{1}{n^c}.$$

We can now easily define a test T that should distinguish simulated conversations between P and V_1^* from real ones. The test simply checks whether V_2^* is able to satisfy the relation R when it gets this conversation as input. More precisely, define $V^* \equiv V_1^*$ and $T(x, y, z, \xi) = R(x, y, z, V_2^*(\xi, x, z))$. Then by Definition 4.9, there exists a probabilistic polynomial-time machine S such that

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < \nu(n).$$

Defining $M(x, z) = V_2^*(S(x, z), x, z)$, this is equivalent to:

$$|\Pr[R(X_n, Y_n, Z_n, V_2^*((P, V_1^*)(X_n, Y_n, Z_n), X_n, Z_n))] - \Pr[R(X_n, Y_n, Z_n, M(X_n, Z_n))]| < \nu(n).$$

This contradicts the assumption that (P, V) does not satisfy Definition 7.4. $\square$

7.2.2 Selective Decommitment Implies $S(V, T, D)$ Zero-Knowledge

Let $\mathcal{C}$ be a commitment scheme that is secure for selective decommitment (under Definition 7.2). We now show how to use $\mathcal{C}$ in order to construct a 3-round $S(V, T, D)$ zero-knowledge argument for all languages in NP. To this end, it is enough to construct a 3-round $S(V, T, D)$ zero-knowledge argument for a single NP-complete problem. In this section we show that the parallel execution of the 3-round argument of Goldreich, Micali and Wigderson [22] for 3-coloring using $\mathcal{C}$ for the commitments of the first round is $S(V, T, D)$ zero-knowledge.

The input of the 3-coloring problem $x = (X, E)$ is a graph with n vertices. Here, n will be the security parameter both for the interactive proof system and for the commitment scheme. For simplicity, we suppress the third parameter, 1^n, in the notation for commitment. That is, in the remainder of this section, we write $C(p, r)$ instead of $C(p, r, 1^n)$. The graph x is 3-colorable iff it has a legal 3-coloring of its vertices, that is, a coloring $y : X \mapsto \{1, 2, 3\}$ such that the endpoints of each edge are colored with different colors $(\forall (u, v) \in E, y(u) \neq y(v))$. Clearly, y is a witness that x is 3-colorable. Denote by W_{3C} the corresponding relation.

In essence, the Goldreich, Micali, and Wigderson protocol (henceforth, the Ur protocol) for 3-coloring goes as follows. (See [22] for exact definitions and analysis. The superscript "b" on P and V stands for "basic block"; a more sound protocol for the 3-coloring problem will run many independent copies of this basic block.)

1. The prover P^{b} chooses a random permutation π of the color set $\{1, 2, 3\}$. Then it creates a sequence of commitments $\bar{c} = (c_1, \ldots, c_n)$ where c_i is a commitment to the permuted color of i. That is, $c_i = \mathcal{C}(p_i, r_i)$ where $p_i = \pi(y(i))$ and r_i is an independent string of random bits. P^{b} sends the string of commitments $\bar{c}$ to the verifier V^{b}.

2. V^{b} chooses an edge $e = (u, v) \in E$ uniformly at random and sends it to P^{b}.

3. P^{b} checks that (u, v) is indeed in E and then sends to V^{b} the pairs (p_u, r_u) and (p_v, r_v) (the opening of c_u and c_v). V^{b} accepts this interaction if p_u and p_v are in the color set $\{1, 2, 3\}$, $p_u \neq p_v$ and the messages of the first and third rounds are consistent (i.e. $\mathcal{C}(p_u, r_u) = c_u$ and $\mathcal{C}(p_v, r_v) = c_v$).

As shown in [22] this is a zero-knowledge argument (or proof if $\mathcal{C}$ is strong-sender) for 3-coloring with the reservation that soundness is achieved in a very weak sense. The probability that a cheating prover fools V^{b} is only polynomially bounded from 1. Assume for example that the prover commits to a 3-coloring of the graph that violates exactly one edge. In such a case, the only way V^{b} will catch the deceit is by choosing the violated edge (which happens with probability $1/|E|$). A natural way of improving the soundness of this system without increasing the number of rounds is to run this protocol polynomially many times independently *in parallel*.[12] Unfortunately, it is not clear if the resulting protocol is still zero-knowledge. This is a very important open problem. Theorem 7.8 reduces this problem (for the weaker notion of $S(V, T, D)$ zero-knowledge) to the security of the commitment scheme $\mathcal{C}$ in the setting of selective decommitment.

The Ur protocol is in fact a proof of knowledge [16] of a witness y to the 3-colorability of x. The intuition that leads to the following theorem is that it suffices to show that nothing about y is leaked by the interaction.

Theorem 7.8 *Let $\mathcal{C}$ be a commitment scheme satisfying Definition 7.2 (semantic security with respect to relations for selective decommitment). Let (P, V) be the parallel execution of the 3-round argument of [22] for 3-coloring using $\mathcal{C}$ for the commitments in the first round. Then (P, V) satisfies Definition 7.4 (semantic security with respect to relations for IPs).*

By Lemma 7.6, it follows that (P, V) satisfies Definition 4.9. To draw a connection between Definitions 7.2 and 7.4, we view the sender B as the prover P, and the adversary A as the "cheating verifier" V^*.

Let us first highlight a few properties of the protocol (P, V) above that are helpful in proving Theorem 7.8 (indeed, Theorem 7.8 holds for any (P, V) satisfying these conditions, where in addition the "legal subsets" for input x are recognizable in polynomial time, given x):

1. The structure of the protocol bears a strong resemblance to the setting of selective decommitment: In the first round P sends to V a polynomial sequence of commitments. V asks P to open a legal subset of these commitments (here a "legal subset" contains two vertices that corresponds to an edge in each block of commitments that corresponds to a single execution of the Ur-protocol $(P^{\mathrm{b}}, V^{\mathrm{b}})$) and it receives these values in the third round.

2. For any legal subset of commitments that the good verifier V may ask to open, the distribution of the corresponding plaintexts is independent of the witness and is easy to sample (for any edge, the corresponding plaintexts are just two different random colors in $\{1, 2, 3\}$; recall here that the good verifier requests positions independent of the ciphertexts).

3. We can assume wlog that the sequence of plaintexts P commits to in the first round completely defines the witness y (this is not really necessary but it simplifies the proof). One way to

[12] Running this 3-round argument in parallel rapidly improves its soundness (the error essentially decreases exponentially fast in the number of executions). This can be deduced from [6] and can also be derived directly using simpler combinatorial arguments.

achieve this is to assume that P^{b} also sends a commitment to π. (An alternate proof, not using this assumption, can be based on the facts that: (i) the sequence $\bar{p}$ of plaintexts defines the witness up to a permutation of the colors; and (ii) P behaves exactly the same on all six of these witnesses.)

Proof. Assume for the sake of contradiction that Definition 7.4 is not satisfied. Then $\exists R' \,\exists V_1^* V_2^* \,\exists D' = \{X_n' Y_n' Z_n'\}$ (with $X_n' Y_n'$ ranging over W_{3C}) $\forall M' \,\exists c$ for infinitely many n

$$|\Pr_{D_n'}[R'(X_n', Y_n', Z_n', V_2^*((P, V_1^*)(X_n', Y_n', Z_n'), X_n', Z_n'))] - \Pr_{D_n'}[R'(X_n', Y_n', Z_n', M'(X_n', Z_n'))]| \geq \frac{1}{n^c}.$$

(where we explicitly note the distribution D_n' to remind the reader of the distribution on the inputs, remembering that the probabilities are also over all coin flips of all machines involved). We will now derive a contradiction to the assumption that $\mathcal{C}$ satisfies Definition 7.2.

Define $A_1(x' \circ z') = V_1^*(x', z')$, and let $A_2(\xi, x' \circ z') = V_2^*(\xi, x', z')$. For any sequence of plaintexts, $\bar{p}$, that the legal prover P may commit to in the first step of the interactive proof system, let $y'(\bar{p})$ be the witness that is derivable from $\bar{p}$. For any legal sequence of plaintexts $\bar{p}$, let $R(\bar{p}, x' \circ z', \theta) = R'(x', y'(\bar{p}), z', \theta)$. Let $D_n = \Pi_n Z_n$, where Π_n and Z_n are defined as follows. First, choose $(x', y', z') \in_R D_n'$. Set $z = x' \circ z'$ and sample Π_n according to the distribution of plaintexts that P commits to in the first step the interactive proof system when x' is the joint input and y' is the prover's auxiliary input (*i.e.*, the witness). D_n is clearly samplable in polynomial time.

