NOTES ON
From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security

Abdalla, An, Bellare and Namprempre

September 27, 2004

1 Background

- A **canonical** id scheme $ID = (G, P, V)$ is simply a three-round, public-coin id scheme. In other words, a key pair (pub, pri) is generated by running G on 1^n, the prover P_{pri} goes first and the verifier V_{pub}'s only message, called the *challenge* and denoted by $\textsc{Ch}$, consists of his random bits. P's two messages are called *committment* and *response*, denoted by $\textsc{Cmt}$ and $\textsc{Rsp}$, respectively.

  The *completeness* property asserts that P_{pri} can convince V_{pub} to accept no matter what the random challenge $\textsc{Ch}$ is.

  NOTE: Throughout, we assume that the public key pub is part of the private key pri and that $|pub|$ uniquely determines n, say $|pub| = \ell(n) \geq n$, where ℓ^{-1} is a polytime computable function.

  Also, we view V's private coins, $\textsc{Ch}$, as being chosen externally, so that $V_{pub}(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$ is a deterministic (boolean) function.

- Consider A canonical id scheme is **nontrivial** if the **min entropy** of the committments distribution is superlogarithmic in the security parameter n.

  Recall that the min entropy of an arbitrary discrete distribution $D = \{p_i\}_{i=0}^k$ on k points is defined as $H_{min}(D) = \log_2(1/p_{max}) = -\log_2(p_{max})$, where $p_{max} = \max\{p_i\}_{i=0}^k$ is the largest probability mass. Since each pri generated by $G(1^n)$ induces its own committments distribution P_{pri} and we'd like to express the min entropy $H_{min}(n)$ as a function of the security parameter n only, we compute $H(pri) = H_{min}(P(pri))$ for each pri and set

  $$H_{min}(n) = \min_{(pub,pri)\leftarrow G(1^n)} \{H(pri)\}.$$

  In the special (but ubiquitous) case that P_{pri} is uniformly distributed over $\{0,1\}^{f(n)}$ for every pri, where $f : \mathbb{N} \to \mathbb{N}$ is some function, we have $p_{max} = \frac{1}{|\{0,1\}^{f(n)}|} = \frac{1}{2^{f(n)}}$ and $H_{min}(n) = \log_2(1/p_{max}) = \log_2(2^{f(n)}) = f(n)$. Hence $H_{min}(n)$ is superlogarithmic in n iff

$f(n)$ is superlogarithmic, $f(n) \equiv n$ say. Note that if $f(n)$ is superlogarithmic in n, i.e. $f(n) = \omega(\log(n))$, then $|\{0,1\}^{f(n)}| = 2^{f(n)}$ is superpolynomial in n: if $2^{f(n)} = n^c$ for some c, then $f(n) = c \cdot \log(n) = \Omega(\log(n))$, contradicting $f(n) = \omega(\log(n))$.

We may therefore informally say that a canonical id scheme is nontrivial if the prover's committment space is "large", i.e. of size superpolynomial in n. Observe that sampling such a committment space polynomially many times is unlikely to yield repetitions, in the sense that the probability of getting the same element twice is negligible in n (this can be shown using the union bound).

- Informally, an id scheme is **secure against passive attacks**, or **passively secure**, if the probability that a probabilistic polytime impersonator I gets the verifier V_{pub} to accept – given *pub* and having seen polynomially many transcripts of interactions between V_{pub} and P_{pri} – is negligible in n. Here the probability is taken over all key pairs $(pub, pri) \leftarrow G(1^n)$.

  In other words, although I can *eavesdrop* on conversations between P_{pri} and V_{pub}, he cannot attempt to extract information from P_{pri} by interacting with it arbitrarily.

  This is a weaker notion of security than the standard *active* one, where I_{pub} gets to interact with P_{pri} in the role of V_{pub} before attempting impersonation.

- The **Fiat-Shamir transform** is a way of converting a canonical id scheme $ID = (G, P, V)$ into a signature scheme $SIG_{\mathcal{H}}(ID) = (G, SIGN_{\mathcal{H}}, VER_{\mathcal{H}})$ using a function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^{c(n)}$ (or, more precisely, an ensemble $\{\mathcal{H}_n\}_{n \in \mathbb{N}}$ of such functions), where $c(n) : \mathbb{N} \rightarrow \mathbb{N}$ is the length of V's challenge on security parameter n.

