On the (In)security of the Fiat-Shamir Paradigm

Shafi Goldwasser* Yael Taumann†

February 19, 2003

Abstract

In 1986, Fiat and Shamir suggested a general method for transforming secure 3-round public-coin identification schemes into digital signature schemes. The significant contribution of this method is a means for designing efficient digital signatures, while hopefully achieving security against chosen message attacks. All other known constructions which achieve such security are substantially more inefficient and complicated in design.

In 1996, Pointcheval and Stern proved that the signature schemes obtained by the Fiat-Shamir transformation are secure in the so called 'Random Oracle Model'. The question is: does the proof of the security of the Fiat-Shamir transformation in the Random Oracle Model, imply that the transformation yields secure signature schemes in the "real-world"?

In this paper we answer this question negatively. We show that there exist secure 3-round public-coin identification schemes for which the Fiat-Shamir methodology produces **insecure** digital signature schemes for **any** implementation of the 'Random Oracle Model' in the 'real-world' by a function ensemble.

*Department of Computer Science and Applied Math, The Weizmann Institute of Science at Rehovot, ISRAEL and The Department of Computer Science and Electrical Engineering at MIT. Email: shafi@theory.lcs.mit.edu

†Department of Computer Science and Applied Math, The Weizmann Institute of Science, Rehovot 76100, ISRAEL.

Contents

1 Introdcution

In their famous paper laying the foundations for modern cryptography, Diffie and Hellman [DH76] proposed the goal of designing secure *digital signatures.* They also proposed a general method for designing digital signatures. Their method uses trapdoor functions as its basic primitive and is known as the *trapdoor function signature method.*

Several drawbacks of the trapdoor function approach have surfaced. In terms of security, by its very definition, it is proned to *existential forgery* as defined in [GMR88]. In terms of efficiency, the time to sign and verify are proportional to the time to invert and compute the underlying trapdoor function – a cost which for some trapdoor functions is prohibitive for certain applications. Since the eighties, several signature schemes were proposed which were proved existentially unforgeable against chosen message attacks under a variety of complexity assumptions [GMR88, NY89, GHR99, CS99].

An entirely different method for designing digital signature schemes was proposed by Fiat and Shamir in 1986. They proposed a two step approach.

- First, design a "secure" 3-round public-coin identification scheme. That is, design a "secure" 3-round identification scheme (α, β, γ) where α, γ are prover's moves and β is a random string chosen by the verifier.

- Second, design a signature scheme as follows: let M be the message to be signed, then the signing algorithm consists of outputting an accepting transcript of the interactive identification protocol (α, β, γ), where $\beta = h(\alpha, M)$ and h a public function which is part of the signer's public-key. The intuition behind why such a signature scheme may be secure is that it would be hard for a forger to find a message M and a transcript (α, β, γ) for which it is true both that $\beta = h(\alpha, M)$ and that (α, β, γ) is an accepting transcript with respect to a public-key chosen by the real signer.

The resulting signature scheme is as efficient as the original identification scheme (which are generally more efficient than known signature schemes) and the cost of evaluating the public function h. Current proposals for a public (keyless) function h are very efficient [MD5].

Due to the efficiency and the ease of design, the Fiat-Shamir method shortly gained much popularity both in theory and in practice. Several digital signature schemes, of which the best known ones are [Sch91, GQ88, Ok92], were designed following this paradigm. The paradigm has also been applied in other domains such as to achieve forward secure digital signature schemes in [AABN02] and to achieve better exact security in [MR02]. Both of the above applications ([AABN02, MR02]) actually use a variation of the Fiat-Shamir paradigm. Still, they all share the same basic structure: start with some secure 3-round identification scheme and transform it into a digital signature scheme, eliminating the random move of the verifier by an application of a fixed function h to different quantities determined by the protocol and the public key.

The main question regarding any of these proposals is what can be proven about the security of the resulting signature schemes.

In 1996 Pointcheval and Stern [PS96] made a significant step toward answering this question. They proved that for every 3-round public-coin identification protocol, which is zero-knowledge with respect to an honest verifier, the signature scheme obtained by applying the Fiat-Shamir transformation is secure in the *Random Oracle Model.* This work was extended by Abdalla et al. [AABN02] to show necessary and sufficient conditions on 3-round identification protocols for which the signature scheme, obtained by applying the Fiat-shamir transformation, is secure in the Random Oracle Model.[1]

The Random Oracle Model is an *ideal* model which assumes that all parties (including the adversary) have oracle access to a truly random function. The so called *random oracle methodology* is a popular methodology that uses the Random Oracle Model for designing cryptographic schemes. It consists of the two steps. First, design a secure scheme in the Random Oracle Model. Then, replace the random oracle with a function, chosen at random from some function ensemble and provide all parties (including the adversary) with a succinct description of this function. Thus, obtain an *implementation* of the ideal scheme in the real world. This methodology introduced implicitly by [FS86], was formalized by Bellare and Rogaway [BR93].

As attractive as the methodology is for obtaining security "proofs", the obvious question was whether it is indeed always possible to replace the random oracle with a 'real world' implementation. This question was answered negatively by Canetti, Goldreich and Halevi [CGH98]. They showed that there exists a signature scheme and an encryption scheme which are secure in the Random Oracle Model but are insecure with respect to any implementation of the random oracle by a function ensemble. Thus, showing that the random oracle methodology fails 'in principle'.

The work of [CGH98] left open the possibility that for particular "natural" cryptographic practices, such as the Fiat-Shamir transformation, the random oracle methodolgy does work.

In this paper we show that this is not the case.

[1] The conditions are for the identification scheme to be secure against impersonation under passive attacks, and that the first message sent by the sender is drawn at random from a large space. [AABN02] show that the latter can be removed for a randomized version of the Fiat-Shamir transformation. For more details, see section 7.2.

1.1 Our Results

We prove that the Fiat-Shamir general paradigm for designing digital signatures can lead to universally forgeable digital signatures. We do so by demonstrating the existence of a secure 3-round public-coin identification scheme for which the corresponding signature scheme, obtained by applying the Fiat-Shamir transformation, is insecure with respect to any function ensemble implementing the public function.

Our result is *unconditional*, and does not depend on any intractability assumptions. Moreover, the problems we demonstrate for the Fiat-Shamir transformation apply to all other variations of the Fiat-Shamir transformation proposed in the literature [MR02, AABN02].

The central technique we employ is a new usage of Barak [Bar01]'s idea of taking advantage of non black-box access to the program of the verifier.

Intuitively, the idea is to take any secure 3-round public-coin identification scheme (which is not necessarily zero-knowledge) and extend its verdict function so that the receiver (verifier) also accepts views which convince him that the sender (prover) knows the receiver's next message. Since the receiver chooses the next message at random, there is no way that the sender can guess the receiver's next message during a real interaction, except with negligible probability, and therefore the scheme remains secure. However, when the identification scheme is converted into a signature scheme by applying the Fiat-Shamir transform, the receiver is replaced with a public function chosen at random from some function ensemble, which is known in advance to everyone. A forger who will now know in advance the receiver's next message on any input, will be able to generate an accepting view for the receiver. This makes the signature scheme insecure regardless of which function ensemble is used to replace the receiver in the identification scheme.

The main technical challenge with implementing this approach is the following: How can the sender convince the receiver that he knows the receiver's 'next message' using a 3-round protocol?

We make strong use of the non-interactive CS-proofs of Micali [Mi94] to overcome this challenge. However, non-interactive CS-proofs themselves are only known to hold in the Random Oracle Model and thus we *first* get the (somewhat odd-looking) conditional result that if CS-proofs are realizable in the 'real world' by some function ensemble, then there exist secure identification schemes for which the Fiat-Shamir transformation always fails in the 'real world' for all function ensembles. Next, we show that even if CS-proofs are not realized in the 'real world' by any function ensemble, the Fiat-Shamir paradigm is not secure. Perhaps, surprisingly, this part of the proof contains the bulk of difficulty and technical complication. This part again entails, showing different transformations on 3-round public-coin identification schemes which preserve their security when used as interactive identification schemes but make them completely insecure as signature schemes obtained by the the Fiat-Shamir transformation.

1.2 Related Work

Following the work of [CGH98], Dwork, Naor, Reingold and Stockmeyer [DNRS99] investigated the security of the Fiat-Shamir Paradigm and showed that it is closely related to previously studied problems: the *the selective decommitment problem*[2], and *the existence of 3-round public-coin weak zero-knowledge arguments for non BPP languages*. We note that the negative results presented in Section 1.1 regarding the insecurity of the Fiat-Shamir transformation have implications for these related problems.

In particular, the result of [DNRS99], that the existence of 3-round public-coin zero-knowledge protocols for non BPP languages implies the insecurity of the Fiat-Shamir paradigm, is worth elaborating on. It follows from the following simple observation. Suppose there exists a 3-round public-coin zero-knowledge argument for some hard language. View this zero-knowledge argument as a secure identification protocol[3]. The fact that the identification protocol is zero-knowledge (and not only honest verifier zero-knowledge) means that for **every verifier** there exists a simulator that can generate identical views to the ones produced during the run of the identification protocol. As the Fiat-Shamir transformation applied to this identification protocol, essentially fixes a public program for the verifier of the zero-knowledge argument, any forger can now simply run the simulator for this fixed verifier to produce a view of the identification protocol (i.e a valid digital signature).

This simple argument extends to any k-round public-coin zero-knowledge argument. Namely, if such a k-round public-coin zero-knowledge argument exists, it can be viewed as an identification protocol. Now, extend the original Fiat-Shamir transformation to an *Extended-Fiat-Shamir* transformation which replaces each message of the verifier (round at a time) by applying a fixed public function to previous messages in the protocol. Then the same argument as above says, that the simulator for the k-round zero-knowledge protocol can be used to produce forgeries in the signature scheme resulting from the Extended-Fiat-Shamir transformation and thus the Extended-Fiat-Shamir transformation fails.

When [DNRS99] pointed the above connection, no constant-round zero-knowledge public-coin protocol for non trivial languages was known. Since, Barak [Bar01] showed that under the assumption that collision resistant function ensembles exist, every language in NP has

[2]In 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 still hidden.

[3]It is not necessarily a proof of knowledge but it is certainly a proof of ability of proving membership in L which is hard for polynomial time impersonating algorithms

a constant-round (for some constant $k > 3$) public-coin zero-knowledge argument. Thus, it follows from [DNRS99] and [Bar01] (as above) that the Extended-Fiat-Shamir Paradigm is insecure.

The Fiat-Shamir paradigm was defined however, and has always been used only for 3-round identification schemes. Barak's work does not apply to this case. Moreover, his work implies that the Fiat-Shamir Paradigm (extended and otherwise) fails on zero-knowledge identification schemes (indeed it is the simulator for the zero-knowledge system which will produce forgeries), and left open the possibility that the (extended and ordinary) Fiat-Shamir paradigm works when the starting identification schemes are secure with respect to a less strict security requirement and are not zero-knowledge.

2 Preliminaries

Notations: We use [GMR88]'s notations and conventions for probabilistic algorithms.
If $\mathcal{A}$ is a probabilistic algorithm then for any input x we let $\mathcal{A}(x)$ refer to the probability space which assigns to any string σ the probability that $\mathcal{A}(x)$ outputs σ. If S is a probability space then $x \leftarrow S$ denotes the algorithm which assigns to x an element randomly selected according to S. For any probabilistic interactive Turing machines A and B, we let $(A, B)(x)$ refer to the transcript of their interaction on input x. At the end of the interaction B will always either accept or reject. We refer to this decision function of B as the verdict function of B. We abuse notion by saying that $(A, B)(x) = 1$ if B accepts. we denote by $VIEW(B(x))$ the set of all transcripts that $B(x)$ accepts. We denote by $A|_{\alpha}$, machine A, restricted to sending α as its first message. More generally, we denote by $A|_{\alpha_1,\ldots,\alpha_t}$, machine A, restricted to sending α_i as its i'th message, for $i = 1, \ldots, t$.

Definition 1. (Negligible): *We say that a function $g(\cdot)$ is negligible if for every polynomial $p(\cdot)$ there exists $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$*

$$g(n) < \frac{1}{p(n)}.$$

For any function $g(\cdot)$, we let $g(n) = negl(n)$ denote that $g(\cdot)$ is a negligible function.

Definition 2. (Non-negligible): *We say that a function $g(\cdot)$ is non-negligible if it is not negligible. That is, we say that $g(\cdot)$ is non-negligible if there exists a polynomial $p(\cdot)$ such that for infinitely many n's*

$$g(n) \geq \frac{1}{p(n)}.$$

For any function $g(\cdot)$, we let $g(n) = \textit{non-negl}(n)$ denote that $g(\cdot)$ is a non-negligible function.

Definition 3. (One-Way): *We say that a function ensemble $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is one-way if given a uniformly chosen $f \in_R \mathcal{F}_n$ and a uniformly chosen y in the image of f, it is hard to find x such that $f(x) = y$. That is, $\mathcal{F}$ is one-way if for every polynomial-size circuit $C = \{C_n\}_{n\in\mathbb{N}}$,*

$$Pr[C_n(f, y) = x : f(x) = y] = negl(n)$$

(where the probability is over uniformly chosen $f \in_R \mathcal{F}_n$ and $y \leftarrow f(U_n)$).

Throughout this paper we assume that one-way function ensembles exist. We stress that if one-way function ensembles do not exist then secure identification schemes and secure signature schemes do not exist, and thus the Fiat-Shamir Transform is trivially satisfied. The existence of one-way function ensembles implies the existence of secure identification schemes and secure signature schemes [NY89].