By Definition 7.2 there exist machines M_1, M_2 such that

$$|\Pr_{D_n}[R(\Pi_n, Z_n, A_2((B, A_1)(\Pi_n, Z_n), Z_n))] - \Pr_{D_n}[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n), Z_n))]| < \nu(n).$$

We define the machine M' as follows: On input (x', z'), M' first simulates a conversation between B' and M_1: it invokes M_1 on input $z = x' \circ z'$ to choose a subset I' of plaintexts. If I' is not a legal subset or is syntactically incorrect, then the simulated transcript consists entirely of I'. If I' is a legal subset, M' gives M_1 a sequence of plaintexts that are distributed according to the restriction of Π_n to I' (here we use the property 2 above that this is easy to sample even without knowing y). Let ξ be the transcript of the simulated conversation between B' and M_1. M' outputs $M_2(\xi, z)$. By the definition of M' and D_n we have that

$$M_2((B', M_1)(\Pi_n, Z_n), Z_n) = M'(X_n', Z_n').$$

By the definition of A_1 A_2 and D_n we also have that

$$A_2((B, A_1)(\Pi_n, Z_n), Z_n) = V_2^*((P, V_1^*)(X_n', Y_n', Z_n'), X_n', Z_n').$$

By the definition of R, we can therefore conclude that

$$|\Pr_{D_n'}[R'(X_n', Y_n', Z_n', V_2^*((P, V_1^*)(X_n', Y_n', Z_n'), X_n', Z_n'))] - \Pr_{D_n'}[R'(X_n', Y_n', Z_n', M'(X_n', Z_n'))]| < \nu(n)$$

contradicting the assumption that Definition 7.4 is not satisfied. $\square$

7.3 An Alternate Flavor of Security

All the definitions of zero-knowledge and security for selective decommitment that we have presented so far are *simulation-based* (or provably equivalent to simulation-based definitions). Here we discuss a relaxation closer to "ordinary" semantic security (*i.e.*, semantic security wrt functions) both for interactive proof systems and for selective decommitment. These relaxations seem easier to achieve: we present some positive results for this type of security in Section 7.4. Recall that Theorem 7.8 established a connection between security (semantic security with respect to relations) for selective decommitment and $S(V,T,D)$ zero-knowledge. We have an analogue for the case of semantic security with respect to polynomial-time computable functions: security of this type for selective decommitment implies the same type of security for 3-round IPs.

In ordinary semantic security the adversary tries to compute some fixed function of the plaintexts (compared with semantic security wrt relations where the adversary tries to come up with a string that is related to the plaintexts in some way). We note that the notions of semantic security in Definitions 7.9 and 7.10 are somewhat weaker than that of [24]. In our case, the function F that the adversary tries to compute must be *efficiently computable* (compared with an arbitrary function in [24]). We emphasize this point in the definitions by explicitly writing "$\forall$ ptime F". We take F to be ptime computable in Definitions 7.9 and 7.10 because we want semantic security wrt functions to be a weakening of semantic security wrt (ptime computable) relations (Definitions 7.2 and 7.4). In the proof of Theorem 7.14, we use that F is ptime. However, it is easy to see that Theorem 7.11 still holds if "*ptime*" is deleted from both Definition 7.9 and Definition 7.10.

Definition 7.9 (Semantic Security wrt Functions for Selective Decommitment) $\forall$ *ptime* $F\,\forall A_1A_2\,\forall D=\{\Pi_nZ_n\}\,\exists M_1M_2$

$$|\Pr[A_2((B,A_1)(\Pi_n,Z_n),Z_n)=F(\Pi_n,Z_n)]-\Pr[M_2((B',M_1)(\Pi_n,Z_n),Z_n)=F(\Pi_n,Z_n)]\,|<\nu(n).$$

Definition 7.10 (Semantic Security wrt Functions for IPs (over W)) *An interactive protocol (P,V) enjoys semantic security with respect to functions over W if* $\forall$ *ptime* $F\,\forall V_1^*V_2^*\,\forall D=\{X_nY_nZ_n\}$ *(with X_nY_n ranging over W)* $\exists M$

$$|\Pr[V_2^*((P,V_1^*)(X_n,Y_n,Z_n),X_n,Z_n)=F(X_n,Y_n,Z_n)]-\Pr[M(X_n,Z_n)=F(X_n,Y_n,Z_n)]\,|<\nu(n).$$

In Section 7.2.2 it was shown that commitment schemes that are semantically secure *wrt relations* imply 3-round arguments for all NP that are semantically secure *wrt relations* (*i.e.*, $S(V,T,D)$ zero-knowledge). In the same way, the existence of commitment schemes that are semantically secure *wrt functions* imply 3-round arguments for all NP that are semantically secure *wrt functions*.

Theorem 7.11 *Let $\mathcal{C}$ be a commitment scheme satisfying Definition 7.9. Let (P,V) be the parallel execution of the 3-round argument of [22] for 3-coloring using $\mathcal{C}$ for the commitments in the first round. Then (P,V) satisfies Definition 7.10.*

The proof of Theorem 7.11 is almost identical to the proof of Theorem 7.8. We therefore only sketch the differences:

Proof. Assume for the sake of contradiction that Definition 7.10 is not satisfied. Then $\exists$ ptime $F'\,\exists V_1^*V_2^*\,\exists D_n'=X_n'Y_n'Z_n'$ (with $X_n'Y_n'$ ranging over W_{3C}) $\forall M'\,\exists c$ for infinitely many n

$$|\Pr_{D_n'}[V_2^*((P,V_1^*)(X_n',Y_n',Z_n'),X_n',Z_n')=F'(X_n',Y_n',Z_n')]-\Pr_{D_n'}[M'(X_n',Z_n')=F'(X_n',Y_n',Z_n')]\,|\geq\frac{1}{n^c}.$$

We will now derive a contradiction to the assumption that $\mathcal{C}$ satisfies Definition 7.9.

Define $A_1(x' \circ z') = V_1^*(x', z')$, and let $A_2(\xi, x' \circ z') = V_2^*(\xi, x', z')$. For any legal sequence of plaintexts $\bar{p}$, let $y'(\bar{p})$ be the witness that is derivable from $\bar{p}$ and define $F(\bar{p}, x' \circ z', \xi) = F'(x', y'(\bar{p}), z', \xi)$. Let $D_n = \Pi_n Z_n$ be defined as in the proof of Theorem 7.8.

By Definition 7.2 there exist machines M_1, M_2 such that

$$|\Pr_{D_n}[A_2((B, A_1)(\Pi_n, Z_n), Z_n) = F(\Pi_n, Z_n)] - \Pr_{D_n}[M_2((B', M_1)(\Pi_n, Z_n), Z_n) = F(\Pi_n, Z_n)]| < \nu(n).$$

Define M' as in the proof of Theorem 7.8. Then as in this proof we have that

$$M_2((B', M_1)(\Pi_n, Z_n), Z_n) = M'(X'_n, Z'_n)$$

and also have that

$$A_2((B, A_1)(\Pi_n, Z_n), Z_n) = V_2^*((P, V_1^*)(X'_n, Y'_n, Z'_n), X'_n, Z'_n).$$

By the definition of F, we can therefore conclude that

$$|\Pr_{D'_n}[V_2^*((P, V_1^*)(X'_n, Y'_n, Z'_n), X'_n, Z'_n) = F'(X'_n, Y'_n, Z'_n)] - \Pr_{D'_n}[M'(X'_n, Z'_n) = F'(X'_n, Y'_n, Z'_n)]| < \nu(n)$$

contradicting the assumption that Definition 7.10 is not satisfied. □

7.4 On Solving Selective Decommitment

In this section we present several positive results for the selective decommitment problem. We start by considering results that apply to any "secure" commitment scheme (e.g., strong-sender or strong-receiver); our more interesting general result is that, in the special case that the plaintext distribution is a cross-product distribution, semantic security with respect to polynomial-time functions is achievable.

We then show two positive results that are unique to strong-receiver commitment schemes that have a trapdoor. Both these results are for mild weakenings of semantic security with respect to relations.

Finally, we examine the case in which commitment is peformed using a random oracle. We show that the commitment scheme $\mathcal{C}(x, r) \stackrel{\text{def}}{=} H(x, r)$, where H is a random oracle, achieves weakened versions of semantic security with respect to relations and semantic security with respect to polynomial-time computable functions.

The results of this section are interesting as a preliminary effort to solve the selective decommitment problem. However, at least as interesting is what we did not manage to prove. In other words, our results demonstrate the limitation of standard definitions and proof techniques in coping with the selective decommitment problem. An amazing example is the case of strong-receiver commitment scheme (recall that strong-receiver commitments do not leak any information even information-theoretically): Intuitively, one might assume that such commitments solve the selective decommitment problem. Unfortunately, we do not see a way to prove such a general result for natural definitions of security as Definition 7.2 or even Definition 7.9. That is, as far as we are able to prove, even such a scheme may leak information about the plaintexts in the setting of selective decommitment. Furthermore, even the seemingly "perfect" commitment scheme that exists in the random oracle model may not satisfy a very natural security requirement for selective decommitment. To gain some insight on the source of this counter-intuitive state of affairs we start by examining a couple of special cases of the selective decommitment problem where any commitment scheme is secure. Eliminating these cases helps us focus on the heart of the problem.

7.4.1 General Results

Let $\mathcal{C}$ be any commitment scheme (either strong-sender or strong-receiver or even one which is both (only) computationally binding and (only) computationally secure). What kind of security (if any) does it achieve for the selective decommitment problem?

Recall the setting of selective decommitment (formally defined in Section 7.1): An adversary A receives a sequence $\bar{c} = (c_1, \ldots, c_m)$ of independent commitments to a sequence of plaintexts $\bar{p} = (p_1, \ldots, p_m)$ that is distributed according to Π_n. A then selects *a legal subset* of the commitments: $I = \{i_1, \ldots, i_k\} \subseteq \{1, 2, \ldots, m\}$ and receives the opening of these commitments (*i.e.*, for each $i_j \in I$, A receives the pair (p_{i_j}, r_{i_j}), where $c_i = \mathcal{C}(p_{i_j}, r_{i_j})$). Upon receiving these values, A tries to "learn something" about the plaintext vector $\bar{p}$. The scheme is secure if there is another machine M who "does as well as A", when M only receives the plaintexts in a legal subset I' of its choice (*i.e.*, M does not receive the vector of commitments $\bar{c}$).

We now show two special cases where $\mathcal{C}$ yields some sort of security for selective decommitment:

1. When there are only polynomially many "legal subsets". In this case M can simulate A and therefore we obtain some sort of semantic security *with respect to relations*.

2. When each plaintext p_i is chosen from an independent distribution. In this case we obtain some sort of semantic security *with respect to functions*.

The second result is much more interesting and surprising than the first one.