  We are mostly interested in *efficiently computable* $\mathcal{H}$, i.e. $\mathcal{H}$ for which there exists a deterministic Turing machine $\mathcal{M}_{\mathcal{H}}$ such that $\mathcal{M}_{\mathcal{H}}(m) = \mathcal{H}(m)$ for all $m \in \{0,1\}^*$. However, our definition of $SIG_{\mathcal{H}}(ID)$ makes sense even if $\mathcal{H}$ isn't efficiently computable.

  To sign a message $m \in \{0,1\}^*$ with respect to $(pub, pri) \leftarrow G(1^n)$, $SIGN_{\mathcal{H}}(pri, m)$ simulates $P(pri)$ to obtain a committment $\textsc{Cmt}$, deterministically computes a challenge $\textsc{Ch} = \mathcal{H}(\textsc{Cmt}, m)$, and again simulates $P(pri, \textsc{Cmt}, \textsc{Ch})$ to obtain a response $\textsc{Rsp}$. It then outputs $\sigma_m = (\textsc{Cmt}, \textsc{Rsp})$ as the signature of m.

  To verify that (α, γ) really is the signature of m with respect to (pub, pri), $VER_{\mathcal{H}}(pub, m, (\alpha, \gamma))$ first computes $\beta = \mathcal{H}(\alpha, m)$ and then outputs 1 if $V(pub, \alpha, \beta, \gamma) = 1$ and 0 otherwise.

  Notice that the completeness property of ID guarantees that, for every message m,

  $$\Pr[VER_{\mathcal{H}}(pub, m, (\textsc{Cmt}, \textsc{Rsp})) = 1] = \Pr[V(pub, \textsc{Cmt}, \mathcal{H}(\textsc{Cmt}, m), \textsc{Rsp}) = 1] = 1,$$

  where $SIGN_{\mathcal{H}}(pri, m) = (\textsc{Cmt}, \textsc{Rsp})$ and the probability is taken over $(pub, pri) \leftarrow G(1^n)$.

  In other words, $SIGN_{\mathcal{H}}(pri, m)$ always outputs a legitimate signature (with respect to (pub, pri)) of m.

- A few words about the **Random Oracle Model** and the **Random Oracle Methodology**, both formally introduced in [BR93], are in order.

  Consider a cryptographic primitive $\pi_{\mathcal{H}}$ which makes use of a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^{c(n)}$, for example a signature scheme $SIG_{\mathcal{H}}(ID)$ obtained by applying the Fiat-Shamir transform (with respect to $\mathcal{H}$) to some canonical id scheme ID. Our goal is to show that $\pi_{\mathcal{H}}$

is secure in some appropriate sense, e.g. that $SIG_{\mathcal{H}}(ID)$ is secure against existential forgery under chosen message attack. Since $\mathcal{H}$ is just an efficiently computable function, we should think of our adversaries as being given $\mathcal{H}$ as part of their input.

If $\mathcal{H}$ is a "cryptographic hash function", then it is assumed to be collision resistant in some sense (note that this implies one-wayness). The strongest assumption typically made is that collisions, i.e. strings $m_1 \neq m_2$ such that $\mathcal{H}(m_1) = \mathcal{H}(m_2)$, are infeasible to find. However, we would need to introduce function ensembles $\{\mathcal{H}_n\}_{n\in\mathbb{N}}$ to really define collision-resistance properly.

In any event, it may not be clear how to prove the security of $\pi_{\mathcal{H}}$ if the only assumption we make about $\mathcal{H}$ is collision-resistance of some kind. The Random Oracle Methodology suggests that we instead model $\mathcal{H}$ as a random function $\rho : \{0,1\}^* \to \{0,1\}^{c(n)}$. In other words, rather than show that $\pi_{\mathcal{H}}$ is secure for any particular $\mathcal{H}$, we settle for showing that π is secure against adversaries which are have oracle access to ρ, where – notionally, at least – ρ is randomly chosen from the set of all functions mapping $\{0,1\}^*$ into $\{0,1\}^{c(n)}$.