Definition 4. (Collision Resistence): *We say that a function ensemble $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is collision resistant if given a uniformly chosen $f \in_R \mathcal{F}_n$ it is hard to find x_1, x_2 such that $f(x_1) = f(x_2)$. That is, $\mathcal{F}$ is collision resistant if for every polynomial-size circuit $C = \{C_n\}_{n\in\mathbb{N}}$,*

$$Pr[C_n(f) = (x_1, x_2) : f(x_1) = f(x_2)] = negl(n)$$

(where the probability is over a uniformly chosen $f \in_R \mathcal{F}_n$).

Hypothesis (Collision Resistance Hypothesis): *There exists a collision resistance function ensemble $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ for which for every $n \in \mathbb{N}$,*

$$f_n : \{0,1\}^{2n} \rightarrow \{0,1\}^n.$$

We refer to this hypothesis as the CR hypothesis. Throughout the paper (excluding Section 3), we assume the CR hypothesis holds and we denote by $\mathcal{F}$ a collision resistance function ensemble given by this hypothesis.

Definition 5. (Commitment Scheme): *A commitment scheme is a function ensemble*

$$COMMIT = \{COMMIT_n\}_{n\in\mathbb{N}},$$

where

$$COMMIT_n = \{commit_k\}_{k\in KEY_n},$$

and there exist functions $l(n)$ and $t(n)$, which are polynomially-related to n, such that for every $n \in \mathbb{N}$ and every $k \in KEY_n$

$$commit_k : \{0,1\}^n \times \{0,1\}^{l(n)} \to \{0,1\}^{t(n)},$$

and the following properties are satisfied.

- *(Computationally-hiding): For every $n \in \mathbb{N}$, given any $k \in KEY_n$ and any $x \in \{0,1\}^n$,*

  $$commit_k(x; r) \cong U_{t(n)},$$

  assuming

  $$r \cong U_{l(n)}$$

  (where $\cong$ denotes computational-indistinguishability).

- *(Computationally-binding): For every $n \in \mathbb{N}$, given a random key $k \in_R KEY_n$ it is hard to find $(x_1, r_1) \neq (x_2, r_2)$ such that*

  $$commit_k(x_1; r_1) = commit_k(x_2; r_2).$$

  That is, for every polynomial-size circuit $C = \{C_n\}_{n \in \mathbb{N}}$

  $$Pr[C_n(k) = ((x_1, r_1), (x_2, r_2)) : commit_k(x_1; r_1) = commit_k(x_2; r_2)] = negl(n)$$

  (where the probability is over a uniformly chosen $k \in_R KEY_n$).

It was proven by Naor in [Na91] that commitment schemes exist, assuming the existence of one-way function ensembles.

For the purposes of this paper, we need a special commitment scheme, which we denote by $COMM = \{COMM_n\}_{n \in \mathbb{N}}$. For any polynomial $m(\cdot)$, $COMM$ is a commitment scheme that for every $n \in \mathbb{N}$ and for every $k \in KEY_n$,

$$COMM_k : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}^n.$$ [4]

In Appendix A we show that such a commitment scheme exists (for any polynomial $m(\cdot)$), under the CR hypothesis.

[4] Note that $COMM$ has the propety that the size of the randomness equals the size of the commitment. We need this property since in the sequel we use one commitment as randomness for another commitment.

2.1 Identification Schemes

Definition 6. (Identification Scheme): *An identification scheme (or ID scheme, for short) consists of a triplet* (G, S, R)*, where* G *is a key generation algorithm and* S *is the sender who wishes to prove his identity to the Receiver* R*. More formally,*

- G *is a probabilistic-polynomial-time Turing machine that, on input* 1^n*, outputs a pair* (SK, PK)*, such that the sizes of* SK *and* PK *are polynomially related to* n*. (*SK *is referred to as the secret-key and* PK *is referred to as the public-key).*

- S *and* R *are probabilistic-polynomial-time interactive Turing machines that are given a public-key* PK *as input. The sender* S *is also given a corresponding secret-key* SK*. It is required that for any pair* (SK, PK) *in the range of* $G(1^n)$*,*

$$Pr[(S(SK), R)(PK) = 1] = 1$$

(where the probability is over the random coin tosses of S *and* R*).*

In this paper we are interested in a special type of ID scheme, which we refer to as a canonical ID scheme.

Definition 7. (Canonical ID Scheme): *A canonical ID scheme is a* 3*-round ID scheme, in which the first message* α *is sent by the sender* S*, the second message* β *is sent by the receiver* R *and consists of* R*'s random coins, and the third message* γ *is sent by the sender* S*.*

For a sender S, with keys (SK, PK) and randomness r, we denote

- $\alpha = S_{(SK,PK)}(r)$
- $\gamma = S_{(SK,PK)}(\alpha, \beta; r)$.

2.1.1 Security of ID Schemes

As with any cryptographic primitive, the notion of security considers adversary goals (what it has to do to win) and adversary capability (what attacks it is allowed). Naturally, for an ID scheme, the adversary's goal is impersonation: it wins if it can interact with the receiver (in the role of a sender), and convince the latter to accept. There are two natural attacks to consider: passive and active. Passive attacks correspond to eavesdropping, meaning the adversary is in possession of transcripts of conversations between the real sender and the receiver. Active attacks means that it gets to play the role of a receiver, interacting with the real sender in an effort to extract information. We note that assuming the existence of one-way function ensembles, there exist ID schemes which are secure against active attacks.[5] Throughout this paper, security of an ID scheme should be interpreted as security against active attacks.

2.2 Signature Schemes

Definition 8. (Signature Scheme): *A signature scheme consists of a triplet*

$$(GEN, SIGN, VERIFY)$$

of probabilistic-polynomial-time Turing machines, where

- GEN*, on input* 1^n*, outputs a pair* (SK, VK)*, such that the sizes of* SK, VK *are polynomially related to* n*. (*SK *is referred to as the signing-key and* VK *is referred to as the verification-key).*
- $SIGN$ *gets as input a pair* (SK, VK) *and a message* M*, and outputs a signature of* M *with respect to* (SK, VK)*.*
- $VERIFY$ *gets as input a verification-key* VK*, a message* M *and a string* S *(which is supposedly a signature of* M *with respect to* VK*), and outputs* 0 *or* 1*.*

It is required that for any pair (SK, VK) *in the range of* $GEN(1^n)$ *and for any message* M*,*

$$Pr[VERIFY(VK, M, SIGN((SK, VK), M)) = 1] = 1$$

(where the probability is over the random coin tosses of $SIGN$ *and* $VERIFY$*).*

2.2.1 Security of Signature Schemes

Several types of security requirements were considered in the literature. In this paper we say that a signature scheme is secure if it is existentially secure against adaptive chosen message attacks.

Definition 9. (Security against adaptive chosen message attacks): *We say that a signature scheme* $SS = (GEN, SIGN, VERIFY)$ *is secure if for every polynomial-size circuit family*

[5] This is the case since the existence of one-way function ensembles imply the existence of secure signature schemes [NY89], which in turn imply the existence of ID schemes which are secure against active attacks (see Section 3).

$\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$, with oracle access to $SIGN$, the probability that, on input a uniformly chosen verification-key $VK \leftarrow GEN(1^n)$, $\mathcal{F}_n$ outputs a pair (M_0, SIG_{M_0}) such that

$$VERIFY(VK, M_0, SIG_{M_0}) = 1$$

and such that M_0 was not sent by $\mathcal{F}_n$ as an oracle query to $SIGN$, is negligible (where the probability is over VK and over the randomness of the oracle SIGN).

2.3 The Fiat-Shamir Transform

Definition 10. (The Fiat-Shamir Transform): *Given any canonical ID scheme (G, S, R) and any function ensemble $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$, the Fiat-Shamir transform transforms (G, S, R) and $\mathcal{H}$ into a signature scheme*

$$(GEN_{\mathcal{H}}, SIGN_{\mathcal{H}}, VERIFY_{\mathcal{H}}),$$

defined as follows.

- *The key generation algorithm $GEN_{\mathcal{H}}$, on input 1^n:*
  1. *Emulates algorithm G on input 1^n to generate $(SK, PK) \leftarrow G(1^n)$.*
  2. *Chooses at random a function $h^{FS} \in \mathcal{H}_n$.*

  Outputs SK as the signing-key and $VK = (PK, h^{FS})$ as the verification-key.

- *The signing algorithm $SIGN_{\mathcal{H}}$, on input a signing-key SK, a corresponding verification-key $VK = (PK, h^{FS})$, and a message M:*
  1. *Tosses coins r (for S).*
  2. *Computes $\alpha = S_{(SK,PK)}(r)$.*
  3. *Computes $\beta = h^{FS}(\alpha, M)$.*
  4. *Computes $\gamma = S_{(SK,PK)}(\alpha, \beta; r)$.*
  5. *Outputs (α, β, γ) as a signature of M.*

- *The verification algorithm $VERIFY_{\mathcal{H}}$, on input a verification-key $VK = (PK, h^{FS})$, a message M and a triplet (α, β, γ) (which is supposedly a signature of M), accepts if and only if both of the following conditions hold.*
  1. *$h^{FS}(\alpha, M) = \beta$.*
  2. *$(\alpha, \beta, \gamma) \in VIEW(R(PK))$.*

We denote by 'FS' the case that for every secure canonical ID scheme, there exists a function ensemble $\mathcal{H}$ such that the corresponding signature scheme (obtained by the Fiat-Shamir transform) is secure. We say that the Fiat-Shamir paradigm is secure if FS is true. Otherwise, we say that the Fiat-Shamir paradigm is insecure. We note that the Fiat-Shamir paradigm, of eliminating interaction by replacing the verifier with a function ensemble, has also been applied in other contexts, such as in the context of CS proofs [Mi94].

We begin by proving the insecurity of the Fiat-Shamir paradigm under the assumption that the CR hypothesis does not hold.

3 Proving the Insecurity of the Fiat-Shamir Paradigm, Assuming $\neg(CR)$

This section is dedicated for proving the following Lemma.

Lemma 3.1. $\neg(CR) \implies \neg(FS)$.

We will establish $\neg(FS)$ by transforming any secure signature scheme SS into a canonical ID scheme, denoted by ID.[6] Intuitively, the sender will identify himself by signing a random message sent by the receiver. The security of ID will follow from the security of SS. The insecurity of the corresponding signature scheme, obtained by applying the Fiat-Shamir transform to ID, will follow from the $\neg(CR)$ assumption.

Proof. Let $SS = (GEN, SIGN, VERIFY)$ be any secure signature scheme.[7] Consider the following ID scheme, $ID = (G, S, R)$.

- G: On input 1^n, emulate $GEN(1^n)$ to obtain a pair (SK, VK), and output SK as the secret-key and VK as the public-key.
- S and R are interactive Turing machines, that for any $(SK, VK) \leftarrow G(1^n)$, the interac-

[6]We note that in some sense this transformation is the inversion of the Fiat-Shamir transform, which converts any secure canonical ID scheme into a signature scheme.

[7]Recall that there exist secure signature schemes assuming the existence of one-way function ensembles [NY89].

tion of $(S(SK), R)(VK)$ is as follows.

$$\begin{array}{ccc} S(SK) & VK & R \\ & \xrightarrow[\emptyset]{\qquad\qquad} & \\ & \xleftarrow[x]{\qquad\qquad} & \\ & \xrightarrow[SIGN((SK,VK)(x))]{} & \end{array}$$

$R(VK)$ accepts a transcript (α, β, γ) if and only if the following two conditions are satisfied.

- $\alpha = \emptyset$
- $VERIFY(VK, \beta, \gamma) = 1$
  (i.e., γ is a valid signature of β, with respect to the verification-key VK).

Claim 3.1.1. *(G, S, R) is secure, assuming the signature scheme $(GEN, SIGN, VERIFY)$ is secure.*

Proof. Trivial! $\square$

We denote the corresponding signature scheme, with respect to the function ensemble $\mathcal{H}$, by

$$(GEN_{\mathcal{H}}, SIGN_{\mathcal{H}}, VERIFY_{\mathcal{H}}).$$

Claim 3.1.2. *Assuming $\neg(CR)$, for any function ensemble $\mathcal{H}$ the signature scheme*

$$(GEN_{\mathcal{H}}, SIGN_{\mathcal{H}}, VERIFY_{\mathcal{H}})$$

is insecure.

Proof. A forger, given a verification-key (VK, h^{FS}) and a signing oracle, will forge a signature to some new message M, as follows.

1. Find $M_1 \neq M_2$ such that $h^{FS}(M_1) = h^{FS}(M_2)$. From our assumption $\neg(CR)$, this can be done in probabilistic-polynomial-time[8].

[8] To be precise, we need to require that β is of size n and that the message to be signed is of size $2n$.

2. Query the signing oracle with the message M_1. The signature of M_1, obtained from the signing oracle, is of the form (α, β, γ) where
   - $\alpha = \emptyset$
   - $\beta = h^{FS}(M_1)$
   - $\gamma = SIGN((SK, VK), \beta)$.
3. Output (α, β, γ) as a signature to M_2.