Few legal subsets can be simulated. As discussed in Section 7.2.1, in the context of selective decommitment, *semantic security with respect to relations is equivalent to simulatability*. Therefore, for $\mathcal{C}$ to achieve Definition 7.2 of security for selective decommitment, the machine M (interacting with B') should be able to simulate the adversary A (interacting with B). The naïve way M can simulate (B, A) is as follows:

1. Sample a subset I' from a distribution that is indistinguishable from the distribution of the subset I that A selects when interacting with B (one way M can do that is by invoking A with a sequence of commitments to "garbage").

2. Ask B' for the plaintexts p_i in the subset I'.

3. Create a sequence of commitments $\bar{c}$ where for every $i \in I'$, c_i is a commitment to p_i (all other c_i's can be commitments to "garbage"). Invoke A with the sequence $\bar{c}$ and "hope" that it will select the subset $I = I'$. Otherwise repeat *this step* all over again.

The problem with this method is that the expected number of times the third step repeats itself is roughly ℓ, where ℓ is the number of legal subsets (that both A and M are allowed to ask for). On the other hand, it is not clear what alternative method M can use (assuming that the subset I that A selects is in some sense a "magic" function of the commitments). In general, ℓ can be exponential in n. Therefore, achieving Definition 7.2 in the general case seems very difficult. Nevertheless, if we can allow M to spend time which is linear in ℓ then selective decommitment is easy to handle. Formalizing the simulation method described above and applying standard arguments we deduce the following theorem:

Theorem 7.12 *Let $\mathcal{C}$ be any secure commitment scheme. Let $\ell = \ell(n)$ be the number of legal subsets. Then, $\forall A = \langle A_1, A_2 \rangle\, \exists M = \langle M_1, M_2 \rangle$ that runs in expected polynomial time $\forall R \forall D = \{\Pi_n Z_n\}$, letting n be the security parameter of $\mathcal{C}$,*

$$|\Pr[R(\Pi_n, Z_n, A_2((B, A_1)(\Pi_n, Z_n), Z_n))] - \Pr[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n, 1^\ell), Z_n))]| < \nu(n).$$

Remark 7.13 *There are a few differences between the statement of the theorem above and Definition 7.2:*

1. *The input of M_1 includes 1^ℓ. This allows the running time of M_1 to be polynomial in ℓ (in fact, linear in ℓ is sufficient).*
2. *The same M works for all R and all D. In fact, M only needs black-box access to A.*
3. *M runs in* expected *polynomial time (rather than strictly polynomial time). The reason for this deficiency is the "one-time" nature of the simulation: M is only allowed to ask for the plaintexts in a single subset I'. Consider another setting (such as simulation of the Ur protocol) where M can ask for the plaintexts in a subset I' and either continue with the simulation or start again with a new sequence of plaintexts chosen according to Π_n. In such a case M has a simpler and better simulation strategy:*
   - *Choose I' uniformly at random from all legal subsets.*
   - *Ask B' for the plaintexts p_i in the subset I'.*
   - *Create a sequence of commitments $\bar{c}$ where for every $i \in I'$, c_i is a commitment to p_i (all other c_i's can be commitments to "garbage"). Invoke A with the sequence $\bar{c}$. If A asks for $I = I'$ the simulation is successful. Otherwise, repeat* the entire process all over again *with a new sequence of plaintexts. With overwhelming probability, the number of repetitions is smaller than ℓn.*

Semantic security for independent plaintexts A common first reaction to the scenario of selective decommitment is that it should not really pose a problem. An important factor in this intuition is that the commitment of each plaintext p_i is performed in an *independent manner* (*i.e.*, using an independent random string r_i). Therefore, one may try to argue that the unopened commitments are independent of the opened ones and are still secure. What this intuition ignores is that, although the random strings used for the different commitments are independent, the *plaintexts themselves may be mutually dependent.* Therefore, the sequence of commitments and their opening may also relate to the unopened plaintexts and it is hard to rule out the possibility that these values give additional information to the adversary. A natural question raised by this discussion is *what happens if the plaintexts are also independently chosen?* Consider a product distribution $\Pi_n = \Pi_n^1 \times \Pi_n^2 \times \ldots \times \Pi_n^m$ (*i.e.*, p_i is independently chosen from Π_n^i). Is the selective decommitment still a problem for such a distribution? The initial answer is disappointing: It is not clear how such a distribution helps to simulate the adversary A. Therefore, achieving semantic security with respect to relations still seems hard. Nevertheless, we show here that for such distributions any secure commitment scheme $\mathcal{C}$ achieves some sort of semantic security with respect to functions. This form of semantic security is weaker than that of Definition 7.9 because M_1, M_2 depend on the constant c. Another way Theorem 7.14 is weaker is that it does not allow auxiliary input. The reason for that is that conditioned on a particular value z of the auxiliary input, the distribution Π_n on the plaintexts may no longer be a product distribution. Put differently, the auxiliary input introduces undesired dependencies between the plaintexts.

Theorem 7.14 *$\forall$ ptime $F \forall A_1 A_2 \forall \Pi_n$ which is a product distribution $\forall c \exists M_1 M_2$ such that for sufficiently large n*

$$|\Pr[A_2((B, A_1)(\Pi_n)) = F(\Pi_n)] - \Pr[M_2((B', M_1)(\Pi_n)) = F(\Pi_n)]| < \frac{1}{n^c}.$$

Proof. Fix F, A_1, A_2, c and a product distribution Π_n. It is enough to show that there exists M_1 and M_2 such that for sufficiently large n

$$\Pr[A_2((B, A_1)(\Pi_n)) = F(\Pi_n)] < \Pr[M_2((B', M_1)(\Pi_n)) = F(\Pi_n)] + \frac{1}{n^c}.$$

(the reason is that it is easy to decrease the success probability of M_2 if necessary).

We first describe a new machine M^{OPT} that ignores the sequence of commitments $\bar{c}$ but still does just as well as A_2. M^{OPT} is not necessarily efficient but can easily be approximated by an efficient machine M_2. It is interesting to note that the definition of M^{OPT} is not based on a simulation of A_2.

The machine M^{OPT} takes as input a transcript of a conversation between A_1 and B. It ignores all but the set I and the opened plaintexts $\bar{v} = \{p_i\}_{i \in I}$. For every possible value f of F, define $\mu_{I,\bar{v},f} = \Pr[F(\Pi_n) = f \mid \Pi_n|_{I=\bar{v}}]$. M^{OPT} outputs f that maximizes $\mu_{I,\bar{v},f}$.

Claim 7.15 $\Pr[A_2((B, A_1)(\Pi_n)) = F(\Pi_n)] < \Pr[M^{OPT}((B, A_1)(\Pi_n)) = F(\Pi_n)] + \nu(n)$.

Proof. Assume for the sake of contradiction that for infinitely many n,

$$\Pr[A_2((B, A_1)(\Pi_n)) = F(\Pi_n)] > \Pr[M^{OPT}((B, A_1)(\Pi_n)) = F(\Pi_n)] + 1/poly(n).$$

This implies that the output of A_2 has a significantly better chance to equal $F(\bar{p})$ than to equal $F(\bar{p}')$, where $\bar{p}'$ is obtained from $\bar{p}$ by replacing p_i for every $i \notin I$ with a new value independently sampled from Π_n^i. This is because, by its definition, the output of M^{OPT} is the best guess for $F(\bar{p}')$. Intuitively, this means that the commitments $\{c_i\}_{i \notin I}$ give useful information on the corresponding plaintexts. Indeed, as we now show, it is possible to construct from A_1 and A_2 a machine $\tilde{A}$ that breaks the commitment scheme $\mathcal{C}$. This implies a contradiction and completes the proof.

The desired properties of $\tilde{A}$ are the following: $\tilde{A}$ gets as input a uniformly distributed index $1 \leq i \leq m$, a random commitment $c_i = C(p_i, r_i)$ of a plaintext p_i sampled from Π_n^i, and a plaintext $\tilde{p}$. We will define $\tilde{A}$ such that it will distinguish the case that $\tilde{p} = p_i$ from the case that $\tilde{p}$ is a different plaintext independently sampled from Π_n^i. Formally, when i and r_i are uniform and p_i and p_i' are independently sampled from Π_n^i, we will have that

$$\Pr[\tilde{A}(i, p_i, C(p_i, r_i)) = 1] > \Pr[\tilde{A}(i, p_i', C(p_i, r_i)) = 1] + 1/poly(n).$$

We now define $\tilde{A}$. On input $(i, \tilde{p}, c_i)$ the algorithm $\tilde{A}$ operates as follows:

1. Sample a plaintext p_j independently from Π_n^j for all $j \neq i$. Generate a random commitment c_j of p_j for all $j \neq i$.

2. Invoke A_1 with input $(c_1, \ldots, c_m)$.

3. Let I be the subset selected by A_1. If $i \in I$, $\tilde{A}$ halts and outputs zero. Otherwise, open the commitments $\{c_j\}_{j \in I}$ by revealing the plaintexts and randomness $\{p_j, r_j\}_{j \in I}$ to A_1.

4. Continue the simulation of A_1 and A_2. Let f be the output of A_2.

5. Sample t uniformly at random in $\{0, \ldots, m-1\}$. If t is larger than $m-|I|-1$ (where $|I|$ stands for the size of I), $\tilde{A}$ halts and outputs zero. Otherwise, select t random indices $j \notin I \cup \{i\}$, denote this set by J. Sample for each $j \in J$ a new plaintext p'_j from Π_n^j. Complete these plaintexts to a sequence $\bar{p}' = (\tilde{p}_1 \ldots, \tilde{p}_m)$ by setting $\tilde{p}_j = p'_j$ for $j \in J$, setting $\tilde{p}_i = \tilde{p}$ (taken from the input of $\tilde{A}$) and by setting $\tilde{p}_k = p_k$ for all other indices.