Since there are infinitely many such functions, we prefer to think of ρ as being defined incrementally: whenever the adversary asks to see $\rho(m)$ for a new message $m \in \{0,1\}^*$ (i.e. one the oracle hasn't been queried on already), it will be given a randomly chosen $y \in \{0,1\}^{c(n)}$; if the advesary subsequently quieries ρ on m again, it will be shown y once more. This ensures that ρ is a well-defined function and results in the same distribution as randomly choosing ρ in advance.

Proving that π is secure in the Random Oracle Model (which we'll refer to as **the ROM** from now on) is often much easier than showing that $\pi_{\mathcal{H}}$ is secure for any particular choice of $\mathcal{H}$. However, security in the ROM is no guarantee of "real-world" security, as shown in [CGH98], where Canetti et al construct a signature scheme and a public key encryption primitive (PKEP) secure in the ROM but not in the standard model. Both constructions are quite artifical – Micali's CS proofs, defined in [Mic00], make an appearance – and exploit the fact that standard-model adversaries effectively get to see the actual code of $\mathcal{H}$, instead of just being given oracle access to it; a similar idea was used by Barak in [Bar01].

Consider the signature scheme obtained by applying the Fiat-Shamir transform with respect to $\rho : \{0,1\}^* \to \{0,1\}^{c(n)}$ to a canonical id scheme $ID = (G, P, V)$, where $c(n)$ is the length of V's challenge on security parameter n, wholly determined by ID; we will denote this signature scheme by $SIG(ID)$. Although ρ isn't really a fixed function, our meaning is hopefully clear.

We say that $SIG(ID) = (G, SIGN, VER)$ is secure in the ROM if no probabilistic polytime adversary ADV which is given the public key pub and oracle access to both $SIGN_{pri}$ and ρ succeeds in producing a new message $m \in \{0,1\}^*$ (i.e. one $SIGN_{pri}$ hasn't been queried on), together with a supposed signature σ such that $VER_{pub}(m,\sigma) = 1$, with probability non-negligible in the security parameter n. Here the probability is taken over $(pub, pri) \leftarrow G(1^n)$, ADV's coins **and the randomness of** ρ.

2 Results

NOTE: results regarding **forward-secure signature schemes** and the **randomized Fiat-Shamir transform** are of no relevance to us and hence have been omitted.

Let $ID = (G, P, V)$ be a **canonical** id scheme.

1. ID is **passively secure** and **nontrivial** $\Rightarrow$ $SIG(ID)$ is **secure in the ROM**

   Proof Sketch: The proof is by a standard black-box reducibility argument. We show how to convert a forger F which breaks the security of $SIG(ID)$ in the ROM into an impersonator I which breaks the passive security of ID.

   Suppose that we are given a probabilistic polytime forger F which breaks the security of $SIG(ID)$ in the ROM. F is given the public key *pub* and has access to both a random oracle $\mathcal{R}$ and a signing oracle $\mathcal{S}$: every time $\mathcal{S}$ is queried on a message $m' \in \{0,1\}^*$, it outputs a signature $\sigma' = (\textsc{Cmt}', \textsc{Rsp}')$ of m' such that $V_{pub}(\textsc{Cmt}', \textsc{Ch}', \textsc{Rsp}') = 1$, where $\textsc{Ch}' = \mathcal{R}(\textsc{Cmt}', m')$.

   The probability that $F_{pub}^{\mathcal{R},\mathcal{S}}$ eventually outputs a new message m together with a signature $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ such that $V_{pub}(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp}) = 1$, where $\textsc{Ch} = \mathcal{R}(\textsc{Cmt}, m)$, taken over $(pub, pri) \leftarrow G(1^n)$, the randomness of $\mathcal{R}$ and $\mathcal{S}$ and the coins of $F_{pub}^{\mathcal{R},\mathcal{S}}$, is non-negligible in n.

   Notice that signing in $SIG(ID)$ is probabilistic, so it makes sense for $F_{pub}^{\mathcal{R},\mathcal{S}}$ to query $\mathcal{S}$ on the same string multiple times. On the other hand, there's nothing to gain from querying $\mathcal{R}$ on the same string more than once, so we may assume (wlog) that $F_{pub}^{\mathcal{R},\mathcal{S}}$ does no such thing.