(α, β, γ) is also a valid signature of M_2, assuming that both (α, β, γ) is a valid signature of M_1 and $h^{FS}(M_1) = h^{FS}(M_2)$. Since both of these conditions are satisfied with non-negligible probability, the forger succeeds in forging a signature of M_2 with non-negligible probability. $\square$

$\square$

We thus established

$$\neg(CR) \implies \neg(FS).$$

The rest of the paper is dedicated to proving

$$(CR) \implies \neg(FS).$$

Henceforth, we assume that the CR hypothesis holds.

4 Central Relation

In this section we define a relation that will be useful for the rest of the paper. Recall that our goal is to establish $\neg(FS)$ under the CR hypothesis. Our basic idea towards establishing this goal is the following: Start with any secure canonical ID scheme. Construct a new canonical ID scheme in which the receiver accepts either views that would have been accepted by the original receiver or views in which the sender convinces the receiver that he knows the receiver's 'next message'. That is, we extend the original verdict function so as to also accept views of the following form: In the first round the sender sends a which is a commitment to a circuit C (which is supposedly the 'next message' function of the receiver). Upon receiving a random message b from the receiver, the sender proves that the circuit C, which he committed to, predicts b. For various technical reasons to be elaborated on later, the type of commitment we use is *tree-commitment*. The notion of *tree-commitment* was introduces by [Mer90] and is defined as follows.

Definition 11. (Tree-Commitment): *A tree-commitment to x with respect to f is computed as follows. Consider a binary tree of depth* $\lg(|x|/n)$*, and label its leaves with the coordinates of x (each leaf is labeled with n coordinates). Label each non-leaf node by applying f to the label of its children. The tree-commitment to x with respect to f, is denoted by $TC_f(x)$, and consists of the label of the root and the depth of the tree.*

More specifically, we take any secure canonical ID scheme and extend its the verdict function so as to also accept views in which the sender, having sent a message a, which is supposedly a tree-commitment to a circuit C, and upon receiving a message b from the receiver, will prove that he knows a circuit C, such that both $TC_f(C) = a$ (for some given function f) and $C(a) = b$. That is, we extend the verdict function so as to also accept views in which the sender, having sent a message a and upon receiving a message b from the receiver, will send a proof that he knows a witness to the triplet (f, a, b) in the following relation, which was defined in [BG01].

Definition 12. (Central Relation):

$$\mathcal{R}_{\mathcal{F}} = \{((f,a,b),\hat{C}) \ : \ C(a) = b \ \wedge \ TC_f(\hat{C}) = a \ \wedge \ |\hat{C}| < n^{\lg n}\}$$

where $C \to \hat{C}$ is a special circuit-encoding which satisfies the following properties.

1. *It is an efficient encoding. That is, there is a polynomial-time algorithm that given any circuit C, outputs $\hat{C}$.*

2. *Given y, it is easy to check whether y is a codeword. That is, there is a polynomial-time algorithm that given y, outputs* 1 *if and only if there exists a circuit C such that $y = \hat{C}$.*

3. *There exists a polynomial-time algorithm that given any circuit-encoding $\hat{C}$ (where C is defined on inputs of size n) and given any $x \in \{0,1\}^n$, computes $C(x)$.*

4. *The circuit-encoding $C \to \hat{C}$ has high minimum distance. More precisely, for every $C_1 \neq C_2$, $\hat{C}_1$ and $\hat{C}_2$ differ in a polynomial fraction of their coordinates.*

Remarks:

1. We assume that the receiver's 'next message' function is of polynomial-size. We cannot bound this size by any fixed polynomial, and therefore we bound this size by some super-polynomial, such as $n^{\lg n}$.

2. We defined $\mathcal{R}_{\mathcal{F}}$ using a tree-commitment, as opposed to a regular commitment, for the following technical reason. In our proof we get a contradiction to the security of the Fiat-Shamir paradigm, by claiming knowledge of $\hat{C}_1 \neq \hat{C}_2$ which commit to the same value.

However, the size of these circuits is not a-priori bounded by some polynomial, and hence we will not be able to extract this knowledge using a polynomial-time algorithm. We get around this technical problem by using a tree-commitment, which allows one to decommit to individual bits.

Proposition 1. *[BG01]:*

$$L_{\mathcal{R}_{\mathcal{F}}} \in NTIME(n^{\lg n}).$$

Proof. Follows immediately from the definition of $\mathcal{R}_{\mathcal{F}}$ and from properties 2 and 3 of the circuit-encoding $C \to \hat{C}$. □

From the theory on Probabilistic-Checkable-Proofs it follows that there exists a polynomial-time Turing machine P_{PCP} and a probabilistic-polynomial-time oracle machine V_{PCP} with the following properties.

1. (Relatively-efficient oracle construction): for every $((f,a,b),\hat{C}) \in \mathcal{R}_{\mathcal{F}}$,
$$P_{PCP}((f,a,b),\hat{C}) = \pi$$
such that
$$Pr[V_{PCP}^{\pi}(f,a,b) = 1] = 1.$$

2. (Non-adaptive verifier:) The verifier's queries are determined based only on its input and on its internal coin tosses. That is, there exists a probabilistic-polynomial-time algorithm Q_{PCP} such that on input (f,a,b) and random coins r, the verifier makes the query sequence $\{q_i\}$, where for every i,
$$q_i = Q_{PCP}((f,a,b),r,i).$$

3. (Efficient reverse-sampling): There exists a probabilistic-polynomial-time oracle machine S such that, on input any string (f,a,b) and integers i and q, outputs a uniformly distributed r that satisfies
$$Q_{PCP}((f,a,b),r,i) = q.$$

4. (Proof-of-knowledge): There exists a probabilistic-polynomial-time oracle machine E and a negligible function $\epsilon(\cdot)$ such that, for every (f,a,b) and for every π, if
$$Pr[V_{PCP}^{\pi}(f,a,b) = 1] > \epsilon(|(f,a,b)|),$$
then there exists $\hat{C}$ such that $((f,a,b),\hat{C}) \in \mathcal{R}_{\mathcal{F}}$ and for every i,
$$Pr[E^{\pi}((f,a,b),i) = \hat{C}_i] \geq 2/3.$$

5 Interactive Arguments for $\mathcal{R}_\mathcal{F}$

In order to carry out the above idea towards establishing $\neg(FS)$, we need a proof-of-knowledge system for $\mathcal{R}_\mathcal{F}$. Moreover, since canonical ID schemes are confined to 3-rounds, we need a proof-of-knowledge system for $\mathcal{R}_\mathcal{F}$ which consists either of one round or of two rounds in which the verifier goes first. We begin by presenting a 4-round interactive argument for $\mathcal{R}_\mathcal{F}$ presented by Barak and Goldreich in [BG01]. We then do a series of modifications and obtain a reduced interaction version of their construction.

5.1 First Interactive Argument: (P^0, V^0)

We begin by reviewing the interactive argument for $\mathcal{R}_\mathcal{F}$, presented by Barak and Goldreich in [BG01]. The idea of such an argument goes back to [Ki92] and [Mi94]. We denote this interactive argument by (P^0, V^0):

- Common input: (f, a, b) (where $f \in \mathcal{F}_n$ and $a, b \in \{0,1\}^n$).
- Auxiliary input to the prover: $\hat{C}$ such that supposedly $((f, a, b), \hat{C}) \in \mathcal{R}_\mathcal{F}$.

1. V^0: Uniformly select $f^{UA} \in_R \mathcal{F}_n$ and send it to the prover.
2. P^0:
   (a) Construction of a PCP-proof: Invoke P_{PCP} on $((f, a, b), \hat{C})$ to obtain
   $$\pi = P_{PCP}((f, a, b), \hat{C}).$$
   (b) Tree-commitment to the PCP-proof: Compute
   $$\beta = TC_{f^{UA}}(\pi),$$ [9]
   which is the tree-commitment to π with respect to f^{UA}.
   (c) Send β to the prover.

[9] Note that there are two levels of use of the tree-commitment.

- In the definition of $\mathcal{R}_\mathcal{F}$: $TC_f(\hat{C}) = a$.
- In the interactive argument for $\mathcal{R}_\mathcal{F}$: $TC_{f^{UA}}(\pi) = \beta$.

In both cases we use a tree-commitment since the size of both $\hat{C}$ and π may be to large to extract. Using a tree-commitment we can extract only a few coordinates, with the ability to verify that these values were committed to.

3. V^0: Uniformly select a random-tape γ for V_{PCP}, and send γ to the prover.

4. P^0: Provide the answers to the (PCP) queries of $V_{PCP}((f, a, b); \gamma)$ augmented by proofs of consistency to these answers.

   (a) Determining the queries: Invoke $Q_{PCP}((f, a, b); \gamma)$, in order to determine the sequence of queries that V_{PCP} makes on input (f, a, b), given a random string γ.

   (b) For every query q_i of $Q_{PCP}((f, a, b); \gamma)$, send the label of the leaf that contains π_{q_i} and send the labels of the path corresponding to this leaf, which consists of the label of its sibling, the labels of its ancestors and the labels of its ancestors siblings, which are needed in order to verify consistency with β.

   We denote this response by $\delta = (label(\gamma), auth(\gamma))$.

V^0 accepts if and only if the following two conditions hold.

1. The answers provided by the prover would have been accepted by V_{PCP}.

2. All the proofs of consistency are valid.

(P^0, V^0), on input (f, a, b), can be schematically viewed as follows.

$$
\begin{array}{ccc}
P^0(\hat{C}) & & V^0 \\
 & \xleftarrow[f^{UA} \in \mathcal{F}_n]{\qquad\qquad} & \\
\begin{array}{c} \pi = P_{PCP}((f, a, b), \hat{C}) \\ \beta = TC_{f^{UA}}(\pi) \end{array} & \xrightarrow[\beta]{\qquad\qquad} & \\
 & \xleftarrow[\gamma]{\qquad\qquad} & \\
 & \xrightarrow[\delta = (label(\gamma), auth(\gamma))]{} &
\end{array}
$$

Lemma 5.1. *[Mi94],[BG01]: (P^0, V^0) satisfies the following properties.*

- *(Completeness): For every $((f, a, b), \hat{C}) \in \mathcal{R}_{\mathcal{F}}$,*

$$Pr[(P^0(\hat{C}), V^0)(f, a, b) = 1] = 1$$

  (where the probability is over the random coin tosses of V^0).

- *(CS-proof-of-knowledge): For every polynomial $p(\cdot)$, there exists a polynomial $p'(\cdot)$ and a probabilistic-polynomial-time oracle machine E such that for every polynomial-size circuit family $P^* = \{P_n^*\}$, for every sufficiently large n, and for every input (f, a, b), if*

$$Pr[(P_n^*, V^0)(f, a, b) = 1] \geq 1/p(n)$$

(where the probability is over the random coin tosses of V^0), then

$$Pr[\exists \hat{C} \text{ s.t. } ((f, a, b), \hat{C}) \in \mathcal{R}_{\mathcal{F}} \text{ and } \forall i \ E^{P_n^*}((f, a, b), i) = \hat{C}_i] \geq 1/p'(n)$$

(where the probability is over the random coin tosses of E).

We will not prove this Lemma since it was proved in [BG01] (using the four properties of (P_{PCP}, V_{PCP})). Moreover, following the proof in [BG01], it can be easily seen that the above proof-of-knowledge property holds even if P_n^* chooses (f, a, b) after receiving the verifier's first message f^{UA}.

5.2 Modified Interactive Argument: (P^1, V^1)

For reasons to be clarified later, we modify slightly the above interactive argument, by modifying the prover's first message from β to a commitment of β. Formally, we define a modified interactive argument, which we denote by (P^1, V^1), as follows.

- Common input: (f, a, b) (where $f \in \mathcal{F}_n$ and $a, b \in \{0, 1\}^n$).
- Auxiliary input to the prover: $\hat{C}$ such that supposedly $((f, a, b), \hat{C}) \in \mathcal{R}_{\mathcal{F}}$.

1. V^1: Uniformly select
   - $f^{UA} \in \mathcal{F}_n$ (a function for the tree-commitment)
   - $k \in KEY_n$ (a seed for $COMM$)
   - $r \in \{0, 1\}^n$ (randomness for $COMM$)

   Send $(f^{UA}, (k, r))$ to the prover.

2. P^1:
   (a) Construction of a PCP-proof: Invoke P_{PCP} on $((f, a, b), \hat{C})$ to obtain

$$\pi = P_{PCP}((f, a, b), \hat{C}).$$

(b) Tree-commitment to the PCP-proof: Compute

$$\beta = TC_{f^{UA}}(\pi),$$

(c) Send

$$\hat{\beta} = comm_k(\beta; r).$$

3. V^1: Uniformly select a random-tape γ for V_{PCP}, and send γ to the prover.

4. P^1: Send β, along with $\delta = (label(\gamma), auth(\gamma))$, which consists of the answers to the (PCP) queries of $V_{PCP}((f, a, b); \gamma)$ augmented by proofs of consistency to these answers.

V^1 accepts if and only if the following two conditions hold.

1. $\hat{\beta} = comm_k(\beta; r)$.

2. $(f^{UA}, \beta, \gamma, \delta) \in VIEW(V^0(f, a, b))$.