6. Output 1 iff $F(\bar{p}') = f$.

We now observe some simple properties of $\tilde{A}$. All along, the probabilities are taken over the inputs of the various machines and their randomness. Particularly, i and r_i are uniform and p_i and p'_i are independently sampled from Π_n^i,

- Holding i fixed and letting r_i, p_i, p'_i be sampled as above, A_1 is invoked on random committments to random plaintexts. Therefore, the distribution on I does not depend on i. We can assume without loss of generality that $I \neq \{1, ..., m\}$ (if $\{1, ..., m\}$ is a legal subset then selective decommitment is trivial). It follows that when we now also choose i at random, the probabilty that $i \notin I$ is at least $1/m$.

- By the definition of $\tilde{A}$,

$$\Pr[\tilde{A}(i, p_i, C(p_i, r_i)) = 1 \mid i \notin I, t = 0] = \Pr[A_2((B, A_1)(\Pi_n)) = F(\Pi_n)].$$

- By the definition of $\tilde{A}$ and M^{OPT}

$$\Pr[\tilde{A}(i, p'_i, C(p_i, r_i)) = 1 \mid i \notin I, t = m - |I| - 1] \leq \Pr[M^{OPT}((B, A_1)(\Pi_n)) = F(\Pi_n)],$$

  as in this case the output of M^{OPT} is the best guess for $F(\bar{p}')$.

- By the definition of $\tilde{A}$, for every particular values $\ell < m$, $0 \leq j < m - \ell - 1$,

$$\begin{aligned} & \Pr[\tilde{A}(i, p'_i, C(p_i, r_i)) = 1 \mid i \notin I, |I| = \ell, t = j] \\ = \; & \Pr[\tilde{A}(i, p_i, C(p_i, r_i)) = 1 \mid i \notin I, |I| = \ell, t = j + 1]. \end{aligned}$$

The proof that $\tilde{A}$ breaks C now easily follows almost as in the standard hybrid argument:

$$
\begin{aligned}
& \Pr[\tilde{A}(i,p_i,C(p_i,r_i))=1] - \Pr[\tilde{A}(i,p_i',C(p_i,r_i))=1] \\
= & \sum_{\ell<m}\sum_{j=0}^{m-\ell-1} \Pr[\tilde{A}(i,p_i,C(p_i,r_i))=1 \mid i\notin I, |I|=\ell, t=j]\cdot \Pr[i\notin I, |I|=\ell, t=j] \\
& \quad - \sum_{\ell<m}\sum_{j=0}^{m-\ell-1} \Pr[\tilde{A}(i,p_i',C(p_i,r_i))=1 \mid i\notin I, |I|=\ell, t=j]\cdot \Pr[i\notin I, |I|=\ell, t=j] \\
= & \frac{1}{m}\Bigg(\sum_{\ell<m}\sum_{j=0}^{m-\ell-1} \Pr[\tilde{A}(i,p_i,C(p_i,r_i))=1 \mid i\notin I, |I|=\ell, t=j]\cdot \Pr[i\notin I, |I|=\ell] \\
& \quad - \sum_{\ell<m}\sum_{j=0}^{m-\ell-1} \Pr[\tilde{A}(i,p_i',C(p_i,r_i))=1 \mid i\notin I, |I|=\ell, t=j]\cdot \Pr[i\notin I, |I|=\ell]\Bigg) \\
= & \frac{1}{m}\Bigg(\sum_{\ell<m} \Pr[\tilde{A}(i,p_i,C(p_i,r_i))=1 \mid i\notin I, |I|=\ell, t=0]\cdot \Pr[i\notin I, |I|=\ell] \\
& \quad - \sum_{\ell<m} \Pr[\tilde{A}(i,p_i',C(p_i,r_i))=1 \mid i\notin I, |I|=\ell, t=m-|I|-1]\cdot \Pr[i\notin I, |I|=\ell]\Bigg) \\
= & (\Pr[i\notin I]/m)\Big(\Pr[\tilde{A}(i,p_i,C(p_i,r_i))=1 \mid i\notin I, t=0] \\
& \quad - \Pr[\tilde{A}(i,p_i',C(p_i,r_i))=1 \mid i\notin I, t=m-|I|-1]\Big) \\
\geq & (\Pr[i\notin I]/m)\Big(\Pr[A_2((B,A_1)(\Pi_n))=F(\Pi_n)] - \Pr[M^{OPT}((B,A_1)(\Pi_n))=F(\Pi_n)]\Big) \\
> & 1/poly(n).
\end{aligned}
$$

□

It is easy to define an *efficient* machine M_2 that approximates M^{OPT}. Note that given the set I and the opened plaintexts $\bar{v} = \{p_j\}_{j\in I}$, it is possible to efficiently sample the distribution $\Pi_n|_{I=\bar{v}}$ (this follow immediately from independence of the plaintexts). Therefore, we can define M_2 as follows: (1) Sample $poly(n)$ independent values $\bar{p}'$ from the above distribution. (2) For each such value compute $f = F(\bar{p}')$. (3) Output a value f that is obtained the most times. For this f it is easy to argue that (when setting *poly* to be a large enough polynomial) with overwhelming probability $\mu_{I,\bar{v},f} \geq \mu_{I,\bar{v},f'} - \frac{1}{4n^c}$ where f' is any other possible value in the range of F. The proof of Claim 7.15 shows that the success probability of A_2 in guessing $F(\bar{p})$ cannot be non-negligably larger then the success of M^{OPT} in guessing $F(\bar{p}')$ where $\bar{p}'$ is sampled from $\Pi_n|_{I=\bar{v}}$. The argument above shows that the success of M_2 in guessing $F(\bar{p}')$ is not off by much. It therefore follows that for sufficiently large n

$$\Pr[A_2((B,A_1)(\Pi_n)) = F(\Pi_n)] < \Pr[M_2((B,A_1)(\Pi_n)) = F(\Pi_n)] + \frac{1}{3n^c}.$$

Finally we define a machine M_1 that also ignores the sequence of commitments $\bar{c}$ and can replace A_1. The machine M_1 simply invokes A_1 on a new (random) sequence of commitments $\bar{c}'$ and selects the same set I chosen by A_1.

Claim 7.16 *For sufficiently large n*

$$\Pr[M_2((B, A_1)(\Pi_n)) = F(\Pi_n)] < \Pr[M_2((B, M_1)(\Pi_n)) = F(\Pi_n)] + \frac{1}{3n^c}.$$

Proof. The success probability of M_2 is only a function of $\bar{p}$ and I. Assume that the sequence $\bar{c}$ helps A_1 select a set I for which this success probability is significantly better (compared to the case that A_1 gets $\bar{c}'$). Then A_1 can easily be used to break the commitment scheme $\mathcal{C}$. We can define an algorithm $\tilde{A}$ that gets a sequence of plaintexts $\bar{p}$ and a sequence of commitments $\tilde{c}$ and distinguishes the case that $\tilde{c}$ contains commitments to $\bar{p}$ from the case that $\tilde{c}$ contains random commitments. This contradicts the assumption that $\mathcal{C}$ is a secure commitment scheme. One way of defining $\tilde{A}$ is the following:

1. Invoke A_1 with input $\tilde{c}$.
2. Let I be the subset selected by A_1.
3. Sample a uniform bit σ. If $\sigma = 1$ set $\tilde{p} = \bar{p}$. Otherwise sample $\tilde{p}$ from the distribution Π_n. Denote $\tilde{p} = (\tilde{p}_1 \ldots, \tilde{p}_m)$.
4. Simulate M_2 with I and $\{\tilde{p}_i\}_{i \in I}$. Let f be the output of M_2.
5. If $F(\tilde{p}) = f$, output σ otherwise output $1 - \sigma$.

□

Since both M_1 and M_2 ignore the sequence of commitments $\bar{c}$, they can interact with B' (instead of B). We therefore conclude the theorem. □

7.4.2 Semantic Security of a Class of Strong-Receiver Commitments

As discussed above, achieving semantic security with respect to relations for selective decommitment requires an efficient way to simulate the adversary selector A. In fact, it is hard to imagine a proof that achieves even weaker notions of security such as semantic security with respect to functions (Definition 7.9) which is not based on a simulation of A (to some extent, the proof of Theorem 7.14 is an exception). The proof of Theorem 7.12 provides a naïve method of such a simulation. In essence, this method requires the machine M to guess the subset I that A will ask to open. For M to have a non-negligible chance of guessing I, the number of legal subsets must be rather small. Unfortunately however, this is the only method of simulation we are aware of that is applicable to every commitment scheme. For the results here and in Section 7.4.3 we use entirely different methods of simulation. These simulations have the following meta structure:

1. Create a random sequence of commitments $\bar{c}$ and invoke A with the sequence $\bar{c}$.
2. Let I be the subset selected by A, ask B' for the plaintexts $\{p_i\}_{i \in I}$.
3. Open $\{c_i\}_{i \in I}$ as commitments to $\{p_i\}_{i \in I}$.

At first, such a simulation seems impossible: The sequence $\bar{c}$ is fixed before M learns the values $\{p_i\}_{i\in I}$. Therefore, to perform Step (3), M should be able to (efficiently) open the commitments $\{c_i\}_{i\in I}$ in many different ways. This seems to contradict the definition of commitment scheme. Moreover, for some commitment schemes (*i.e.*, strong-sender) it is impossible even to a computationally unbounded M. Nevertheless, we show ways to bypass this contradiction for two special types of commitment schemes: (1) Trapdoor strong-receiver commitments and (2) Random oracle commitments.