   We can also safely assume that $F_{pub}^{\mathcal{R},\mathcal{S}}$ queries $\mathcal{R}$ on $(\textsc{Cmt}, m)$ before outputting m and σ. If it doesn't, we can easily construct another forger, $F_{pub}^{\prime\mathcal{R},\mathcal{S}}$, that does: $F_{pub}^{\prime\mathcal{R},\mathcal{S}}$ simply simulates $F_{pub}^{\mathcal{R},\mathcal{S}}$ until the latter outputs a message m and a signature σ), then queries $\mathcal{R}$ on $(\textsc{Cmt}, m)$ and outputs (m, σ). This additional random oracle query, which we'll call the "crucial query", has no effect on $F_{pub}^{\prime\mathcal{R},\mathcal{S}}$'s success probability, since it does not affect the choice of m or σ.

   We construct a probabilistic polytime impersonator I which, given a public key *pub* and access to a transcript-generating oracle $\mathcal{T}$, gets V_{pub} to accept with probability non-negligible in the security parameter n. Here the probability is taken over $(pub, pri) \leftarrow G(1^n)$, the randomness of $\mathcal{T}$ and the coins of $I_{pub}^{\mathcal{T}}$.

   Say that the running time of $F_{pub}^{\mathcal{R},\mathcal{S}}$ is bounded above by n^c, no matter what the outcome of its coin tosses is; such a c exists since $F_{pub}^{\mathcal{R},\mathcal{S}}$ is assumed to run in strict polynomial time. This means that for any random tape, $F_{pub}^{\mathcal{R},\mathcal{S}}$ makes at most n^c random oracle queries, which we'll number from 1 to n^c.

   $I_{pub}^{\mathcal{T}}$ begins its simulation of $F_{pub}^{\mathcal{R},\mathcal{S}}$ by guessing the index of the "crucial query": $I_{pub}^{\mathcal{T}}$ uniformly selects an index $i \in \{1, \cdots, n^c\}$ and hopes that $F_{pub}^{\mathcal{R},\mathcal{S}}$'s i^{th} random oracle query is about $\textsc{Cmt} \circ m$; since we've assumed that $F_{pub}^{\mathcal{R},\mathcal{S}}$ does make the crucial query at some point and that it never queries $\mathcal{R}$ on the same string twice, this happens with probability $\frac{1}{n^c}$.

   What does $I_{pub}^{\mathcal{T}}$ gain by correctly guessing i? It learns the value of $\textsc{Cmt}$, which it can send to V_{pub} to obtain a challenge $\textsc{Ch}$. $I_{pub}^{\mathcal{T}}$ can then give $\textsc{Ch}$ to $F_{pub}^{\mathcal{R},\mathcal{S}}$ as the answer to the crucial query, thereby ensuring that the message $\textsc{Rsp}$, which the latter eventually outputs as part of σ, is the correct answer to $\textsc{Ch}$.

   During the simulation, $I_{pub}^{\mathcal{T}}$ responds to all of $F_{pub}^{\mathcal{R},\mathcal{S}}$'s random oracle queries but the i^{th} with a uniformly chosen string from $\{0,1\}^{c(n)}$ (recall that $F_{pub}^{\mathcal{R},\mathcal{S}}$ never queries $\mathcal{R}$ on the same string twice, by assumption); the i^{th} query is handled specially.

Suppose that that for its i^{th} random oracle query, $F_{pub}^{\mathcal{R},\mathcal{S}}$ asks to see $\mathcal{R}(m')$ for some $m' \in \{0,1\}^*$. $I_{pub}^{\mathcal{T}}$ parses m' as $\textsc{Cmt} \circ m$, sends $\textsc{Cmt}$ to V_{pub} and receives a challenge $\textsc{Ch} \in \{0,1\}^{c(n)}$ in response. It then gives $\textsc{Ch}$ to $F_{pub}^{\mathcal{R},\mathcal{S}}$ as the answer to its query and continues the simulation.

Whenever $F_{pub}^{\mathcal{R},\mathcal{S}}$ asks to see a signature of a message m', $I_{pub}^{\mathcal{T}}$ queries $\mathcal{T}$ to obtain a random transcript $\textsc{Tr}' = (\textsc{Cmt}', \textsc{Ch}', \textsc{Rsp}')$ and then gives $\textsc{Tr}'$ to $F_{pub}^{\mathcal{R},\mathcal{S}}$.