(P^1, V^1), on input (f, a, b), can be schematically viewed as follows.

$$\begin{array}{ccc}
P^1(\hat{C}) & & V^1 \\
 & \xleftarrow[f^{UA}, (k, r)]{\qquad\qquad} & \\
\begin{array}{c}\pi = P_{PCP}((f, a, b), \hat{C}) \\ \beta = TC_{f^{UA}}(\pi)\end{array} & \xrightarrow[\hat{\beta} = comm_k(\beta; r))]{\qquad\qquad} & \\
 & \xleftarrow[\gamma]{\qquad\qquad} & \\
\delta = (label(\gamma), auth(\gamma)) & \xrightarrow[\beta, \delta]{\qquad\qquad} &
\end{array}$$

Lemma 5.2. *(P^1, V^1) satisfies the following properties.*

- *(Completeness): For every $((f, a, b), \hat{C}) \in \mathcal{R}_{\mathcal{F}}$,*

$$Pr[(P^1(\hat{C}), V^1)(f, a, b) = 1] = 1$$

(where the probability is over the random coin tosses of V^1).

- *(CS-proof-of-knowledge): For every polynomial $p(\cdot)$, there exists a polynomial $p'(\cdot)$ and a probabilistic-polynomial-time oracle machine E such that for every polynomial-size circuit family $P^* = \{P_n^*\}$, for every sufficiently large n, and for every input (f, a, b), if*

$$Pr[(P_n^*, V^1)(f, a, b) = 1] \geq 1/p(n)$$

(where the probability is over the random coin tosses of V^1), then

$$Pr[\exists \hat{C} \text{ s.t. } ((f, a, b), \hat{C}) \in \mathcal{R}_{\mathcal{F}} \text{ and } \forall i \ E^{P_n^*}((f, a, b), i) = \hat{C}_i] \geq 1/p'(n)$$

(where the probability is over the random coin tosses of E).

As before, the above proof-of-knowledge property holds even if P_n^* chooses (f, a, b) after receiving the verifier's first message $(f^{UA}, (k, r))$.

5.3 Reduced-Interaction Argument: $(P^{\mathcal{H}}, V^{\mathcal{H}})$

As mentioned earlier, we would like to use an interactive argument for $\mathcal{R}_{\mathcal{F}}$, to construct a secure canonical ID scheme such that the corresponding signature scheme (obtained from the Fiat-Shamir transform) will be insecure with respect to any function ensemble. However, canonical ID schemes are confined to three rounds, and using the above interactive arguments we end up with an ID scheme with too many rounds. Thus, we would like to reduce the number of rounds in (P^1, V^1). We reduce the number of rounds by applying the Fiat-Shamir transform itself to (P^1, V^1) (i.e., by replacing V^1's second message with some function applied to P^1's first message).

For any function ensemble $\mathcal{H}$, we define a reduced-interaction argument $(P^{\mathcal{H}}, V^{\mathcal{H}})$ for $\mathcal{R}_{\mathcal{F}}$, with respect to $\mathcal{H}$, as follows.

- Common input: (f, a, b).
- Auxiliary input to the prover: $\hat{C}$ such that supposedly $((f, a, b), \hat{C}) \in \mathcal{R}_{\mathcal{F}}$.

1. $V^{\mathcal{H}}$: Uniformly select
    - $f^{UA} \in \mathcal{F}_n$ (a function for the tree-commitment)
    - $k \in KEY_n$ (a seed for $COMM$)
    - $r \in \{0, 1\}^n$ (randomness for $COMM$)
    - $h_1, \ldots, h_n \in \mathcal{H}_n$

Send $(f^{UA}, (k, r), (h_1, \ldots, h_n))$ to the prover.

2. $P^{\mathcal{H}}$:

   (a) Invoke P_{PCP} on $((f, a, b), \hat{C})$ to obtain $\pi = P_{PCP}((f, a, b), \hat{C})$.

   (b) Compute $\beta = TC_{f^{UA}}(\pi)$.

   (c) Compute $\hat{\beta} = comm_k(\beta; r)$.

   (d) For $i = 1, \ldots, n$,

      - compute $\gamma_i = h_i(\hat{\beta})$.
      - Let δ_i be the (PCP) answers corresponding to the queries $Q_{PCP}((f, a, b); \gamma_i)$ augmented by proofs of consistency to these answers.

   (e) send $(\hat{\beta}, \{\gamma_i\}_{i=1}^n, \beta, \{\delta_i\}_{i=1}^n)$.

$V^{\mathcal{H}}$ accept if and only if the following two conditions hold.

1. $\hat{\beta} = comm_k(\beta; r)$.

2. For $i = 1, \ldots, n$

   - $\gamma_i = h_i(\hat{\beta})$.
   - $(f^{UA}, \beta, \gamma_i, \delta_i) \in VIEW(V^0(f, a, b))$.

$(P^{\mathcal{H}}, V^{\mathcal{H}})$, on input (f, a, b), can be schematically viewed as follows.

$$
\begin{array}{ccc}
P^{\mathcal{H}}(\hat{C}) & & V^{\mathcal{H}} \\
 & \xleftarrow[f^{UA}, (k, r), (h_1, \ldots, h_n)]{} & \\
\begin{array}{c}
\pi = P_{PCP}((f, a, b), \hat{C}) \\
\beta = TC_{f^{UA}}(\pi) \\
\hat{\beta} = comm_k(\beta; r) \\
\gamma_i = h_i(\hat{\beta}) \\
\delta_i = (label(\gamma_i), auth(\gamma_i))
\end{array} & \xrightarrow[\hat{\beta}, \{\gamma_i\}, \beta, \{\delta_i\}]{} &
\end{array}
$$

Remarks on $(P^{\mathcal{H}}, V^{\mathcal{H}})$:

1. The reason that we require the prover to convince the verifier with n functions (rather than just one function) is to achieve error reduction.

2. We introduce some notation which will be useful later. Let q denote the message sent by $V^{\mathcal{H}}$, and let ans denote the response to q sent by $P^{\mathcal{H}}$. Recall that if $V^{\mathcal{H}}(f,a,b)$ accepts the view $(q;ans)$, then we say that $(q;ans) \in VIEW(V^{\mathcal{H}}(f,a,b))$.

5.3.1 $(P^{\mathcal{H}}, V^{\mathcal{H}})$ and CS-Proofs

The proof system $(P^{\mathcal{H}}, V^{\mathcal{H}})$ is closely related to CS-proofs, defined by Micali [Mi94]. Loosely speaking, CS-proofs are non-interactive proof systems for languages in $NEXP$. In CS proofs, Micali eliminated interaction from an interactive proof system for $NEXP$ (which is essentially (P^0, V^0)) by replacing the verifier with a random oracle. Micali proved that, in the Random Oracle Model, CS proofs satisfy both the completeness property and the CS-proof-of-knowledge property.[10] One can make the following hypothesis.

Hypothesis (CSP): *There exists a function ensemble $\mathcal{H}$ such that if the random oracle is replaced with a function uniformly chosen from $\mathcal{H}$, then CS-proofs still satisfy both the completeness property and the CS-proof-of-knowledge property.*

For every function ensemble $\mathcal{H}$, $(P^{\mathcal{H}}, V^{\mathcal{H}})$ satisfies the completeness requirement. However, we do not know if $(P^{\mathcal{H}}, V^{\mathcal{H}})$ satisfies the CS-proof-of-knowledge property. Looking carefully into the definition of CS-Proofs one can easily verify the following.

Proposition 2. *The CSP hypothesis implies that there exists a function ensemble $\mathcal{H}$ for which $(P^{\mathcal{H}}, V^{\mathcal{H}})$ satisfies both the completeness property and the CS-proof-of-knowledge property.*

Namely, if CS-proofs can be realized in the real world by some function ensemble $\mathcal{H}$, then so can $(P^{\mathcal{H}}, V^{\mathcal{H}})$.

6 Proving The Insecurity of the Fiat-Shamir Paradigm, Assuming (CR)

Our goal is to construct a secure canonical ID scheme such that the corresponding signature scheme, obtained from the Fiat-Shamir transform with respect to any function ensemble, will be insecure. Our first idea is the following. Take any secure canonical ID scheme and

[10] The definitions of completeness and of CS-proof-of-knowledge were given in Lemma 5.1 and Lemma 5.2.

extend its verdict function so as to also accept transcripts which convince the receiver that the sender knows the receiver's 'next message'. Since the receiver chooses the next message at random (follows from the definition of a canonical ID scheme), there is no way that a sender can guess the receiver's 'next message', except with negligible probability, and therefore the scheme remains secure. However, when the ID scheme is converted into a signature scheme by the Fiat-Shamir transform, the receiver is replaced with a public function from a function ensemble, and then everyone knows in advance the receiver's 'next message' on any input, and so can generate an accepting transcript, which corresponds to a legitimate signature. Hence, the corresponding signature scheme, with respect to any function ensemble, will be insecure.

The main problem with this approach is the following: How can the sender convince the receiver that he knows the receiver's 'next message'? One idea is to send the receiver a polynomial-size encoding of a circuit which computes the receiver's 'next message' function. However, the size of the interaction is bounded by an explicit polynomial, whereas the receiver's 'next message' circuit may be of any polynomial size. Therefore, we need to find a protocol of a-priori bounded size, in which the sender will be able to convince the receiver of knowledge of *any* polynomial-size circuit.

To achieve this goal, the sender, instead of sending an encoding to his circuit in hand (which may be too big), will send a commitment to his encoding. The type of commitment we use is a tree-commitment, which allows a fixed polynomial-size commitment for any polynomial-size circuit. Then, upon receiving a message from the receiver, the sender will convince the receiver that the circuit which he had committed to predicts this message. Recall that $\mathcal{R}_{\mathcal{F}}$ was designed exactly for this purpose. The sender will convince the receiver that he knows a circuit-encoding $\hat{C}$ which is a witness to the triplet (f, a, b), where a is the tree-commitment (with respect to f) sent by the sender and b is the message sent by the receiver. This will be done using the reduced-interaction argument $(P^{\mathcal{H}}, V^{\mathcal{H}})$ for $\mathcal{R}_{\mathcal{F}}$.

The FS Transform and CS Proofs

As we shall see shortly (in 6.1), if there exists a function ensemble $\mathcal{H}$ such that $(P^{\mathcal{H}}, V^{\mathcal{H}})$ satisfies the CS-proof-of-knowledge property, then the above approach works, and the insecurity of the Fiat-Shamir paradigm is easily established. Thus, from Proposition 2, we conclude that the CSP hypothesis implies $\neg(FS)$. This is quite surprising, since it essentially implies that if the FS transform applied to CS-proofs is secure, then the FS transform applied to canonical ID-schemes is not secure.

It turns out that the bulk of complication is in showing that if the CSP hypothesis is false then still $\neg(FS)$ is established. In other words, the bulk of complication is in proving that if the FS transform, applied to CS-proofs, is not secure then the FS transform, applied to canonical ID-schemes, is also not secure. This is also surprising since we expected this direction to be the easy one.

6.1 Construction of ID^1

We begin by carrying out the above idea. Let $\mathcal{F}$ be a collision resistance function ensemble. Let $\mathcal{H}$ be some a-priori fixed function ensemble. Let $ID = (G, S, R)$ be any secure canonical ID scheme. We extend the public-key and the verdict function of ID to obtain a new ID scheme $ID^1_{\mathcal{H}} = (G^1, S^1, R^1)$, defined as follows.

- G^1: on input 1^n,

  1. Run $G(1^n)$, to obtain a pair $(SK, PK) \leftarrow G(1^n)$.
  2. Choose $f \in_R \mathcal{F}_n$.

  Output SK as the secret-key and $PK' = (PK, f)$ as the public-key.

- R^1: On input a public-key $PK' = (PK, f)$, R^1 will accept either views that $R(PK)$ accepts or views of the form

$$\begin{array}{ccc} S^1 & & R^1 \\ & \xrightarrow[a]{\qquad} & \\ & \xleftarrow[b,q]{\qquad} & \\ & \xrightarrow[ans]{\qquad} & \end{array}$$

  such that $(q; ans) \in VIEW(V^{\mathcal{H}}(f, a, b))$.

To establish $\neg(FS)$, we need to show that $ID^1_{\mathcal{H}}$ is a secure ID scheme whereas the corresponding signature scheme (obtained from the Fiat-Shamir transform) is insecure with respect to any function ensemble. We begin by proving the insecurity of the corresponding signature scheme.

Let us denote the signature scheme, obtained by applying the Fiat-Shamir transform to $ID^1_{\mathcal{H}}$ and to $\mathcal{H}^{FS}$, by

$$SS^1_{\mathcal{H}^{FS}} = (GEN^1_{\mathcal{H}^{FS}}, SIGN^1_{\mathcal{H}^{FS}}, VERIFY^1_{\mathcal{H}^{FS}}).$$

6.1.1 On the Insecurity of $SS^1_{\mathcal{H}^{FS}}$

Lemma 6.1. *For any function ensemble $\mathcal{H}^{FS}$, the signature scheme $SS^1_{\mathcal{H}^{FS}}$ is insecure.*

Proof. We construct a forger that, on input any message M and any verification-key $VK = (PK', h^{FS})$ (where $PK' = (PK, f)$ and $h^{FS} \in \mathcal{H}_n^{FS}$) , generates a signature of M with respect to VK, as follows.

1. Let C be a circuit computing the hash function h^{FS}. Let C_M be a circuit such that for every x, $C_M(x) =$ n most-significant-bits of $C(x, M)$.
2. Compute $\hat{C}_M$.
3. Compute the tree-commitment $a = TC_f(\hat{C}_M)$.
4. Compute $(b, q) = C(a, M)$.
5. Emulate the interaction $(P^{\mathcal{H}}(\hat{C}_M), V^{\mathcal{H}}|_q)(f, a, b)$, to produce a transcript
$$(q, ans) \leftarrow (P^{\mathcal{H}}(\hat{C}_M), V^{\mathcal{H}}|_q)(f, a, b).^{11}$$
6. Output $(a, (b, q), ans)$.