To consider trapdoor commitments, we first need to extend our discussion to efficient *ensembles* $\mathcal{C} = \{\mathcal{C}_k^n\}$ of commitment schemes (where n is the security parameter and k is the key). For such an ensemble, each one of the two properties of commitment schemes (being binding and secure) may either hold for every key k or with high probability over the choice of k. Trapdoor strong-receiver commitments are ensembles of strong-receiver commitment schemes such that every key k has a trapdoor t with the following property: Given the trapdoor t and a pair of commitment c and plaintext p it is easy to sample uniformly at random a string r that satisfies $c = C_k^n(p, r)$. Naturally, given the key k it should be hard to come up with the corresponding trapdoor t. However, it should also be easy to sample a key k with its corresponding trapdoor t. Consider for example the commitments of [39] that are based on the hardness of the discrete log. In this case the key contains a prime Q and a pair of generators g and h of Z_Q^*. The commitment $\mathcal{C}_{g,h,Q}$ is defined by $\mathcal{C}_{g,h,Q}(p, r) = g^p h^r \bmod Q$. The trapdoor is α such that $h = g^\alpha \bmod Q$. Given h and g computing α is exactly the discrete log problem. However, one can sample g and h by first sampling g and then sampling α.

We now use the simulation method described above to prove the security of trapdoor strong-receiver commitment schemes for the selective decommitment problem. However, neither Definition 7.2 nor Definition 7.9 immediately applies to *commitment ensembles.* Still we show in Theorem 7.17, that trapdoor schemes obtain some sort of semantic security with respect to relations. Since there are several ways Definition 7.2 can be extended to commitment ensembles, we emphasize two properties of what we managed to prove: (1) Security is achieved for a random key k (rather than any key) and (2) The relation R and the distribution D do not depend on the key k.

Theorem 7.17 *Let $\{\mathcal{C}_k^n\}$ be an efficient ensemble of trapdoor strong-receiver commitment schemes. For every k, denote by $(B, A_1)(k, \Pi_n, Z_n)$ the protocol where B uses the commitment scheme $\mathcal{C}_k^n$ (and k is a joint input of B and A_1). Then $\forall R \forall A_1 A_2 \forall D = \{\Pi_n Z_n\} \exists M_1 M_2$*

$$\begin{aligned} |\Pr_k[&R(\Pi_n, Z_n, A_2((B, A_1)(k, \Pi_n, Z_n), k, Z_n))] \\ &- \Pr[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n), Z_n))]\,| < \nu(n). \end{aligned}$$

(The notation $\Pr_k$ indicates that the probability is also *over the choice of the key k.)*

Proof. Define the machine M_1 such that on input z it simulate a conversation of A_1 with B as follows:

1. Sample a key k' with its trapdoor t'.

2. Sample a random sequence of commitments $\bar{c}$.

3. Invoke A_1 with input k', z and give it the sequence $\bar{c}$.

4. Let I be the subset selected by A_1. Ask B' for the plaintexts $\{p_i\}_{i \in I}$.
5. Use the trapdoor t' to open $\{c_i\}_{i \in I}$ as commitments to $\{p_i\}_{i \in I}$. Give these values to A_1.
6. Output the simulated conversation ξ of A_1 with B.

The machine M_2 outputs $A_2(\xi, k', z)$.

As $\mathcal{C}_k^n$ is a family of trapdoor *strong-receiver* commitments, we easily have that the simulated conversation ξ *together with the inputs* $k', \bar{p}, z$, are identical, or at least statistically close (depending on the commitment scheme), to these variables in a real conversation of A_1 with B. The lemma then easily follows. □

As mentioned above, the relation R in the statement of Theorem 7.17 does not have direct access to the key k (and similarly the distribution D does not depend on k). One way to interpret this fact is that the interaction with B is not helpful if A tries to learn something about the sequence of plaintexts $\bar{p}$ alone. It is also easy to see that A does not learn anything new about the key k. Nevertheless, we are not able to rule out the possibility that A learns something about the combination of the plaintexts and the key. In some settings this may be a problem (especially when the same key is used many times). Somewhat related is the next theorem, which shows that for any relation R there are not too many "bad" keys (*i.e.*, keys that give A a non-negligible advantage in satisfying R).

Theorem 7.18 *Let $\{\mathcal{C}_k^n\}$ be an efficient ensemble of trapdoor strong-receiver commitment schemes. For every key k, denote by $(B, A_1)(k, \Pi_n, Z_n)$ the protocol where B uses the commitment scheme $\mathcal{C}_k^n$ (and k is a joint input of B and A_1). Then $\forall R \, \forall A_1 A_2 \, \forall D = \{\Pi_n Z_n\} \, \forall c \, \exists M_1 M_2$ such that if we define for every key k*

$$p_k^n = \Pr[R(\Pi_n, Z_n, A_2((B, A_1)(k, \Pi_n, Z_n), k, Z_n))],$$

then for sufficiently large n

$$\Pr_k[p_k^n - \Pr[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n), Z_n))] \geq \frac{1}{n^c}] < \frac{1}{n^c}.$$

Proof. The machines M_1 and M_2 that satisfy the requirement in the statement of the theorem are defined in almost the same way as in the proof of Theorem 7.17. The only difference is in the way M_1 chooses the key k' (Step (1) in its definition). Note that for every key k, the definition of M_1 and M_2 implies that

$$|p_k^n - \Pr[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n), Z_n)) \mid k' = k] \,| \leq \nu(n).$$

Therefore, in order to obtain the theorem it is enough to select a key k' with a relatively large value $p_{k'}$. More specifically, we will select a key k', such that with overwhelming probability $p_{k'} \geq p_{large} - \frac{1}{2n^c}$, where p_{large} is defined to be the maximal value such that $\Pr_k[p_k^n \geq p_{large}] \geq \frac{1}{2n^c}$. Note that by this definition $\Pr_k[p_k^n > p_{large}] < \frac{1}{2n^c}$. Selecting such a key k' is done using standard sampling techniques as follows:

- Set $\ell = n^{c+1}$. Instead of sampling a single key k' (with its trapdoor t'), M_1 now samples ℓ keys with their trapdoors: $\{\langle k_1, t_1 \rangle \ldots \langle k_\ell, t_\ell \rangle\}$. This guarantees that with overwhelming probability, for some i, $p_{k_i} \geq p_{large}$.

- For each i, estimate p_{k_i} up to error $\frac{1}{4n^c}$ and set $\langle k', t'\rangle = \langle k_i, t_i\rangle$ for i with the highest estimated p_{k_i}.

  To estimate p_{k_i}, M_2 can simulate many (say n^{2c+1}) independent conversations $(B, A_1)(k_i, \Pi_n, Z_n)$ (each time M_2 samples $\bar{p}, z$ and runs $(B, A_1)(k_i, \bar{p}, z)$ playing the role of both B and A_1).

Note that if all the estimates are accurate up to error $\frac{1}{4n^c}$ (which happens with overwhelming probability), then for all i we have that $p_{k'} \geq p_{k_i} - \frac{1}{2n^c}$. Therefore with overwhelming probability we indeed have that $p_{k'} \geq p_{large} - \frac{1}{2n^c}$.

We can conclude that

$$\begin{aligned} & \Pr_k[p_k^n - \Pr[R(\Pi_n, Z_n, M_2((B', M_1)(\Pi_n, Z_n), Z_n)) \geq \frac{1}{n^c}] \\ \leq\ & \Pr_k[p_k^n > p_{large}] + \Pr[p_{k'} < p_{large} - \frac{1}{2n^c}] \\ <\ & \frac{1}{n^c}. \end{aligned}$$

□

7.4.3 Commitments Using a Random Oracle

We now consider the seemingly "perfect" commitment scheme that exist in *the random oracle model*: Let $\mathcal{C}$ be the commitment scheme defined by $\mathcal{C}(x, r) \stackrel{\text{def}}{=} H(x, r)$, where H is a random oracle (*i.e.*, a random function accessible to all parties). We show that $\mathcal{C}$ achieves a strong notion of semantic security for selective decommitment (although not as strong as one may wish). Similarly to the case of commitment ensembles, in order to define the selective decommitment problem in the random oracle model it is important to decide what may depend on the oracle H. We are able to show that the commitment scheme $\mathcal{C}$ achieves semantic security *with respect to any relation R that does not depend on H*. The distribution D as well as all the machines involved may depend on H.

Theorem 7.19 *Let (B^H, A_1^H) denote the protocol where B^H uses the commitment scheme $\mathcal{C}(x, r) \stackrel{\text{def}}{=} H(x, r)$. Then, $\forall R \forall A_1 A_2 \forall D = \{\Pi_n Z_n\} \exists M_1 M_2$*

$$\begin{aligned} & |\Pr_H[R(\Pi_n^H, Z_n^H, A_2^H((B^H, A_1^H)(\Pi_n^H, Z_n^H), Z_n^H))] \\ & \quad - \Pr_H[R(\Pi_n^H, Z_n^H, M_2^H((B'^H, M_1^H)(\Pi_n^H, Z_n^H), Z_n^H))]\,| < \nu(n). \end{aligned}$$

Proof. The machines M_1 and M_2 simulate A_1 and A_2 as follows:

- Invoke A_1 with a uniformly distributed sequence $\bar{c}$. Answer all the queries of A_1 to H by querying H itself.
- Let I be the subset selected by A_1. Ask B' for the plaintexts $\{p_i\}_{i \in I}$.
- Open $\{c_i\}_{i \in I}$ as commitments to $\{p_i\}_{i \in I}$: Sample uniformly chosen strings $\{r_i\}_{i \in I}$ and give A_1 the values $\{p_i, r_i\}_{i \in I}$ as "decommitments".

- Continue the simulation of A_1 and then invoke A_2. Answer the queries of A_1 and A_2 by querying H itself *on all points that differ from* p_i, r_i. On the points $\{p_i, r_i\}_{i \in I}$ answer consistently with $\bar{c}$.