There is a problem with this approach if $F_{pub}^{\mathcal{R},\mathcal{S}}$ queries $\mathcal{S}$ on the same message m' many times, which is a reasonable thing to do because signing in $SIG(ID)$ is probabilistic. In order for the resulting transcripts to be legitimate signatures of m', it must be the case that $\mathcal{R}(\textsc{Cmt}', m') = \textsc{Ch}'$ for all of them. $\mathcal{T}$, however, generates $\textsc{Ch}'$ randomly.

If the committments $\textsc{Cmt}'$ are distinct then $\textsc{Cmt}' \circ m'$ is a new message, so that $\mathcal{R}(\textsc{Cmt}', m')$ is distributed uniformly over $\{0,1\}^{c(n)}$ and we're fine. If the committments match, on the other hand, then we will likely run into trouble: should the challenges $\textsc{Ch}'$ differ, as they almost certainly will (since they're chosen randomly from $\{0,1\}^{c(n)}$), $\mathcal{R}(\textsc{Cmt}', m')$ will appear to have multiple values, preventing $\mathcal{R}$ from being a well-defined function. Therefore a new committment $\textsc{Cmt}'$ is added to the (notional) set of "forbidden committments" every time $\mathcal{S}$ is queried on m'. In the worse case, $F_{pub}^{\mathcal{R},\mathcal{S}}$ does nothing but query $\mathcal{S}$ on some message m', which means that the size of the "forbidden committment set" is bounded above by n^c.

A similar situation can occur even if $F_{pub}^{\mathcal{R},\mathcal{S}}$ doesn't query $\mathcal{S}$ on the same message more than once. Observe that if $\textsc{Tr}' = (\textsc{Cmt}', \textsc{Ch}', \textsc{Rsp}')$ is to be a legitimate signature of m', we must have $\mathcal{R}(\textsc{Cmt}', m') = \textsc{Ch}'$. So every $\mathcal{S}$ query effectively involves an implicit $\mathcal{R}$ query. But what if $\mathcal{R}$ has been queried on $\textsc{Cmt}' \circ m'$ already? Unless we're very lucky and $\textsc{Ch}'$ matches the value previously assigned to $\mathcal{R}(\textsc{Cmt}', m')$, this, too, prevents $\mathcal{R}$ from being well-defined. Therefore a new committment $\textsc{Cmt}'$ is added to the "forbidden committment set" every time $\mathcal{R}$ is queried on $x \in \{0,1\}^*$ – simply parse x as $\textsc{Cmt}' \circ m'$. In the worst case, $F_{pub}^{\mathcal{R},\mathcal{S}}$ queries $\mathcal{R}$ on as many strings $x_1, x_2, x_3, \ldots$, which parse as $\textsc{Cmt}'_1 \circ m'$, $\textsc{Cmt}'_2 \circ m'$, $\textsc{Cmt}'_3 \circ m'$,$\ldots$, as possible, and then queries $\mathcal{S}$ on m'. Here the size of the "forbidden committment set" is bounded above by $n^c - 1$, because the signing query contributes no forbidden committment.

Both of these difficulties can be resolved by appealing to the nontriviality assumption: since ID is nontrivial, the probability that a randomly chosen committment $\textsc{Cmt}'$ belongs to the "forbidden committment set" is negligible in n, so we can safely ignore that eventuality.

Eventually, $F_{pub}^{\mathcal{R},\mathcal{S}}$ outputs a message m together with a supposed signature $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ of m. This only happens after all of its random oracle queries, and in particular the i^{th}, have been answered, which means that V_{pub} has already challenged $I_{pub}^{\mathcal{T}}$ by this point.

$I_{pub}^{\mathcal{T}}$ responds by sending $\textsc{Rsp}$ to V_{pub}. What is the probability that V_{pub} accepts?