It is trivial to verify that all forger steps are polynomial-time computable, and by completeness of $(P^{\mathcal{H}}, V^{\mathcal{H}})$, the forger will always be successful. ☐

6.1.2 On the Security of ID^1

To establish $\neg(FS)$ it remains to show that there exists a function ensemble $\mathcal{H}$, such that $ID^1_{\mathcal{H}}$ is secure. It is easy to prove the security of $ID^1_{\mathcal{H}}$ under the CSP hypothesis.

Lemma 6.2. *Under the CSP hypothesis, there exists a function ensemble $\mathcal{H}$ such that $ID^1_{\mathcal{H}}$ is secure.*

Proof. The CSP hypothesis implies that there exists a function ensemble $\mathcal{H}$ for which $(P^{\mathcal{H}}, V^{\mathcal{H}})$ satisfies both the completeness property and the CS-proof-of-knowledge property (follows from Proposition 2). It is easy to verify that $ID^1_{\mathcal{H}}$ is secure, with respect to this function ensemble $\mathcal{H}$. ☐

[11] Note that $((f, a, b), \hat{C}_M) \in \mathcal{R}_{\mathcal{F}}$.

Thus, we proved $(CSP) \implies \neg(FS)$.

Unfortunately, we do not know how to prove (directly) $\neg(CSP) \implies \neg(FS)$. Instead we proceed as follows. Consider the following two cases.

- (Case 1): There exists a function ensemble $\mathcal{H}$ such that $ID^1_{\mathcal{H}}$, is secure.
- (Case 2): For every function ensemble $\mathcal{H}$, $ID^1_{\mathcal{H}}$ is not secure.

If we are in Case 1 we are done, since then there exists a function ensemble $\mathcal{H}$ such that $ID^1_{\mathcal{H}}$ is secure, whereas the corresponding signature scheme is insecure with respect to any function ensemble, and $\neg(FS)$ is established. Hence, we assume that we are in Case 2. That is, we assume that for every function ensemble $\mathcal{H}$, there exists polynomial-size circuit family $F_1 = \{F_1^n\}$ (which we call a FINDER), a polynomial-size circuit family $\tilde{P}_1 = \{\tilde{P}_1^n\}$ (which corresponds to a cheating prover) and a polynomial $p(\cdot)$, such that for infinitely many n's,

$$Pr[(\tilde{P}_1^n, V^{\mathcal{H}})(f, a, b) = 1 \ : \ a = F_1^n(f)] \geq \frac{1}{p(n)}$$

(where the probability is over $f \in_R \mathcal{F}_n$, over $b \in_R \{0,1\}^n$ and over the random coin tosses of $V^{\mathcal{H}}$). We refer to this case by

$$'\forall \mathcal{H} \ \exists \ FINDER'.$$

We distinguish between two subcases.

- (Case $2a$): For every function ensemble $\mathcal{H}$, $ID^1_{\mathcal{H}}$ is 'extremely insecure'.
- (Case $2b$): For every function ensemble $\mathcal{H}$, $ID^1_{\mathcal{H}}$ is insecure and there exists a function ensemble $\mathcal{H}^1$ such that $ID^1_{\mathcal{H}^1}$ is not 'extremely insecure'.

We define Case $2a$ to be the case that for every function ensemble $\mathcal{H}$ there exists a polynomial-size circuit family $F_2 = \{F_2^n\}$ (called a SUPER-FINDER), a polynomial-size circuit family $\tilde{P}_2 = \{\tilde{P}_2^n\}$ (which corresponds to a cheating prover) and a polynomial $p(\cdot)$ such that for infinitely many n's,

$$Pr[(\tilde{P}_2^n, V^0)(f, a, b_1) = 1 \ \ \wedge \ (\tilde{P}_2^n, V^{\mathcal{H}})(f, a, b_2) = 1 : \ (a, b_1) = F_2^n(f)] \geq \frac{1}{p(n)}$$

(where the probability is over $f \in_R \mathcal{F}_n$, over $b_2 \in_R \{0,1\}^n$ and over the random coin tosses of $V^{\mathcal{H}}$ and V^0). We refer to sub-case $2(a)$ by

$$\forall \mathcal{H} \ \exists \ \text{SUPER-FINDER}$$

and we refer to sub-case $2(b)$ by

$$\neg(\forall \mathcal{H} \ \exists \ \text{SUPER-FINDER}).$$

6.2 Construnction of ID^2

Throughout this subsection we assume

$$(\forall \mathcal{H}\ \exists \text{ SUPER-FINDER}) \Rightarrow \neg(FS).$$

Fix a collision resistant function ensemble $\mathcal{F}$. We establish $\neg(FS)$ by extending any secure canonical ID scheme into a new ID scheme $ID^2 = (G^2, S^2, R^2)$. The security of ID^2 will follow from the fact that $\mathcal{F}$ is collision resistant. The insecurity of the corresponding signature scheme (obtained by the Fiat-Shamir transform applied to ID^2) will follow from the fact that for every function ensemble $\mathcal{H}$, $ID^1_{\mathcal{H}}$ is 'extremely insecure'.

Take a secure ID scheme $ID = (G, S, R)$, and define ID^2 as follows.

- G^2: On input 1^n,
  1. Run $G(1^n)$, to obtain a pair $(SK, PK) \leftarrow G(1^n)$.
  2. Choose uniformly
     - $f, f_1^{UA}, f_2^{UA} \in \mathcal{F}_n$
     - $k \in KEY_n$ (a seed for $COMM$)
     - $r \in \{0,1\}^n$ (randomness for $COMM$)
     - γ_1' (randomness for V_{PCP}).

  Output SK as the secret-key and $PK' = (PK, f, (f_1^{UA}, f_2^{UA}), (k, r), \gamma_1')$ as the public-key.

- R^2: On input a public-key $PK' = (PK, f, (f_1^{UA}, f_2^{UA}), (k, r), \gamma_1')$, R^2 will accept either views that $R(PK)$ will accept or views of the form

$$\begin{array}{ccc}
S^2 & & R^2 \\
 & \xrightarrow[\hat{\beta}_2]{\hspace{6em}} & \\
 & \xleftarrow[\gamma_1'', \gamma_2]{\hspace{6em}} & \\
 & \xrightarrow[a, b_1, b_2, \beta_1, \beta_2, \delta_1, \delta_2]{} &
\end{array}$$

  where

  - $(f_1^{UA}; \beta_1; \gamma_1' \oplus \gamma_1''; \delta_1) \in VIEW(V^0(f, a, b_1))$.

– $(f_2^{UA}; \beta_2; \gamma_2; \delta_2) \in VIEW(V^0(f, a, b_2))$.

– $\hat{\beta}_2$ commits to $a, b_1, b_2, \beta_1, \beta_2$, as follows

$$\hat{\beta}_2 = comm_k(\beta_2; comm_k(a, b_1, b_2, \beta_1; r)).$$

Intuitively, the above view can be thought of as an interleaved execution of the following two views:

$$\begin{array}{ccccccc}
P^0 & (f, a, b_1) & V^0 & \qquad & P^0 & (f, a, b_2) & V^0 \\
 & \xleftarrow[f_1^{UA}]{\qquad\qquad} & & & & \xleftarrow[f_2^{UA}]{\qquad\qquad} & \\
 & \xrightarrow[\beta_1]{\qquad\qquad} & & & & \xrightarrow[\beta_2]{\qquad\qquad} & \\
 & \xleftarrow[\gamma_1'' \oplus \gamma_1']{\qquad\qquad} & & & & \xleftarrow[\gamma_2]{\qquad\qquad} & \\
 & \xrightarrow[\delta_1]{\qquad\qquad} & & & & \xrightarrow[\delta_2]{\qquad\qquad} &
\end{array}$$

Remark: It is necessary to append γ_1' to the public-key in order to later establish the insecurity of the corresponding signature scheme. More specifically, when ID^2 will be converted into a signature scheme (by applying the Fiat-Shamir transform), the verifier will be replaced with a hash function, and thus γ_1'' will no longer necessarily be chosen at random. Yet, we only know how to establish the insecurity of the signature scheme assuming that γ_1'' is chosen at random. We get around this problem by XORing γ_1'' with a uniformly distributed string γ_1', from the public-key.

6.2.1 The Security of ID^2

Lemma 6.3. *Assuming $\mathcal{F}$ is collision resistant, ID^2 is secure.*

Proof. Assume for contradiction that ID^2 is not secure. That is, assume that there exists a cheating sender $\tilde{S} = \{\tilde{S}_n\}$ and a polynomial $p(\cdot)$ such that for infinitely many n's,

$$Pr[(\tilde{S}_n, R^2)(PK') = 1] \geq \frac{1}{p(n)}$$

(where the probability is over $PK' \leftarrow G^2(1^n)$ and over the random coin tosses of R^2).

Proof Plan: We will prove that the existence of $\tilde{S}$ implies the existence of a circuit that finds collisions in $\mathcal{F}$. This will be done in two parts, as follows.

- **(Part 1):** We will first show that there exist non-uniform probabilistic-polynomial-time Turing machines $F = \{F_n\}$ and $\tilde{P} = \{\tilde{P}_n\}$, such for infinitely many n's the following holds.

  For $(a, b_1, b_2, aux_1, aux_2) = F_n(f, f_1^{UA}, f_2^{UA})$,

  $$Pr\left[(\tilde{P}_n(aux_1), V^0|_{f_1^{UA}})(f, a, b_1) = 1 \ \wedge \ (\tilde{P}_n(aux_2), V^0|_{f_2^{UA}})(f, a, b_2) = 1\right] \geq 1/p(n)^3$$

  (where the probability is over a uniformly chosen $f, f_1^{UA}, f_2^{UA} \in \mathcal{F}_n$, and over the random coin tosses of F_n, $\tilde{P}_n$, $V^0|_{f_1^{UA}}$ and $V^0|_{f_2^{UA}}$).[12]

  The proof-of-knowledge property of (P^0, V^0) will imply that there exists a probabilistic-polynomial-time oracle machine E and a polynomial $p'(\cdot)$ such that for any $(a, b_1, b_2, aux_1, aux_2)$ which satisfy the above inequality,

  $$Pr\begin{bmatrix} \forall i \ E^{\tilde{P}_n(aux_1)}((f, a, b_1), i) = \hat{C}_i^1 \text{ s.t. } ((f, a, b_1), \hat{C}^1) \in \mathcal{R}_{\mathcal{F}} \\ and \\ \forall i \ E^{\tilde{P}_n(aux_2)}((f, a, b_2), i) = \hat{C}_i^2 \text{ s.t. } ((f, a, b_2), \hat{C}^2) \in \mathcal{R}_{\mathcal{F}} \end{bmatrix} \geq \frac{1}{p'(n)}$$

  (where the probability is over the random coin tosses of $E^{\tilde{P}_n(aux_1)}$ and $E^{\tilde{P}_n(aux_2)}$).

- **(Part 2):** We will then show that there exists a probabilistic-polynomial-time oracle machine, with oracle access to E, F_n and $\tilde{P}_n$, such that, on input a uniformly chosen $f \in_R \mathcal{F}_n$, outputs a collision in f, with non-negligible probability.

Note that since non-uniform probabilistic-polynomial-time Turing machines can be modeled as polynomial-size circuits, Part 1 together with Part 2 imply the existence of a polynomial-size circuit such that, on input a uniformly chosen $f \in_R \mathcal{F}_n$, outputs a collision in f, with non-negligible probability. This will contradict the assumption that $\mathcal{F}$ is collision resistant.

We proceed to carry out the proof plan.

Part 1:

- $F_n(f, f_1^{UA}, f_2^{UA})$ operates as follows.

  1. Choose uniformly
     - $PK \leftarrow G(1^n)$
     - $k \in KEY_n$ (a key for $COMM_n$)

[12]recall that $V^0|_{f^{UA}}$ is V^0, restricted to sending f^{UA} as the first message.

- $r \in \{0,1\}^n$ (randomness for $COMM_n$)
- γ_1' (randomness for V_{PCP})

and set $PK' = (PK, f, (f_1^{UA}, f_2^{UA}), (k, r), \gamma_1')$.

2. Emulate an interaction of $(\tilde{S}_n, R^2)(PK')$ to obtain a transcript

$$(\hat{\beta}_2; (\gamma_1'', \gamma_2); (a, b_1, b_2, \beta_1, \beta_2, \delta_1, \delta_2)) \leftarrow (\tilde{S}_n, R^2)(PK').$$

3. Set $aux_1 = (\beta_1, PK')$ and $aux_2 = (\beta_2, PK')$.

Output $(a, b_1, b_2, aux_1, aux_2)$.

- $\tilde{P}_n(aux_1)$, where $aux_1 = (\beta_1, PK')$, interacts with $V^0|_{f_1^{UA}}(f, a, b_1)$ as follows.
  - V^0 sends f_1^{UA} to $\tilde{P}_n$.
  - $\tilde{P}_n$ sends β_1 to V^0.
  - V^0 chooses γ_1^1 at random, and sends γ_1^1 to $\tilde{P}_n$.
  - $\tilde{P}_n$ chooses γ_2^1 at random and emulates the interaction of

$$(\tilde{S}_n|_{\beta_1}, R^2|_{\gamma_1^1 \oplus \gamma_1', \gamma_2^1})(PK'),$$

  to obtain a transcript

$$(\beta_1; (\gamma_1^1 \oplus \gamma_1', \gamma_2^1); (a', b_1', b_2', \beta_1', \beta_2', \delta_1', \delta_2')) \leftarrow (\tilde{S}_n|_{\beta_1}, R^2|_{\gamma_1^1 \oplus \gamma_1', \gamma_2^1})(PK').$$

  $\tilde{P}_n$ sends δ_1' to V^0.