For the sake of the proof, let us assume that at the end of the simulation, M_2 also samples uniformly chosen strings $\{r_i\}_{i \notin I}$. Now define the oracle H' which is identical to H on all points that differ from p_i, r_i. On the points $\{p_i, r_i\}$ answer consistently with $\bar{c}$. That is, define $H'(p_i, r_i) = c_i$. As $\bar{c}$ and H are uniformly distributed, we have that H' is also uniformly distributed. Note that there is only a negligible probability that one of the points $\{p_i, r_i\}$ is identical a query to H made during the sampling of $\bar{p}, z$. Also note that there is only a negligible probability that one of the points $\{p_i, r_i\}_{i \in I}$ is identical to a query of A_1 made before A_1 received the decommitments. Finally there is only a negligible probability that one of the points $\{p_i, r_i\}_{i \notin I}$ is identical to *any* query of A_1 or A_2. In case none of these rare events happen, the probability that M_2's output satisfies the relation R is identical to the probability that A_2's output does. □

The fact that the relation R may not depend on H is certainly a disadvantage. In particular, we cannot claim that the corresponding 3-round argument discussed in Section 7.2.2 is zero-knowledge (it is zero-knowledge in the weak sense where the simulator is allowed to "change" H). The following theorem shows that the commitment scheme $\mathcal{C}$ achieves semantic security even with respect to functions that *depend on* H.

Theorem 7.20 *Let* (B^H, A_1^H) *denote the protocol where* B^H *uses the commitment scheme* $\mathcal{C}(x,r) \stackrel{\text{def}}{=} H(x,r)$. *Then,* $\forall$ *ptime* $F\, \forall A_1 A_2\, \forall D = \{\Pi_n Z_n\}\, \exists M_1 M_2$

$$\begin{aligned} |\Pr_H[A_2^H((B^H, A_1^H)(\Pi_n^H, Z_n^H), Z_n^H) = F^H(\Pi_n^H, Z_n^H)] \\ - \Pr_H[M_2^H((B'^H, M_1^H)(\Pi_n^H, Z_n^H), Z_n^H) = F^H(\Pi_n^H, Z_n^H)]\,| < \nu(n). \end{aligned}$$

Proof. Define M_1 and M_2 exactly as in the proof of Theorem 7.19. Also define H' in the same way as in that proof. Again we have the following facts: (1) H' is uniformly distributed. (2) There is only a negligible probability that one of the points $\{p_i, r_i\}$ is identical a query to H made during the sampling of $\bar{p}, z$. (3) There is only a negligible probability that one of the points $\{p_i, r_i\}_{i \in I}$ is identical to a query of A_1 made before A_1 received the decommitments. (4) There is only a negligible probability that one of the points $\{p_i, r_i\}_{i \notin I}$ is identical to *any* query of A_1 or A_2. One additional fact we need to note here is that there is only a negligible probability that one of the points $\{p_i, r_i\}$ is identical to any query made by F^H when applied to $\bar{p}, z$. In case none of these rare events happen, the probability that M_2's output equals $F^H(\bar{p}, z)$ is identical to that of A_2's output. □

8 Aggregate-Probability Versus Almost-All-Individuals

As mentioned in Section 4, one of the original motivations for the uniform definition of zero-knowledge, *e.g.*, Definition 4.5, was to relax somewhat the requirement of Definition 4.3, that the zero-knowledge property must hold for all (x, y, z). Goldreich [19] has noted that the uniform definition, given in terms of aggregate probabilities, is equivalent to a definition saying that the zero-knowledge property holds for almost all individual (x, y, z)'s in the sense that it is infeasible to find by random sampling a (x, y, z) not having the zero-knowledge property. We refer to a definition of the first type, *e.g.*, Definition 4.5, as an *aggregate probability*, (AP) definition, and a definition of the second type as an *almost-all-individuals* (AAI) definition. In Section 4 we have given only the AP version at each level that defines a form of uniform zero-knowledge (Definition 4.5 and below). At each level of our hierarchy, starting at Definition 4.5, one can also define an AAI version. At a given level, the two types of definitions have exactly the same quantifier prefix; they differ only in the predicate following the quantifiers. To illustrate this, we recall the (AP) Definition 4.5, and give its AAI version. To help illustrate the similarities, we replace $\nu(n)$ by its definition in the AP version. Although we include the augmentations for consistency with Section 4, the augmentations do not play any role in the proofs in this section. Let $\forall_{\mathrm{ae}} n$ abbreviate "for all sufficiently large n", *i.e.*, $\exists n_0 \, \forall n \geq n_0$.

Definition 8.1 (AP Augmented Uniform $S(V)$ Zero-Knowledge (over W))
$\forall V^* \, \exists S \, \forall T \, \forall D = \{X_n Y_n Z_n\} \, \forall c \, \forall_{\mathrm{ae}} n$

$$|\Pr[T(X_n, Y_n, Z_n, (P, V^*)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < n^{-c}.$$

We next give the corresponding AAI version, also due to Goldreich [19]. In giving AAI definitions, the following notation is useful: for a given P, V, T, S (which will always be clear from context), a number c, and a size n, define

$$B_n^{(c)} = \{ (x, y, z) \in \Sigma_n^X \Sigma_n^Y \Sigma_n^Z \mid |\Pr[T(x, y, z, (P, V^*)(x, y, z))] - \Pr[T(x, y, z, S(x, z))]| > n^{-c} \}.$$

Intuitively, $B_n^{(c)}$ is the set of (x, y, z) of size n that are "bad" for the simulator S with respect to T, in that S is off by more than n^{-c}. Intuitively, an AAI definition says that the bad sets have negligible density.

Definition 8.2 (AAI Augmented Uniform $S(V)$ Zero-Knowledge (over W))
$\forall V^* \, \exists S \, \forall T \, \forall D = \{X_n Y_n Z_n\} \, \forall c \, \forall_{\mathrm{ae}} n$

$$\Pr[(x, y, z) \in B_n^{(c)}] < n^{-c}.$$

As proved below, Definitions 8.1 and 8.2 are equivalent.

It is natural to ask, for each uniform level in our hierarchy, whether the AP and AAI versions are equivalent at that level. (It should be noted that equivalence or inequivalence of the two versions at one level does not imply anything about this question at any other level, using only that each level is a weakening of the one above.) Although we do not have a complete answer to this question, we have identified the level at which the known proof of equivalence breaks down. One direction of this equivalence, that the AAI definition implies the AP defintion, is trivial (Lemma 8.3). Although the other direction appears equally trivial on first glance, a subtle difference between the AP and AAI definitions seems to need a "nontrivial" proof of this direction (Lemma 8.5). Our hierarchy highlights the change in the quantification structure that causes even our nontrivial argument to fail.

We next give three lemmas. As a consequence of the first lemma, the AAI definition implies the AP definition at each level of the hierarchy.

Lemma 8.3 *For all P, V^*, T, D, S, c, c' with $c' < c$, and all sufficiently large n, if* $\Pr[(x, y, z) \in B_n^{(c)}] < n^{-c}$ *then* $|\Pr[T(X_n, Y_n, Z_n, (P, V)(X_n, Y_n, Z_n))] - \Pr[T(X_n, Y_n, Z_n, S(X_n, Z_n))]| < n^{-c'}$.

Proof. If the hypothesis of the lemma holds, then the absolute value in the conclusion is less than $(1-n^{-c})n^{-c}+n^{-c}<n^{-c'}$ for all sufficiently large n. $\square$

The converse of this lemma does not hold, because the absolute values occur in the AAI definition at a lower level than in the AP definition, permitting more cancelations in the AP definition. For example, $|\frac{1}{2}-0|$ and $|\frac{1}{2}-1|$ both equal $\frac{1}{2}$, but $|(\frac{1}{2}-0)+(\frac{1}{2}-1)|=0$. One case where this cannot occur is when T is a "validity check", as in Definitions 4.13 and 4.14, that almost always outputs 1 on real conversations.

Lemma 8.4 *Assume that for all c and all sufficiently large n,*

$$\Pr[T(X_n,Y_n,Z_n,(P,V^*)(X_n,Y_n,Z_n))]>1-n^{-c} \text{ and}$$
$$|\Pr[T(X_n,Y_n,Z_n,(P,V^*)(X_n,Y_n,Z_n))]-\Pr[T(X_n,Y_n,Z_n,S(X_n,Z_n))]|<n^{-c}.$$

Then $\Pr[(x,y,z)\in B_n^{(c)}]<n^{-c}$ *for all c and all sufficiently large n.*

Proof. The assumptions of the lemma imply that for all c and for all sufficiently large n,

$$\Pr[T(x,y,z,(P,V^*)(x,y,z))]>1-n^{-c} \quad\text{and}\quad \Pr[T(x,y,z,S(x,z))]>1-n^{-c}$$

for all but a n^{-c} fraction of the (x,y,z)'s. Because a probability cannot be larger than 1, it follows for all c and all sufficiently large n, that these two probabilities are within n^{-c} of each other for all but a n^{-c} fraction of the (x,y,z)'s. The conclusion of the lemma is now immediate from the definition of $B_n^{(c)}$. $\square$

Recall the abbreviations $\pi_T(n)$, $\pi_T(x,y,m)$, $\pi_T^S(n)$, and $\pi_T^S(x,y,m)$ from Section 6.3, where in this context M_n is replaced by Z_n and m is replaced by z. The AP and AAI versions of Definition 4.5 are given in Definitions 8.1 and 8.2, respectively.