Since $F_{pub}^{\mathcal{R},\mathcal{S}}$ succeeds in outputting a legitimate signature $\sigma = (\textsc{Cmt}, \textsc{Rsp})$ of the message m with non-negligible probability, there is a d such that

$$p(n) = \Pr[V_{pub}(\textsc{Cmt}, \mathcal{R}(\textsc{Cmt}, m), \textsc{Rsp}) = 1] > \frac{1}{n^d} \text{ for infinitely many } n,$$

where the probability is taken over the coins of $F_{pub}^{\mathcal{R},\mathcal{S}}$, the randomness of $\mathcal{S}$ – recall that signing in $SIG(ID)$ is probabilistic – and the randomness of $\mathcal{R}$.

What is the probability that $I^{\mathcal{T}}_{pub}$ gets V_{pub} to accept after sending it $\textsc{Cmt}$ and $\textsc{Rsp}$, i.e. that $V_{pub}(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp}) = 1$, taken over V_{pub}'s coins ($\textsc{Ch}$), the randomness of the transcript oracle $\mathcal{T}$ and the coins of $I^{\mathcal{T}}_{pub}$?

If $I^{\mathcal{T}}_{pub}$ correctly guesses the index i of $F^{\mathcal{R},\mathcal{S}}_{pub}$'s $\mathcal{R}(\textsc{Cmt}, m)$ query and its simulation of $F^{\mathcal{R},\mathcal{S}}_{pub}$ is not plagued by any "committment collision" issues – which only crop up with negligible probability and can therefore be safely ignored (informally, at least) – then the answer is again $p(n)$. Since the guessing is independent from the simulation, we obtain:

$$\Pr[V_{pub}(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp}) = 1] \approx \frac{1}{n^c} \cdot p(n) > \frac{1}{n^c} \cdot \frac{1}{n^d} = \frac{1}{n^{c+d}} \text{ for infinitely many } n.$$

This shows that $I^{\mathcal{T}}_{pub}$'s success probability is non-negligible in n, so $I^{\mathcal{T}}_{pub}$ breaks the passive security of ID and we are done.

Remarks: Observe that the interaction between $\mathcal{S}$ and $\mathcal{R}$ – namely the whole implicit queries business – forced us to invoke the nontriviality assumption even in the case that $F^{\mathcal{R},\mathcal{S}}_{pub}$ did not query $\mathcal{S}$ on the same message more than once. More precisely, we required *some* assumption strong enough to guarantee that the probability of landing in the forbidden commitment set when sampling the committment space is negligible in n.

However, there would be no need for this type of assumption if F_{pub} were not allowed to query $\mathcal{S}$ at all, which yields the following result:

$$ID \text{ passively secure } \Rightarrow SIG(ID) \text{ passively secure in the ROM},$$

where a signature scheme is deemed **passively secure in the ROM** if it is secure in the ROM against adversaries not allowed any $\mathcal{S}$ queries; note that ordinary ROM security implies passive ROM security.

2. ID is **passively secure** $\Leftarrow SIG(ID)$ is **secure in the ROM**

   Proof Sketch: The proof is again by a black-box reducibility argument. We show how to convert an impersonator that breaks the passive security of ID into a forger that breaks the security of $SIG(ID)$ in the ROM.

   Suppose that we are given a probabilistic polytime impersonator I that breaks the passive security of ID: a key pair (pub, pri) is generated by running G on 1^n together with random bits; I is given the public key pub and has access to a transcript-generating oracle $\mathcal{T}$ which probabilistically simulates the interaction between P_{pri} and V_{pub} to produce a transcript $\textsc{Tr} = (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$ every time it's queried; the probability that $I^{\mathcal{T}}_{pub}$ gets V_{pub} to accept after being shown a bunch of legitimate transcripts by $\mathcal{T}$, taken over the choice of (pub, pri), the randomness of $\mathcal{T}$ and the coins of $I^{\mathcal{T}}_{pub}$, is non-negligible in n.

   We construct a probabilistic polytime forger F which, given a public key pub and access to a random oracle $\mathcal{R}$ and a signing oracle $\mathcal{S}$, produces a signature $\sigma_m = (\textsc{Cmt}, \textsc{Rsp})$ such that $\Pr[V_{pub}(\textsc{Cmt}, \mathcal{R}(\textsc{Cmt}, m), \textsc{Rsp}) = 1]$ is non-negligible in n, for any message $m \in \{0, 1\}^*$; here the probability is taken over $(pub, pri) \leftarrow G(1^n)$, the randomness of $\mathcal{R}$ and $\mathcal{S}$, and the coins of $F^{\mathcal{R},\mathcal{S}}_{pub,m}$.