- $\tilde{P}_n(aux_2)$, where $aux_2 = (\beta_2, PK')$, interacts with $V^0|_{f_2^{UA}}(f, a, b_2)$ as follows.
  - V^0 sends f_2^{UA} to $\tilde{P}_n$.
  - $\tilde{P}_n$ sends β_2 to V^0.
  - V^0 chooses γ_2^2 at random and sends γ_2^2 to $\tilde{P}_n$.
  - $\tilde{P}_n$ chooses γ_1^2 at random and emulates the interaction of

$$(\tilde{S}_n|_{\beta_2}, R^2|_{\gamma_1^2, \gamma_2^2})(PK')$$

  to obtain a transcript

$$(\beta_2; (\gamma_1^2, \gamma_2^2); (a'', b_1'', b_2'', \beta_1'', \beta_2'', \delta_1'', \delta_2'')) \leftarrow (\tilde{S}_n|_{\beta_2}, R^2|_{\gamma_1^2, \gamma_2^2})(PK').$$

  $\tilde{P}_n$ sends δ_1'' to V^0.

Claim 6.3.1. *Let* $F_n(f, f_1^{UA}, f_2^{UA}) = (a, b_1, b_2, aux_1, aux_2)$. *Then, for infinitely many* n*'s*

$$Pr\left[(\tilde{P}_n(aux_1), V^0|_{f_1^{UA}})(f, a, b_1) = 1 \ \wedge (\tilde{P}_n(aux_2), V^0|_{f_2^{UA}})(f, a, b_2) = 1\right] \geq 1/p(n)^3$$

(where the probability is over $f, f_1^{UA}, f_2^{UA} \in_R \mathcal{F}_n$*, and over the random coin tosses of* $V^0|_{f_1^{UA}}$ *and* $V^0|_{f_2^{UA}}$*).*

Proof. By the assumption made for contradiction, for infinitely many n's

$$Pr[(\tilde{S}_n, R^2)(PK') = 1] \geq 1/p(n)$$

(where the probability is over PK' and over the random coin tosses of R^2).
The fact that $\gamma_1'', \gamma_2, \gamma_1^1 \oplus \gamma_1', \gamma_2^1, \gamma_1^2, \gamma_2^2$ are all uniformly distributed and independent of PK', implies that for infinitely many n's, the following three conditions hold with probability at least $1/p(n)^3$.

- $(\tilde{S}_n, R^2|_{\gamma_1'', \gamma_2})(PK') = 1$
- $(\tilde{S}_n, R^2|_{\gamma_1^1 \oplus \gamma_1', \gamma_2^1})(PK') = 1$
- $(\tilde{S}_n, R^2|_{\gamma_1^2, \gamma_2^2})(PK') = 1$

In other words,

- $(\hat{\beta}_2; (\gamma_1'', \gamma_2); (a, b_1, b_2, \beta_1, \beta_2, \delta_1, \delta_2)) \in VIEW(R^2|_{\gamma_1'', \gamma_2})(PK')$
- $(\hat{\beta}_2; (\gamma_1^1 \oplus \gamma_1', \gamma_2^1); (a', b_1', b_2', \beta_1', \beta_2', \delta_1', \delta_2')) \in VIEW(R^2|_{\gamma_1^1 \oplus \gamma_1', \gamma_2^1})(PK')$
- $(\hat{\beta}_2; (\gamma_1^2, \gamma_2^2); (a'', b_1'', b_2'', \beta_1'', \beta_2'', \delta_1'', \delta_2'')) \in VIEW(R^2|_{\gamma_1^2, \gamma_2^2})(PK')$.

Equivalently, all the following conditions hold.

- 1. $\hat{\beta}_2 = comm_k(\beta_2; comm_k(a, b_1, b_2, \beta_1; r))$
  2. $(f_1^{UA}; \beta_1; \gamma_1'' \oplus \gamma_1'; \delta_1) \in VIEW(V^0(f, a, b_1))$
  3. $(f_2^{UA}; \beta; \gamma_2; \delta_2) \in VIEW(V^0(f, a, b_2))$.
- 1. $\hat{\beta}_2 = comm_k(\beta_2'; comm_k(a', b_1', b_2', \beta_1'; r))$
  2. $(f_1^{UA}; \beta_1'; (\gamma_1^1 \oplus \gamma_1') \oplus \gamma_1'; \delta_1') \in VIEW(V^0(f, a, b_1))$
  3. $(f_2^{UA}; \beta_2'; \gamma_2^1; \delta_2') \in VIEW(V^0(f, a, b_2))$.

- 1. $\hat{\beta}_2 = comm_k(\beta_2''; comm_k(a'', b_1'', b_2'', \beta_1''; r)$
  2. $(f_1^{UA}; \beta_1''; \gamma_1^2 \oplus \gamma_1'; \delta_1'') \in VIEW(V^0(f, a, b_1))$
  3. $(f_2^{UA}; \beta_2''; \gamma_2^2; \delta_2'') \in VIEW(V^0(f, a, b_2))$.

Since $comm_k$ is computationally-binding and $\tilde{S}_n$ is of polynomial-size, conditions (1) imply that

$$(a, b_1, b_2, \beta_1, \beta_2) = (a', b_1', b_2', \beta_1', \beta_2') = (a'', b_1'', b_2'', \beta_1'', \beta_2'').$$

The above equality combined with conditions (2) and (3) imply that

1. $(f_1^{UA}; \beta_1; \gamma_1^1; \delta_1') \in VIEW(V^0(f, a, b_1))$
2. $(f_2^{UA}; \beta_2; \gamma_2^2; \delta_2'') \in VIEW(V^0(f, a, b_2))$.

□

The proof-of-knowledge property of (P^0, V^0) implies that there exists a probabilistic-polynomial-time oracle machine E and a polynomial $p'(\cdot)$ such that for infinitely many n's, for $(a, b_1, b_2, aux_1, aux_2) = F_n(f, f_1^{UA}, f_2^{UA})$,

$$Pr\left[\begin{array}{c} \forall i\ E^{\tilde{P}_n(aux_1)}((f, a, b_1), i) = \hat{C}_i^1 \text{ s.t. } ((f, a, b_1), \hat{C}^1) \in \mathcal{R}_{\mathcal{F}} \\ and \\ \forall i\ E^{\tilde{P}_n(aux_2)}((f, a, b_2), i) = \hat{C}_i^2 \text{ s.t. } ((f, a, b_2), \hat{C}^2) \in \mathcal{R}_{\mathcal{F}} \end{array}\right] \geq \frac{1}{p'(n)}$$

(where the probability is over uniformly chosen $f, f_1^{UA}, f_2^{UA} \in \mathcal{F}_n$ and over the random coin tosses of F_n, $E^{\tilde{P}_n(aux_1)}$ and $E^{\tilde{P}_n(aux_2)}$).

Part 2: We next show how one can use E and F_n and $\tilde{P}_n$ to find a collision in $\mathcal{F}$. We define a probabilistic-polynomial-time oracle machine $\mathcal{M}$, which is given oracle access to E, F_n and $\tilde{P}_n$, and such that on input a random function $f \in \mathcal{F}_n$ outputs a collision in f, with non-negligible probability.

$\mathcal{M}^{E, F_n, \tilde{P}_n}$, on input $f \in \mathcal{F}_n$, operates as follows.

1. Choose $f_1^{UA}, f_2^{UA} \in_R \mathcal{F}_n$ and run $F_n(f, f_1^{UA}, f_2^{UA})$ to obtain

$$(a, b_1, b_2, aux_1, aux_2) \leftarrow F_n(f, f_1^{UA}, f_2^{UA}).$$

2. Choose a random i, and compute
   (a) $\hat{C}_i^1 = E^{\tilde{P}_n(aux_1)}((f, a, b_1), i)$

(b) $\hat{C}_i^2 = E^{\tilde{P}_n(aux_2)}((f, a, b_2), i)$.

3. Use the efficient reverse-sampling property of (P_{PCP}, V_{PCP}), to find random γ_1 and γ_2 such that i belongs to the set of queries of $Q_{PCP}((f, a, b_1), \gamma_1)$, and to the set of queries of $Q_{PCP}((f, a, b_2), \gamma_2)$.

4. Emulate the interaction of $(\tilde{P}_n(aux_1), V^0|_{f_1^{UA}, \gamma_1})(f, a, b_1)$ to get the labels of the path of $\hat{C}_i^1$, and emulate the interaction of $(\tilde{P}_n(aux_2), V^0|_{f_2^{UA}, \gamma_2})(f, a, b_2)$ to get the labels of the path of $\hat{C}_i^2$.

Claim 6.3.2. *With non-negligible probability (over $f \in_R \mathcal{F}_n$ and over the random coin tosses of $\mathcal{M}$, E, F_n, and $\tilde{P}_n$) somewhere along these paths there will be a collision in f.*

Proof. With non-negligible probability (over the random coin tosses of $\mathcal{M}$, E, F_n, and $\tilde{P}_n$), $\hat{C}_i^1$ is the i'th bit of $\hat{C}^1$ and $\hat{C}_i^2$ is the i'th bit of $\hat{C}^2$, where

$$((f, a, b_1), \hat{C}^1), ((f, a, b_2), \hat{C}^2) \in \mathcal{R}_{\mathcal{F}}.$$

Since $\hat{C}_1 \neq \hat{C}_2$ and since the circuit-encoding $C \rightarrow \hat{C}$ has large minimum distance, it follows that with probability $\frac{1}{poly}$ the following inequality holds

$$\hat{C}_i^1 \neq \hat{C}_i^2$$

(where *poly* is a polynomial and the probability is over a random chosen i).
This implies that somewhere along these paths there will be a collision to f, since

$$\hat{C}_i^1 \neq \hat{C}_i^2$$

and yet

$$a = TC_f(\hat{C}^1) = TC_f(\hat{C}^2).$$

□

This Contradicts our assumption that $\mathcal{F}$ is a collision resistance function ensemble. □

We thus established the security of ID^2. We denote the signature scheme, obtained by applying the Fiat-Shamir transform to ID^2 and the function ensemble $\mathcal{H}$, by

$$SS_{\mathcal{H}}^2 = (GEN_{\mathcal{H}}^1, SIGN_{\mathcal{H}}^2, VERIFY_{\mathcal{H}}^2).$$

6.2.2 The Insecurity of $SS^2_{\mathcal{H}}$

Lemma 6.4. *Assuming '$\forall \mathcal{H}\ \exists$ SUPER-FINDER', for any function ensemble $\mathcal{H}$, the signature scheme $SS^2_{\mathcal{H}}$ is insecure.*

Proof. Fix a function ensemble $\mathcal{H}$. We show that for every message M there exists a forger $FORG^M$ which, on input a random verification-key VK, outputs a signature of M, with non-negligible probability.

Fix any message M. For any $h \in \mathcal{H}$, define

$$h^M(x) = \ n \text{ least-significant-bits of } h(x, M),$$

and let $\mathcal{H}^M = \{h^M\}_{h\in\mathcal{H}}$. Let $F_2 = \{F_2^n\}_{n\in\mathbb{N}}$ be a SUPER-FINDER for $\mathcal{H}^M$ and let $\tilde{P}_2 = \{\tilde{P}_2^n\}$ be the corresponding cheating prover such that for $(a, b_1) = F_2^n(f)$,

$$Pr[(\tilde{P}_2^n, V^0)(f, a, b_1) = 1\ \wedge\ (\tilde{P}_2^n, V^{\mathcal{H}^M})(f, a, b_2) = 1] = \textit{non-negl}(n)$$

(where the probability is over $f \in_R \mathcal{F}_n$, $b_2 \in_R \{0,1\}^n$ and the random coin tosses of V^0 and $V^{\mathcal{H}^M}$).

On input a random verification-key $VK = (PK', h)$, where $h \in \mathcal{H}_n$ and

$$PK' = (PK, f, (f_1^{UA}, f_2^{UA}), (k, r), \gamma_1'),$$

the forger $FORG^M$ generates a signature of M as follows.

1. Compute $(a, b_1) = F_2^n(f)$.
2. Emulate the interaction of $(\tilde{P}_2^n, V^0|_{f_1^{UA}})(f, a, b_1)$, to obtain a transcript
$$(f_1^{UA}; \beta_1; *; *) \leftarrow (\tilde{P}_2^n, V^0|_{f_1^{UA}})(f, a, b_1).$$
3. Choose randomly $b_2 \in \{0,1\}^n$, and let $r' = \textit{comm}_k(a, b_1, b_2, \beta_1; r)$.
4. Choose randomly $h_2 \ldots, h_n \in \mathcal{H}_n$, and let
$$q_M = (f_2^{UA}, (k, r'), (h^M, h_2^M \ldots, h_n^M)).$$
5. Emulate the interaction of $(\tilde{P}_2^n, V^{\mathcal{H}^M}|_{q_M})(f, a, b_2)$, to obtain a transcript
$$(q_M; ans) \leftarrow (\tilde{P}_2^n, V^{\mathcal{H}^M}|_{q_M})(f, a, b_2).$$

   Denote $\quad ans = (\hat{\beta}_2, (\gamma_2, \gamma_2^2, \ldots \gamma_2^n), \beta_2, (\delta_2, \delta_2^2, \ldots, \delta_2^n)).$

6. Compute $(\gamma_1'', *) = h(\hat{\beta}_2, M)$.

7. Emulate the interaction
$$(\tilde{P}_2^n, V^0|_{f_1^{UA}, \gamma_1' \oplus \gamma_1''})(f, a, b_1)$$
to obtain a transcript
$$(f_1^{UA}; \beta_1; \gamma_1' \oplus \gamma_1''; \delta_1) \leftarrow (\tilde{P}_2^n, V^0|_{f_1^{UA}, \gamma_1' \oplus \gamma_1''})(f, a, b_1).$$

8. Output
$$(\hat{\beta}_2; (\gamma_1'', \gamma_2); (a, b_1, b_2, \beta_1, \beta_2, \delta_1, \delta_2))$$
as a signature of M.