Lemma 8.5 *For Definitions 4.5 and 4.7, the AP version implies the AAI version.*

Proof. We give the proof for Definition 4.5. The proof for Definition 4.7 is identical, because the only fact about the quantification order used in the proof is that D comes after V,T,S. We prove the contrapositive. Assume that P does not satisfy the AAI version of Definition 4.5, that is,

$$\exists V\,\forall S\,\exists T\,\exists D=\{X_nY_nZ_n\}\,\exists c\,\exists \text{ an infinite set } I\,\forall n\in I:\ \Pr[(x,y,z)\in B_n^{(c)}]\geq n^{-c}. \tag{22}$$

To show that P does not satisfy the AP version, we show that for the same V, for all S there exists a T (the same T as in (22)), and there exists a $D'=\{X_nY_nZ_n\}$ and a c' such that for all n in I (the same I as in (22)),

$$|\Pr_{D'}[T(X_n,Y_n,Z_n,(P,V)(X_n,Y_n,Z_n))]-\Pr_{D'}[T(X_n,Y_n,Z_n,S(X_n,Z_n))]|\geq n^{-c'}, \tag{23}$$

where $\Pr_{D'}$ means that the distribution of (x,y,z) is given by D' rather than D.

Let S be arbitrary. At a high level, we first check if the AP definition fails to be satisfied on D directly. If that is the case, then we are done. Otherwise we modify D to obtain D' as follows. From $\neg AAI$ we show it is possible to find many triples (x,y,z) that are bad for S with respect to T. We then define D' to consist of those "bad" triples that we find.

Using the assumption (22), let T, D, c, and I be such that $\Pr[(x, y, z) \in B_n^{(c)}] \geq n^{-c}$ for all $n \in I$. The ptime sampler D' operates as follows when given input 1^n. For a constant r suitably larger than c, D' computes estimates $\tilde{\pi}_T(n)$ and $\tilde{\pi}_T^S(n)$ to $\pi_T(n)$ and $\pi_T^S(n)$, respectively, that are accurate to within $\pm n^{-r}$ with probability at least $1 - n^{-r}$. The sampler D' can do this using a polynomial number of trials that depends on r. If $|\,\tilde{\pi}_T(n) - \tilde{\pi}_T^S(n)\,| \geq n^{-3c}$ then D' calls D with input 1^n to obtain a (x, y, z), and outputs this (x, y, z). Intuitively, in this case, $\pi_T(n)$ and $\pi_T^S(n)$ are far enough apart for this n that the original D and any $c' > 3c$ works.

If the first case does not hold, then we can assume that $|\,\pi_T(n) - \pi_T^S(n)\,| \leq n^{-2c}/2$ with probability at least $1 - n^{-r}$.

If the first case did not hold, then D' calls D to obtain a (x, y, z). It then computes good (in the sense above) estimates $\tilde{\pi}_T(x, y, z)$ and $\tilde{\pi}_T^S(x, y, z)$ for $\pi_T(x, y, z)$ and $\pi_T^S(x, y, z)$, respectively. If $\tilde{\pi}_T(x, y, z) \geq \tilde{\pi}_T^S(x, y, z) + n^{-4c}$, then D' outputs (x, y, z). Otherwise, D' calls D again to obtain another (x, y, z), and continues in the same way. If this procedure has not stopped after n^t calls to D, for a suitably large constant t, then D' outputs the (x, y, z) given by D at an arbitrary one of the trials. In the next paragraph, we show that with high probability (that can be made $> 1 - n^{-q}$ for any q by choosing t large enough), D' finds a (x, y, z) with $\tilde{\pi}_T(x, y, z) \geq \tilde{\pi}_T^S(x, y, z) + n^{-4c}$ before stopping. It should then be clear that (23) holds with $c' = 5c$, because with probability at least $1 - n^{-q} - n^{-r}$, D' outputs a (x, y, z) with $\pi_T(x, y, m) \geq \pi_T^S(x, y, m) + n^{-4c} - 2n^{-r}$.

Let $\varepsilon = n^{-3c}$. It is sufficient to show that at least a fraction ε of the (x, y, z) under distribution D satisfy $\pi_T(x, y, z) \geq \pi_T^S(x, y, z) + \varepsilon$, because t can be chosen arbitrarily larger than c. Suppose to the contrary that at least a fraction $1 - \varepsilon$ of the (x, y, z) have $\pi_T(x, y, z) < \pi_T^S(x, y, z) + \varepsilon$. If $n \in I$, we know by $\Pr[(x, y, z) \in B_n^{(c)}] \geq n^{-c}$ that at least a fraction n^{-c} of the (x, y, z) satisfy $|\,\pi_T(x, y, z) - \pi_T^S(x, y, z)\,| > n^{-c}$. Note that $\varepsilon < n^{-c}$. Therefore, in the fraction $n^{-c} - \varepsilon$ overlap of the $1 - \varepsilon$ fraction and the n^{-c} fraction, we have both $\pi_T(x, y, z) < \pi_T^S(x, y, z) + \varepsilon$ and $|\,\pi_T(x, y, z) - \pi_T^S(x, y, z)\,| > n^{-c}$, which implies that $\pi_T^S(x, y, z) > \pi_T(x, y, z) + n^{-c}$ for all (x, y, z) in this overlap because $\varepsilon < n^{-c}$. It follows that

$$\pi_T^S(n) - \pi_T(n) > (n^{-c} - \varepsilon)(n^{-c}) + (1 - n^{-c})(-\varepsilon) + \varepsilon(-1) = n^{-2c} - 2n^{-3c}.$$

This contradicts the assumption (because the first case did not hold) that $|\,\pi_T(n) - \pi_T^S(n)\,| \leq n^{-2c}/2$ with probability at least $1 - n^{-r}$. Any errors in the argument above occur with probability $O(n^{-r} + n^{-q})$, which can be made negligible compared to n^{-5c}. $\square$

This proof breaks down at the level of uniform $S(V, T, D)$ zero-knowledge, because the proof relies on the fact that D comes after V, T, S in the quantification order. For $S(V, T, D)$ definitions of uniform zero knowledge, D comes before S.

The following is immediate from Lemmas 8.3, 8.4, and 8.5. (For Definitions 4.11, 4.13, and 4.14 where T does not get y, the definition of $B_n^{(c)}$ should be modified by removing y from the inputs to T.)

Theorem 8.6 *For Definitions 4.5, 4.7, 4.13, and 4.14, the AP version and AAI version are equivalent.*

As noted above, the proof of Lemma 8.5 breaks down at the levels of Definitions 4.9 and 4.11 where the quantifiers start $\forall V\, \forall T\, \forall D\, \exists S$, and Lemma 8.4 does not apply. These two levels are identical, except that T gets y as input in Definition 4.9, but does not in Definition 4.11.

For Definition 4.11, we can show inequivalence of the two definitions in the case that all provers and verifiers are restricted to sending messages from a polynomial-size set, the protocol is constant-round, tests are deterministic, and simulators can be circuit families. It is relevant that the proof of equivalence at the $S(V)$ and $S(V,T)$ levels holds without modification under this set of restrictions, provided that D can be a circuit family; this is needed so that D can estimate probabilities involving S. The interactive argument systems meeting these restrictions are uninteresting because they are argument systems only for languages in BPP. Our point is that is it possible to contrive an (admittedly strange) definition of weak zero-knowledge for which the AP and AAI definitions are equivalent at the $S(V)$ and $S(V,T)$ levels of the hierarchy, but are not equivalent at the $S(V,T,D)$ level.

9 Discussion and Future Work

We have established strong connections among the selective decommitment problem, the question"GAP?" of the existence of 3-round public-coin (weak) zero-knowledge protocols for languages outside of BPP, and the Fiat-Shamir methodology. In particular:

1. The existence of a "good" commitment scheme secure against selective decommitment implies the existence of 3-round public-coin moderately-weak zero-knowledge proofs for NP;

2. There exists a 3-round public-coin very-weak zero-knowledge proof for even one language outside of BPP if and only if our (generalized) version of the Fiat-Shamir methodology for converting 3-round public-key identification schemes into signatures fails. (If our version of the methodology fails, then the original version of the methodology, in which the acceptance test T for the signature is constrained essentially to be the same as in the identification scheme, also fails.)

We have obtained no definitive solution or impossibility result. This work shows that an answer to any of the three problems would be of major interest, in particular: *What kind of security can be shown for selective decommitment?*

$S(V,T,D)$ zero-knowledge has practical applications, in particular, in the non-malleable bit commitment protocol of Dolev, Dwork, and Naor [14]. In that protocol, there is a proof of consistency between two commitments. The zero-knowledge property is used in order to argue that no information is leaked about the bit itself. However, since this is the *only* bit that must be protected, and, for example, it is not necessary to completely protect the secret bits used in the commitments, we can assume that the test T is known (there aren't that many on a single-bit). It would be interesting to find other settings in which our various weakenings of zero-knowledge suffice.

There are weaknesses in the result $\neg\text{GAP} \Rightarrow \text{MAGIC}$, having to do with the feasibility of finding public keys and messages in the hard cores. (Finding messages is an issue only in the first model of Section 5 where messages are chosen randomly, rather than being chosen by the simulator.) One obstacle is that we show the existence of psize *circuits* for deciding membership in the hard cores, but we do not know how to construct them efficiently. This might be a difficult obstacle to overcome, given that the hard core protects against simulators that are themselves circuits. As has been noted, the proofs of the hard core results use that simulators are circuit families (although, the assumption of circuit simulators is made in both the assumption and the conclusion of the results). One direction toward overcoming this obstacle would be to prove analogous hard core results for the case that simulators are probabilistic ptime machines. Even if ptime machines for deciding membership in the hard cores are known, there is still an efficiency issue in using them to find members of the hard cores by random sampling, as described at the beginning of Section 6: the expected time to find a single member in a set is the reciprocal of the density of the set. In the case that the hard cores have small $1/\text{poly}(n)$ density, another question is whether there is a more efficient way to find members of the hard cores. This is less of an issue in finding a public key x and private key y, because this need be done only once to define the signature scheme, but it is more of an issue in finding messages in the hard core of messages for the chosen (x, y). In the case that the simulator may use a chosen message attack to choose a particular message, there is no separate hard core of messages; however, this requires a stronger assumption, that chosen-message-attack simulators cannot simulate well in the aggregate.