   Given a public key pub and a message m to sign, $F^{\mathcal{R},\mathcal{S}}_{pub,m}$ simulates $I^{\mathcal{T}}_{pub}$ as follows: whenever $I^{\mathcal{T}}_{pub}$ queries $\mathcal{T}$, $F^{\mathcal{R},\mathcal{S}}_{pub,m}$ computes $(\textsc{Cmt}', \textsc{Rsp}') = \mathcal{S}(m')$, $\textsc{Ch}' = \mathcal{R}(\textsc{Cmt}', m')$ and gives the transcript $\textsc{Tr} = (\textsc{Cmt}', \textsc{Ch}', \textsc{Rsp}')$ to $I^{\mathcal{T}}_{pub}$. The choice of messages $m' \in \{0,1\}^*$ is immaterial as long as they are distinct and different from m. We need them to be distinct so that $\mathcal{R}$ is queried on a new string every time, ensuring that $\mathcal{R}(\textsc{Cmt}', m')$ is uniformly distributed over $\{0,1\}^{c(n)}$ and $\textsc{Tr}$ is distributed exactly like a random transcript. They should differ from m because F can't win by forging the signatures of messages it queried $\mathcal{S}$ on.

Eventually, $I^{\mathcal{T}}_{pub}$ outputs a committment $\textsc{Cmt}$ and awaits V_{pub}'s challenge. At this point, $F^{\mathcal{R},\mathcal{S}}_{pub,m}$ computes $\textsc{Ch} = \mathcal{R}(\textsc{Cmt}, m)$ and gives $\textsc{Ch}$ to $I^{\mathcal{T}}_{pub}$, which responds with $\textsc{Rsp}$; notice that up till now $\mathcal{R}$ has only been queried on strings of the form $\textsc{Cmt}' \circ m'$, where $m' \neq m$, so $\textsc{Cmt} \circ m$ is a new string and $\mathcal{R}(\textsc{Cmt}, m)$ is distributed uniformly over $\{0,1\}^{c(n)}$. $F^{\mathcal{R},\mathcal{S}}_{pub,m}$ then outputs $\sigma = (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$ as the signature of m.

The probablity that σ is a legitimate signature of m with respect to (pub, pri) is equal to the probability that $I^{\mathcal{T}}_{pub}$ gets V_{pub} to accept, which is non-negligible by assumption. Hence $F^{\mathcal{R},\mathcal{S}}_{pub,m}$ breaks the security of $SIG(ID)$ in the ROM.

Remarks: Notice that the proof works whether ID is trivial or not, since $F^{\mathcal{R},\mathcal{S}}_{pub,m}$ can easily generate properly distributed transcripts by appropriately choosing the messages it queries $\mathcal{S}$ and $\mathcal{R}$ on. This shows that the nontriviality of ID is **not** a necessary condition for the security of $SIG(ID)$ in the ROM, although the authors appear to claim that it is. In other words, we can't argue that ID is non-trivial just because $SIG(ID)$ is secure in the ROM.

Hence it might be the case that the nontriviality assumption used by the authors to prove the converse direction, i.e. that $SIG(ID)$ is secure in the ROM if ID is both passively secure and nontrivial, might be either too strong or unnecessary altogether.

3 Open Questions

- What if ID is actively secure, can we get rid of the nontriviality assumption then? If so, how?

- Let us informally label a canonical id scheme "hypertrivial" if its committment space consists of only a single message, say λ (any fixed $m \in \{0,1\}^*$ will do).

  Although Bellare et al appear to claim that nontriviality of ID is necessary for ROM-security of $SIG(ID)$, i.e. that **every trivial passively secure canonical id scheme yields a signature scheme insecure in the ROM**, all they actually show is that, assuming that factoring certain integers is hard, **there exists a "hypertrivial" passively secure id scheme which yields a signature scheme insecure in the ROM**.

  Might there not exist, under plausible assumptions, a passively secure "hypertrivial" id scheme which *does* yield a ROM-secure signature scheme? All signs point to "yes"; give a concrete example.

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 106. IEEE Computer Society, 2001.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM Press, 1993.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 209–218. ACM Press, 1998.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.