We claim that the forger will be successful with non-negligible probability.

Claim 6.4.1.

$$Pr[VERIFY_{\mathcal{H}}^2(VK, M, FORG^M(VK)) = 1] = \textit{non-negl}(n)$$

(where the probability is over VK and over the random coin tosses of $FORG^M$).

Proof. Denote the output of $FORG^M(VK)$ by $(\hat{\beta}_2; (\gamma_1'', \gamma_2); (a, b_1, b_2, \beta_1, \beta_2, \delta_1, \delta_2))$.
By the definition of $\tilde{P}_2^n$, for $(a, b_1) = F_2^n$,

$$Pr[(\tilde{P}_2^n, V^0)(f, a, b_1) = 1 \ \wedge \ (\tilde{P}_2^n, V^{\mathcal{H}^M})(f, a, b_2) = 1] = \textit{non-negl}(n) \qquad (1)$$

(where the probability is over $f \in_R \mathcal{F}_n$, $b_2 \in_R \{0,1\}^n$ and the random coin tosses of V^0 and $V^{\mathcal{H}^M}$).
We claim that similarly, for $(a, b_1) = F_2^n$,

$$Pr[(\tilde{P}_2^n, V^0|_{f_1^{UA}, \gamma_1'' \oplus \gamma_1'})(f, a, b_1) = 1 \ \wedge \ (\tilde{P}_2^n, V^{\mathcal{H}^M}|_{q_M})(f, a, b_2) = 1] = \textit{non-negl}(n) \qquad (2)$$

(where the probability is over $f \in_R \mathcal{F}_n$, $b_2 \in_R \{0,1\}^n$, and over f_1^{UA}, $\gamma_1'' \oplus \gamma_1'$ and q_M).
This is so for the following reasons

1. f_1^{UA} was chosen uniformly in $\mathcal{F}_n$

2. $\gamma_1'' \oplus \gamma_1'$ was chosen uniformly (follows from the fact that γ_1' was chosen uniformly and γ_1'' was chosen independently of γ_1').

3. $\tilde{P}_2^n$ (in step 7) cannot distinguish between the distribution of q_M and the distribution of a random query of $V^{\mathcal{H}^M}$.

For all of the above reasons, $\tilde{P}_n^2$ in (2) should succeed with essentially the same probability as in (1).

The fact that $(\tilde{P}_2^n, V^0|_{f_1^{UA}, \gamma_1'' \oplus \gamma_1'})(f, a, b_1) = 1$ implies that

- $(f_1^{UA}; \beta_1; \gamma_1'' \oplus \gamma_1'; \delta_1) \in VIEW(V^0(f, a, b_1))$.

The fact that $(\tilde{P}_2^n, V^{\mathcal{H}^M}|_{q_M})(f, a, b_2) = 1$ implies that $(q_M; ans) \in VIEW(V^{\mathcal{H}^M}(f, a, b_2))$, which in turn implies that both of the following conditions hold.

- $(f_2^{UA}; \beta_2; \gamma_2; \delta_2) \in VIEW(V^0(f, a, b_2))$
- $(\gamma_1'', \gamma_2) = h(\hat{\beta}_2, M))$.

The satisfaction of above three conditions imply that the forgery was successful. $\square$

$\square$

6.3 Construction of ID^3

Throughout this subsection we assume

$$(\forall \mathcal{H} \ \exists \ \text{FINDER}) \ \wedge \ \neg(\forall \mathcal{H} \ \exists \ \text{SUPER-FINDER}) \ \Rightarrow \ \neg(\text{FS})$$

We assume that for every function ensemble $\mathcal{H}$, $ID^1_{\mathcal{H}}$ is insecure, and that there exists a function ensemble $\mathcal{H}^1$ such that $ID^1_{\mathcal{H}^1}$ is not 'extremely insecure'. We establish $\neg(FS)$ by extending any secure ID scheme into a new ID scheme $ID^3 = (G^3, S^3, R^3)$. The security of ID^3 follows from the fact that $ID^1_{\mathcal{H}^1}$ is not 'extremely insecure'. The insecurity of the corresponding signature scheme (obtained by applying the Fiat-Shamir transform to ID^3) follows from the fact that for every function ensemble $\mathcal{H}$, $ID^1_{\mathcal{H}}$ is insecure.

Take any secure canonical ID scheme $ID = (G, S, R)$ and the function ensemble $\mathcal{H}^1$, and define ID^3 as follows.

- G^3: On input 1^n,
  1. Run $G(1^n)$, to obtain a pair $(SK, PK) \leftarrow G(1^n)$.
  2. Choose uniformly
     - $f, f^{UA} \in \mathcal{F}_n$
     - $k \in KEY_n$ (a key for $COMM$)
     - $r \in \{0,1\}^n$ (randomness for $COMM$)

– $b_2' \in \{0,1\}^n$

– q_1' (a first message sent by $V^{\mathcal{H}^1}$).

Output SK as the secret-key and

$$PK' = (PK, f, f^{UA}, (k,r), (b_2', q_1'))$$

as the public-key.

- R^3: On input a public-key $PK' = (PK, f, f^{UA}, (k,r), (b_2', q'))$, R^3 accepts either views that $R(PK)$ accepts or views of the form

$$\begin{array}{ccc} S^3 & & R^3 \\ & \xrightarrow[\hat{\beta}_1]{\hspace{3cm}} & \\ & \xleftarrow[\gamma_1, (b_2'', q'')]{} & \\ & \xrightarrow[a, b_1, \beta_1, \delta_1, ans]{} & \end{array}$$

where

– $(f^{UA}; \beta_1; \gamma_1; \delta_1) \in VIEW(V^0(f, a, b_1))$

– $(q' \oplus q''; ans) \in VIEW(V^{\mathcal{H}^1}(f, a, b_2' \oplus b_2''))$

– $\hat{\beta}_1$ commits to a, b_1, β_1, as follows

$$\hat{\beta}_1 = comm_k(\beta_1; comm_k(a, b_1; r)).$$

Intuitively, the above view can be thought of as an interleaved execution of the following two views:

$$\begin{array}{ccc} P^0 & (f, a, b_1) & V^0 \\ & \xleftarrow[f^{UA}]{} & \\ & \xrightarrow[\beta_1]{} & \\ & \xleftarrow[\gamma_1]{} & \\ & \xrightarrow[\delta_1]{} & \end{array} \qquad \begin{array}{ccc} P^{\mathcal{H}^1} & (f, a, b_2' \oplus b_2'') & V^{\mathcal{H}^1} \\ & \xleftarrow[q' \oplus q'']{} & \\ & \xrightarrow[ans]{} & \end{array}$$

Remark: It is necessary to append b_2', q' to the public-key in order to later establish the insecurity of the corresponding signature scheme. More specifically, when ID^3 will be converted into a signature scheme (by applying the Fiat-Shamir transform), the verifier will be replaced with a hash function, and thus b_2'' and q'' will no longer necessarily be chosen at random. Yet, we only know how to establish the insecurity of the signature scheme assuming that b_2'' and q'' are chosen at random. We get around this problem by XORing b_2'' with a uniformly distributed string b_1' and XORing q'' with a uniformly distributed string q'.

Lemma 6.5. *Assuming $\mathcal{H}^1$ does not have a SUPER-FINDER, ID^3 is secure.*

Proof. Follows easily from the definition of a SUPER-FINDER. □

We denote the signature scheme, obtained by applying the Fiat-Shamir transform to ID^3 and the function ensemble $\mathcal{H}$, by

$$SS^3_{\mathcal{H}} = (GEN^3_{\mathcal{H}}, SIGN^3_{\mathcal{H}}, VERIFY^3_{\mathcal{H}}).$$

Lemma 6.6. *Assuming '$\forall \mathcal{H}\ \exists$ FINDER', for any function ensemble $\mathcal{H}$, the signature scheme $SS^3_{\mathcal{H}}$ is insecure.*

Proof. Fix a function ensemble $\mathcal{H}^{FS}$. We exhibit a forger for $SS^3_{\mathcal{H}^{FS}}$. More specifically, we show that for every message M there exists a forger $FORG^M$ which, on input a random verification key VK, outputs a signature of M, with non-negligible probability.

Fix any message M. For any $h \in \mathcal{H}^{FS}_n$, define

$$h^M(x) = \ n \text{ most-significant-bits of } h(x, M).$$

Let $\mathcal{H}^M = \{h^M\}_{h \in \mathcal{H}^{FS}}$, and let $\mathcal{H} = \mathcal{H}^1 \cup \mathcal{H}^M$. By our assumption '$\forall \mathcal{H}\ \exists FINDER$', there exist $F_1 = \{F_1^n\}_{n \in \mathbb{N}}$ and $\tilde{P}_1 = \{\tilde{P}_1^n\}$, such that for $a = F_1^n(f)$,

$$Pr[(\tilde{P}_1^n, V^{\mathcal{H}})(f, a, b) = 1] = \textit{non-negl}(n)$$

(where the probability is over $f \in_R \mathcal{F}_n$, $b \in_R \{0,1\}^n$ and the random coin tosses of $V^{\mathcal{H}}$).

It is easy to see that the existence of $\tilde{P}_1^n$ implies the existence of a polynomial-size circuit $\tilde{\tilde{P}}_1^n$ such that for $a = F_1^n(f)$,

$$Pr[(\tilde{\tilde{P}}_1^n, V^{\mathcal{H}^1})(f, a, b_1) = 1\ \wedge\ (\tilde{\tilde{P}}_1^n, V^{\mathcal{H}^M})(f, a, b_2) = 1] = \textit{non-negl}(n)$$

(where the probability is over $f \in_R \mathcal{F}_n$, $b_1, b_2 \in_R \{0,1\}^n$ and the random coin tosses of $V^{\mathcal{H}^1}$ and $V^{\mathcal{H}^M}$).

We are now ready to exhibit the forger $FORG^M$.
On input a verification-key $VK = (PK', h)$, where $h \in \mathcal{H}_n^{FS}$ and

$$PK' = (PK, f, f^{UA}, (k, r), (b_2', q')),$$

The forger $FORG^M$ generates a signature of M, with respect to VK, as follows.

1. Compute $a = F_1^n(f)$.
2. (a) Choose $b_1 \in_R \{0,1\}^n$, and compute $r' = comm_k(a, b_1; r)$.

   (b) Choose $h_2, \ldots, h_n \in_R \mathcal{H}_n^{FS}$, and let

   $$q^M = (f^{UA}, (k, r'), (h^M, h_2^M, \ldots, h_n^M)).$$

   (c) Emulate the interaction of $(\tilde{P}_1^n, V^{\mathcal{H}^M}|_{q^M})(f, a, b_1)$ to obtain a transcript

   $$(q^M; ans^M) \leftarrow (\tilde{P}_1^n, V^{\mathcal{H}^M}|_{q^M})(f, a, b_1).$$

   Denote $ans^M = (\hat{\beta}_1, (\gamma_1, \ldots, \gamma_n), \beta_1, (\delta_1, \ldots, \delta_n))$.
3. Compute $(*, (b_2'', q'')) = h(\hat{\beta}_1, M)$.
4. Emulate the interaction of $(\tilde{P}_n^1, V^{\mathcal{H}^1}|_{q' \oplus q''})(f, a, b_2' \oplus b_2'')$, to obtain a transcript

   $$(q' \oplus q''; ans) \leftarrow (\tilde{P}_n^1, V^{\mathcal{H}^1}|_{q' \oplus q''})(f, a, b_2' \oplus b_2'').$$

5. Output

   $$(\hat{\beta}_1, (\gamma_1, (b_2'', q'')), (a, b_1, \beta_1, \delta_1, ans))$$

   as a signature of M.

We claim that the forger will be successful with non-negligible probability.

Claim 6.6.1.

$$Pr[VERIFY_{\mathcal{H}}^3(VK, M, FORG^M(VK)) = 1] = non\text{-}negl(n)$$

(where the probability is over VK and over the random coin tosses of $FORG^M$).

Proof. Denote the output of the forger $FORG^M(VK)$ by $(\hat{\beta}_1, (\gamma_1, (b_2'', q'')), (a, b_1, \beta_1, \delta_1, ans))$. By definition of $\tilde{\tilde{P}}_1^n$, for $a = F_1^n(f)$,

$$Pr[(\tilde{\tilde{P}}_1^n, V^{\mathcal{H}^1})(f, a, b_1) = 1 \ \wedge \ (\tilde{\tilde{P}}_1^n, V^{\mathcal{H}^M})(f, a, b_2) = 1] = non\text{-}negl(n) \tag{3}$$

(where the probability is over $f \in_R \mathcal{F}_n,\ b_1, b_2 \in_R \{0,1\}^n$ and the random coin tosses of $V^{\mathcal{H}^1}$ and $V^{\mathcal{H}^M}$).