The positive results of Section 7.4 substantially advance understanding of the selective decommitment problem, demonstrating the limitations of standard definitions and proof techniques and helping to identify the heart of the problem. For example, the result for independent plaintexts (cross-product distributions) formalizes the intuition that a very important component of the selective decommitment problem lies in the dependence between the plaintexts. In addition, the results on strong-receiver selective decommitment are rather promising (especially compared with the previous state of knowledge). Nevertheless, our results leave (and suggest) many interesting open questions. For example, can one prove that *every strong-receiver commitment scheme* achieves semantic security with respect to functions for the selective decommitment problem? Can such a result hold *for every key* of the commitment scheme (rather than with high probability over the choice of the key)?

As mentioned in Section 2, Canetti, Goldreich, and Halevi [10] considered relations $R(x_1, \ldots, x_\ell, f_k(x_1), \ldots, f_k(x_\ell))$ where each of $x_1, \ldots, x_\ell$ is an n-bit string, and showed that if f_k takes inputs of length n then f_k is not ℓ-input ptime correlation intractable for any $\ell \geq \frac{|k|}{n}$. The relations that cause difficulty are those that depend on the choice of k. Intuitively, Theorems 7.17 and 7.18 suggest a converse to the Canetti et al. results, saying that the only kind of things that might conceivably be learned are things that depend on k. It would be significant if such a statement (once made rigorous) were to prove true.

Another interesting avenue for future research is the relation between selective decommitment and selective decryption. Can we solve one using the other? In particular, results along the lines of what we have obtained for trapdoor strong-receiver commitment may be sufficient for certain applications, such as the distribution of clip-art on CD-ROMS (users receive the CD-ROMs containing encrypted art for free, and pay for sets of images to be "opened"). The key k can be chosen independently for every CD-ROM, and it may be safe to trust the manufacturer not to "lie" about the revealed values.

A practical variant on the selective decryption problem is to require that the decryption information be *substantially shorter* than the ciphertext to be decrypted. This version has obvious application in e-commerce: encrypted content is available for free. The (short) decryption key for an item or a set of items is available for a fee. Purchasing a set of keys should not endanger the security of remaining ciphertexts. What type of security can be achieved in this setting?

The problems studied in this work are closely related to *circuit obfuscation* or *code obfuscation*. Our intuition is that if pseudo-random functions can be obfuscated, then 3-round zero-knowledge is possible: the obfuscated function acts as a magic function. However, the exact relationship should be understood. After this work first appeared, careful definitions of obfuscation and a proof that there is no general purpose obfuscator were obtained by [4]. The lack of a general purpose obfuscator does not settle the question of obfuscation of the specific families of circuits or functions relevant to the questions studied in this paper.

Acknowledgements. We are grateful to Amos Fiat, Russell Impagliazzo, and Mike Luby for helpful discussions regarding this work and the related literature. Thanks also to Oded Goldreich and Shai Halevi for many comments on earlier versions of this paper.

References

[1] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre, From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security, *Proc. EUROCRYPT 2002, Lecture Notes in Computer Science, vol. 2332*, Springer-Verlag, pp. 418–433.

[2] J.L. Balcazar, J. Diaz, and J. Gabarro, *Structural Complexity I*, Springer-Verlag, 1988.

[3] B. Barak, How to go beyond the black-box simulation barrier, *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, 2001.

[4] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, K. Yang, On the (im)possibility of obfuscating programs, *Advances in Cryptology – CRYPTO 2001, Lecture Notes in Computer Science, vol. 2139*, Springer-Verlag, pp. 1–18.

[5] M. Bellare, O. Goldreich, and R. Impagliazzo, The correlation intractability of ideal families, unpublished manuscript, 1998.

[6] M. Bellare, R. Impagliazzo, and M. Naor, Does parallel repetition reduce the error in computationally sound protocols?, *Proc. 38th IEEE Symp. on Foundations of Computer Science*, 1997, pp. 374–383.

[7] R. Canetti, C. Dwork, M. Naor and R. Ostrovsky, Deniable encryption, *Proc. CRYPTO'97, Lecture Notes in Computer Science, vol. 1294*, Springer-Verlag, 1997, pp. 90–104.

[8] R. Canetti, U. Feige, O. Goldreich, and M. Naor, Adaptively secure multiparty computation, *Proc. 28th ACM Symp. on Theory of Computing*, 1996, pp. 639–648.

[9] R. Canetti and R. Gennaro, Incoercible multiparty computation (extended abstract), *Proc. 37th IEEE Symp. on Foundations of Computer Science*, 1996, pp. 504–513.

[10] R. Canetti, O. Goldreich, and S. Halevi, The random oracle methodology, revisited, *Proc. 30th ACM Symp. on Theory of Computing*, 1998.

[11] L. Carter and M. Wegman, Universal classes of hash functions, *J. of Computer and System Sciences*, vol. 18, 1979, pp. 143-154.

[12] D. Chaum and R. Impagliazzo, personal communication.

[13] R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, *Advances in Cryptology – CRYPTO'98, Lecture Notes in Computer Science, vol. 1462*, Springer-Verlag, pp. 13–25.

[14] D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography, *SIAM J. Comput.* 30 (2000), pp. 391–437.

[15] C. Dwork, M. Naor and A. Sahai, Concurrent zero-knowledge, *Proc. 30th ACM Symp. on Theory of Computing*, 1998.

[16] U. Feige, A. Fiat, and A. Shamir, Zero knowledge proofs of identity, *J. of Cryptology* 1(2) (1988), pp. 77–94. (Preliminary version in STOC 87).

[17] U. Feige and A. Shamir, Witness indistinguishability and witness hiding protocols, *Proc. 22nd ACM Symp. on Theory of Computing*, 1990, pp. 416–426.

[18] A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, *Proc. CRYPTO'86, Lecture Notes in Computer Science, vol. 263*, Springer-Verlag, 1987, pp. 186–194.

[19] O. Goldreich, A uniform-complexity treatment of encryption and zero-knowledge, Technion CS-TR 570, June 1989, revised August 1998.

[20] O. Goldreich, private communication.

[21] O. Goldreich and H. Krawczyk, On the composition of zero knowledge proof systems, *SIAM J. Computing* 25(1) (1996), pp. 169–192.

[22] O. Goldreich, S. Micali, and A. Wigderson, Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems, *J.ACM* 38(3) (1991), pp. 691–729.

[23] O. Goldreich and Y. Oren, Definitions and properties of zero-knowledge proof systems, *J. of Cryptology* 7 (1994), pp. 1–32.

[24] S. Goldwasser and S. Micali, Probabilistic encryption, *J. Comput. Syst. Sci.* 28(2) (1984), pp. 270–299.

[25] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM J. Comput.* 18(1) (1989), pp. 186–208.

[26] S. Goldwasser, S. Micali, and R. L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, *SIAM J. Comput.* 17(2) (1988), pp. 281–308.

[27] S. Goldwasser and Y. Tauman, On the (in)security of the Fiat-Shamir paradigm, Cryptology ePrint Archive: Report 2003/034, *Proc. 2003 IEEE Symp. on Foundations of Computer Science.*

[28] S. Hada and T. Tanaka, A relationship between one-wayness and correlation intractability, *Foundations of Cryptography Library*, March, 1999.

[29] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby, A pseudorandom generator from any one-way function, *SIAM J. Comput.* 28(4) (1999), pp. 1364–1396.

[30] R. Impagliazzo, L. Levin, M. Luby, Pseudo-random generation from one-way functions (extended abstract), *Proc 21st ACM Symp. on Theory of Computing*, 1989, pp. 12–24.

[31] R. Impagliazzo, Hard-core distributions for somewhat hard problems, *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995, pp. 538–545.

[32] R. Impagliazzo and M. Jakobsson, personal communication.

[33] J. Kilian, E. Petrank, and C. Rackoff, Lower bounds for zero knowledge on the internet, *Proc. 39th IEEE Symp. on Foundations of Computer Science*, 1998, pp. 484–492.

[34] S. Micali, C. Rackoff, and R. Sloan, The notion of security for probabilistic cryptosystems, *SIAM J. Computing* 17(2) (1988), pp. 412–426.

[35] M. Naor, Bit commitment using pseudorandomness, *J. of Cryptology* 4 (1991).

[36] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung, Perfect zero-knowledge arguments for NP using any one-way permutation, *J. of Cryptology* 11 (1998), pp. 87–108.

[37] M. Naor and A. Wool, Access control and signatures via quorum secret sharing, *IEEE Trans. on Parallel and Distributed Systems* 9 (1998), pp. 909–922. Preliminary version: *3rd ACM Conference of Computer and Communication Security*, 1996, pp. 157–168.

[38] Y. Oren, On the cunning power of cheating verifiers: some observations about zero knowledge proofs, *Proc. 28th IEEE Symp. on Foundations of Computer Science*, 1987, pp. 462–471.

[39] T. P. Pederson, Distributed provers with application to undeniable signatures, *Proc. EUROCRYPT'91*, Lecture Notes in Computer Science, vol. 547, Springer-Verlag, pp. 221–238.

[40] D. Pointcheval and J. Stern, Security proofs for signature schemes, *Proc. EUROCRYPT'96, Lecture Notes in Computer Science, vol. 1070*, Springer-Verlag, pp. 387–398.