We claim that similarly, for $a = F_1^n(f)$,

$$Pr[(\tilde{\tilde{P}}_1^n, V^{\mathcal{H}^1}|_{q' \oplus q''})(f, a, b_1) = 1 \ \wedge \ (\tilde{\tilde{P}}_1^n, V^{\mathcal{H}^M}|_{q_M})(f, a, b_2' \oplus b_2'') = 1] = non\text{-}negl(n) \tag{4}$$

(where the probability is over $f \in_R \mathcal{F}_n,\ \ b_1, b_2' \oplus b_2'' \in_R \{0,1\}^n$, $q' \oplus q''$, q_M).
This is so for the following reasons

1. $b_2' \oplus b_2''$ is uniformly distributed in $\{0,1\}^n$.
2. $q' \oplus q''$ is uniformly distributed among the set of all queries of $V^{\mathcal{H}^1}$.
3. $\tilde{\tilde{P}}_1^n$ (in step 2(c)) cannot distinguish between the distribution of q_M and the distribution of a uniform query of $V^{\mathcal{H}^M}$.

For all of the above reasons, $\tilde{\tilde{P}}_n^1$ in (4) should succeed with essentially the same probability as in (3).

Thus, with non-negligible probability both of the following conditions hold.

1. $(q' \oplus q''; ans) \in VIEW(V^{\mathcal{H}^1}(f, a, b_2' \oplus b_2''))$.
2. $(q^M; ans^M) \in VIEW((V^{\mathcal{H}^M}(f, a, b_1))$, which in turn implies that the following conditions hold.
   (a) $\gamma_1 = h^M(\hat{\beta}_1)$, which implies that $(\gamma_1, (b_2'', q'')) = h(\hat{\beta}_1, M)$
   (b) $(f^{UA}; \beta_1; \gamma_1; \delta_1) \in VIEW(V^0(f, a, b_1))$
   (c) $\hat{\beta}_1 = comm_k(\beta_1; comm_k(a, b_1; r)$.

Recall that $VERIFY_{\mathcal{H}}^3(VK)$ accepts if conditions (1) and (2) hold, and thus $FORG^M(VK)$ is successful with non-negligible probability. □

Thus, we have established the insecurity of SS^3. □

7 On the Insecurity of FS Modifications

The FS method was designed for constructing signature schemes by eliminating interaction from canonical ID schemes. We proved that this method is insecure, in the sense that there exist secure canonical ID schemes for which the corresponding signature scheme (obtained by the FS method) is insecure with respect to any function ensemble. A question that remains is: Do there exist other *secure* methods for eliminating interaction?

We present two modifications of the FS method considered in the literature. Using similar ideas to the ones presented in this paper, we show the insecurity of these FS modifications as well.

7.1 First Modification

We first present the FS modification introduced by Micali and Reyzin. In their paper, *'Improving The Exact Security of Digital Signature Schemes'* [MR02], they presented a method for constructing FS-like signature schemes that yields better "exact security" than the original FS method (α, β, γ). In their method, the signer first chooses β (originally sent by the receiver R) and then produces α (the first message of the sender S), by applying H to β and to the message to be signed, i.e., $\alpha = H(\beta, M)$.[13] This method can be applied only to ID schemes in which the sender, given public-key PK and a pair (α, β), can efficiently compute γ for which $(\alpha, \beta, \gamma) \in VIEW(R(PK))$. This method does not apply to ID schemes in which the information used during the generation of α is necessary to compute γ.

We argue that this FS-like method proposed by [MR02] is insecure, as follows. Take any secure ID scheme and modify it by appending $f \in_R \mathcal{F}$ to the public key, and extending its verdict function so as to also accept views of the following form

$$\begin{array}{ccc} S & (PK, f) & R \\ & \xrightarrow[a]{\hspace{3em}} & \\ & \xleftarrow[b,q]{\hspace{3em}} & \\ & \xrightarrow[ans]{\hspace{3em}} & \end{array}$$

where

$$(q; ans) \in VIEW(V^{\mathcal{H}}(f, (b, q), a)).$$

[13]They called this method the Swap method since they swapped the roles of α and β in the original FS method.

We denote this extended ID scheme by $ID_{\mathcal{H}}$. It is relatively easy to show that the signature scheme, obtained by applying the above FS-like method to $ID_{\mathcal{H}}$, is insecure with respect to any function ensemble. Thus, if there exists a function ensemble $\mathcal{H}$ such that $ID_{\mathcal{H}}$ is secure, then the aove FS-like method is insecure. Namely, under the CSP hypothesis, the above FS-like method is insecure. To complete the proof one needs to assume that for every function ensemble $\mathcal{H}$, $ID_{\mathcal{H}}$ is insecure. The rest of the proof is quite technical and follows the lines of Sections 6.2 and 6.3.

7.2 Second Modification

We next present the FS modification introduced by Abdalla, An, Bellare and Namprempre. In their paper, *'From Identification to Signatures via the Fiat-Shamir Transform: Minimizing assumptions for Security and Forward-Security'* [AABN02], they define a randomized generalization of the Fiat-Shamir transform, and prove that a necessary and sufficient condition for the security (resp. forward-secure) of signature schemes obtained from the generalized FS transform in the Random Oracle Model, is that the underlying ID scheme is secure (resp. forward-secure) against impersonation under passive attacks.

The randomized generalization of the FS transform transforms any canonical ID scheme (α, β, γ) into a signature scheme by replacing β with the value of H applied to α, M (the message to be signed) and R (randomness chosen by the Signer). That is, a valid signature of a message M, with respect to a public-key PK, is a triplet $(\alpha, (\beta, R), \gamma))$ such that

1. $\beta = H(\alpha, M, R)$
2. $(\alpha, \beta, \gamma) \in VIEW(R(PK))$.

The insecurity of the above generalized FS paradigm, follows trivially from the fact that it is a generalization of the original FS paradigm with $R = \emptyset$, and from the fact that the original FS paradigm is insecure.

8 Open Problems

Do there exist other "natural" cryptographic schemes which are secure in the Random Oracle Model, and become insecure when the random oracle is replaced with any public function? An example of a "natural" cryptographic scheme that we are interested in is CS-proofs, defined by Micali [Mi94]. The question is whether or not there exists a function ensemble $\mathcal{H}$, such that CS-proofs remain sound (or remain a proof-of-knowledge) when the random oracle is replaced with a public function chosen at random from $\mathcal{H}$?

Perhaps most interestingly, one would like to prove that either every "natural" task which is realizable in the Random Oracle Model is also realizable in the "real world," or that there exists a "natural" task which is realizable in the Random Oracle Model and is not realizable in the "real world." For example, we know that there exists an identity based encryption scheme which is secure in the Random Oracle Model [BF01]. But, does there exist an identity based encryption scheme which are secure in the "real world."?

References

[AABN02] M. Abdalla, J. An, M. Bellare and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: minimizing assumptions for security and forward-security. *Advances in Cryptography-EUROCRYPT 02, Lecture Notes in Computer Science, Springer-Verlag*, 2002.

[Bar01] B. Barak. How to go beyond the black-box simuation barrier. In *Proc. of the 42nd FOCS*, 2001.

[BF01] D. Boneh and M. Franklin. Identity-based encryption from Weil pairing. Preliminary version in *Crypto*, 2001.

[BG01] B. Barak and O. Goldreich. Universal arguments and their applications. *Proceedings of the 17th IEEE Annual Conference on Computational Complexity*, 2002.

[BGI+01] B. Barak, O. Goldreich, R. Rudich, A. Sahai, S. Vadhan, and K.Yang. On the (im)possibility of obfuscating programs. In *Crypto* 2001.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the First Annual Conference on Computer and Communications Security*. ACM, November 1993.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 209-218, Dalllas, 23-26 May, 1998.

[CS99] R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. In *5th ACM Conference on Computer and Communications Security*, pages 46-51. Singapore, Nov. 1999. ACM Press.

[DH76] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22 (Nov.), pages 644-654, 1976.

[DNRS99] C. Dwork, M. Naor, O. Reingold and L.Stockmeyer. Magic funcitons. In *IEEE, editor, 40th Annual Symposium of Foundations of Computer Science*: October 17-19, 1999, New York City, New York, pages 523-534. *IEEE Computer Society Press*, 1999.

[FS86] Amos Fiat and Adi Shamir. How to prove to yourself: practical solutions to identification and signature problems. In *Advances in Cryptology–Crypto 86,* pages 186-194, Springer, Berlin, 1987.

[GHR99] R. Gennaro, S. Halevi and T. Rabin. Secure hash-and-sign signatures without the random oracle. *Advances in Cryptology - EUROCRYPT 99*, Lecture Notes in Computer Science Vol. 1592, J. Stern ed., Springer-Verlag, 1999.

[GGM86] Oded Goldreich, Shafi Goldwasser and Silvio Micali. How to construct random functions. *Journal of the Association of Computing Machinery*, 33(4): 792-807, 1986.

[GMR88] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17(2):281-308, April 1988.

[GQ88] L. Guillou and J. J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. *Advances in Cryptology-CRYPTO 88*, Lecture Notes in Computer Science Vol. 403, S. Goldwasser ed., Springer- Verlag, 1988.

[Ki92] J. Kilian. A note on efficient zero-knowledge proofs and arguments. In *24'th STOC*, pages 723-732, 1992.

[Mer90] R.C. Merkle. A certified digital signature. *Proceedings on Advances in Cryptology*, pages 218-238, July 1989, Santa- Barbara, California.

[Ok92] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. *Advances in Cryptology - CRYPTO 92*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.

[MD5] R. Rivest. The MD5 message-digest algorithm. *RFC* 1321, April 1992.

[Mi94] Silvio Micali. Computationally sound proofs. *SICOMP*, vol. 30(4), pages 1253-1298, 2000. Preliminary version in *35th FOCS*, 1994.

[MR02] S. Micali and L. Reyzin. Improving the exact security of digital signature schemes. *Jornal of Cryptology*, 15(1):1-18, 2002.

[Na91] M. Naor. Bit commitment using pseudorandom generators. *Journal of Cryptology*, Vol.4, pages 151-158, 1991.

[NY89] M. Naor and M. Yung. Universal one-way hash functions and their cryptogracphic applications. *STOC 89*.

[PS96] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Advances in Cryptology-EUROCRYPT 96*, vol.1070 of Lecutre Notes in Compute Science, pages 387-398. Springer-Verlag, 1996.

[Sch91] Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology* 4(3):161-174.

A Commitment Schemes

Naor [Na91] proved that commitment schemes exist assuming the existence of one-way function ensembles. Namely, assuming the existence of one-way function ensembles, there exists functions $l(n)$ and $t(n)$, which are polynomially related to n, and there exists a commitment scheme $COMMIT$ such that for every $n \in \mathbb{N}$ and for every $k \in KEY_n$,

$$commit_k : \{0,1\}^n \times \{0,1\}^{l(n)} \to \{0,1\}^{t(n)}.$$

Proposition 3. *Under the CR hypothesis, For any function $m(n)$, which is polynomially-related to n, there exists a commitment scheme $COMM$, with a corresponding set of keys KEY', such that for every $n \in \mathbb{N}$ and for every $k' \in KEY'_n$,*

$$comm_{k'} : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}^n.$$

Proof. Let $\mathcal{F}^m$ be a collision resistant function ensemble such that for every $n \in \mathbb{N}$ and for every $f_n^m \in \mathcal{F}_n^m$,

$$f_n^m : \{0,1\}^{m(n)} \to \{0,1\}^n.$$

(the existence of such a function ensemble follows from the CR hypothesis).

Let $\mathcal{F}^t$ be a collision resistant function ensemble such that

1. for every $n \in \mathbb{N}$ and for every $f_n^t \in \mathcal{F}_n^t$,
$$f_n^t : \{0,1\}^{t(n)} \to \{0,1\}^n$$

2. for every $n \in \mathbb{N}$,
$$f_n^t(U_{t(n)}) \cong U_n.$$

(It is quite easy to see that such a function ensemble exists under the CR hypothesis).

The set of keys for $COMM$ is defined as follows: For every $n \in \mathbb{N}$,

$$KEY'_n = \{(k, f_n^m, f_n^t) : \ k \in KEY_n, \ f_n^m \in \mathcal{F}_n^m, \ , f_n^t \in \mathcal{F}_n^t\}.$$

For every $n \in \mathbb{N}$, every $(k, f_n^m, f_n^t) \in KEY'_n$ and every $(x, r) \in \{0,1\}^{m(n)} \times \{0,1\}^n$, define

$$comm_{(k, f_n^m, f_n^t)}(x; r) = f_n^t(commit_k(f^m(x); g(r)),$$

where $g : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$ is a one-way pseudorandom generator.[14]

$COMM$ is computationally-hiding since

1. g is a pseudorandom generator
2. $COMMIT$ is computationally-hiding
3. $f_n^t(U_{t(n)}) \cong U_n$.

$COMM$ is computationally-binding since

1. $\mathcal{F}^t$ is a collision-resistance function ensemble
2. $COMMIT$ is computationally-binding
3. $\mathcal{F}^m$ is a collision-resistance function ensemble.
4. g is one-way.

$\square$

[14] It was proven in [GGM86] that one-way pseudorandom generators exist assuming the existence of one-way function ensembles.