Monero’s Building Blocks
Part 2 of 10 – Pointcheval & Stern’s Generic Signature Scheme [1]

Bassam El Khoury Seguías
BTC: 3FcVvBZwTUkUrecqJdI6RejR42qT2tDWHWn
ETH: 0xb79Fb9194C8Cc6221368b70976e18609Ab9AcA8

March 3, 2018

1 Introduction

In the next few parts of this series, we look at various signature schemes and prove their security in the RO model. This part is dedicated to the analysis of a generic signature scheme introduced in [1], of which the non-interactive Schnorr scheme is an example. The generic scheme is built around a single \((sk, pk)\) pair. Later parts of this series will focus on ring signature schemes. Ring signatures embed the actual signer in a ring of other possible signers to hide her identity. We will discuss them in parts 3, 4, 5, 6, and 7.

2 Pointcheval & Stern’s generic scheme

For a given message \(m\), our generic scheme creates a signature \(\sigma(m) \equiv (r, h, \alpha)\) where \(r\) is a random element chosen from a pre-defined set, \(h = H(m, r)\) (i.e., RO output on query \((m, r)\)), and \(\alpha\) is fully determined by \(m, r,\) and \(h\). By design, we require that the probability of selecting any particular \(r\) be upper-bounded by \(\frac{1}{2^k-1}\) for a given security parameter \(k\).

Schnorr’s signature scheme is an example that fits this generic model. To see why, recall that \(\Sigma_{\text{Schnorr}}\) chooses a random commitment \(k \in \mathbb{Z}_q^*\) where \(q\) is a pre-defined prime number. It then assigns \(r \equiv g^k\) where \(g\) denotes a chosen generator of \(\mathbb{Z}_q^*\). Afterwards, \(h\) is set to \(H(m, r)\). Finally, \(\alpha\) is calculated as \(k - hx\) where \(x\) denotes the signer’s private key. Note that \(r\) can be any element of \(\mathbb{Z}_q^*\) and so the probability that it takes on a specific value is equal to \(\frac{1}{q-1}\). By design, we choose the security parameter \(k \leq \log_2(q)\). This choice of \(k\) guarantees that the aforementioned probability is upper-bounded by \(\frac{1}{2^k-1} \leq \frac{1}{2^{2k-1}}\).

3 Security analysis - Unforgeability vis-a-vis EFACM

For unforgeability proofs, we follow the 5-step approach mentioned in part 1 of this series.
Step 1: To prove that this generic scheme is secure against EFACM in the RO model, we proceed by contradiction and assume that there exists a PPT adversary $A$ such that:

$$P_{\omega,r,H}[A(\omega)^{H,\Sigma_H(r)} \text{ succeeds in EFACM}] = \epsilon(k),$$

for some $\epsilon$ non-negligible in $k$.

Step 2: Next, we build a simulator $S(r')$ such that it:

- Does not have access to the private key of any signer.
- Has the same range as $\Sigma$ (i.e., they output signatures taken from the same pool of potential signatures over all possible choices of RO functions and respective random tapes $r$ and $r'$).
- Has indistinguishable probability distribution from that of $\Sigma$ over this range.

$S(r')$ is specific to the particular instance of the generic scheme being used. In what follows, we build a simulator for the case of a Schnorr signature.

**Original Signer $\Sigma(r)$**

- $x$ (signer’s $sk$)
- $m$ (message)

  Randomly choose $k \in \mathbb{Z}_q^*$
  Assign $r \equiv g^k \in \mathbb{Z}_q^*$
  Send $(m, r)$ to RO
  Receive $h \equiv H(m, r) \in \mathbb{Z}_q$
  Compute $\alpha \equiv k - hx \in \mathbb{Z}_q$

  **output** $(r, h, \alpha)$

  **Verifier** checks if:
  1) $h = H(m, r)$
  2) $r = g^\alpha y^h$

**Simulator $S(r')$ bypasses RO**

- $y$ (signer’s $pk$)
- $m$ (message)

  Randomly choose $k' \in \mathbb{Z}_q$
  Randomly choose $h' \sim \mathbb{Z}_q$
  Assign $r' \equiv g^k y^{h'} \in \mathbb{Z}_q^*$
  Assign $h$ to $H(m, r')$
  (bypass RO)
  Set $\alpha' \equiv k \in \mathbb{Z}_q$

  **output** $(r', h', \alpha')$

  **Verifier** checks if:
  1) $h' = H(m, r')$
  2) $r' = g^\alpha' y^{h'}$

By construction, the output of $S$ will satisfy the verification equations. Moreover, it assigns a random value for $h$ and bypasses the RO in doing so. Next, note the following:

1. $S$ does not use any private key.
2. $\Sigma$ and $S$ both have a range $R \equiv \{(\epsilon, \beta, \gamma) \in (\mathbb{Z}_q^* \times \mathbb{Z}_q \times \mathbb{Z}_q) \text{ s.t. } \epsilon = g^\gamma \times y^\beta\}$. 

2018 Bassam El Khoury Seguias ©
3. \( \Sigma \) and \( S \) have the same probability distribution over \( R \). Indeed, \( \forall (\epsilon, \beta, \gamma) \in R \) we have:

- For \( \Sigma \): \( P[(r, h, \alpha) = (\epsilon, \beta, \gamma)] = P_{k \neq 0, h}[g^k = \epsilon, h = \beta, k - hx = \gamma] = \frac{1}{(q-1)q} \).
- For \( S \): \( P[(r^~, h^~, \alpha^~) = (\epsilon, \beta, \gamma)] = P_{r^~, h^~}[r^~ \equiv g^ky^{h^~} = \epsilon, h^~ = \beta, \alpha^~ \equiv k = \gamma] = \frac{1}{(q-1)q} \).

With \( S \) adequately built for the Schnorr scheme, we conclude that (refer to section 6 of part 1 of this series for a description of collision types):

\[
P_{w,r,h}[A(\omega)H^S(r') \text{ succeeds in EFACM}] = \epsilon(k), \text{ for some } \epsilon \text{ non-negligible in } k.
\]

**Step 3**: We now show that the probability of faulty collisions is negligible (refer to section 6 of part 1 of this series for a justification). The 2 types of collisions for the generic scheme are:

- **ColType 1**: A tuple \((m, r)\) that \( S \) encounters – it makes its own random assignment to \( H(m, r) \) and bypasses RO – also appears in the list of queries that \( A(\omega) \) sends to RO. A conflict in the 2 values will happen with overwhelming probability and the execution will halt.

- **ColType 2**: A tuple \((m, r)\) that \( S \) encounters – it makes its own random assignment to \( H(m, r) \) – is the same as another tuple \((m', r')\) that \( S \) encountered at an earlier time instance – here too, \( S \) would have made its own random assignment to \( H(m', r') \). Since the 2 tuples are identical (i.e., \((m, r) = (m', r')\)), it must be that the 2 random assignments match (i.e., \( H(m, r) = H(m', r') \)). With overwhelming probability, the 2 values will be different and the execution will halt.

The aforementioned collisions must be avoided. In order to do so, we first calculate the probability of their occurrence. We assume that during an EFACM attack, \( A(\omega) \) can make a maximum of \( Q \) queries to RO and a maximum of \( Q_S \) queries to \( S(r') \). \( Q \) and \( Q_S \) are both assumed to be polynomial in the security parameter \( k \), since the adversary is modeled as a PPT Turing machine.

\[
P[\text{ColType 1}] = P[\bigcup_{(m, r) \in \{(m, r) \text{ appeared in at least one of the } Q_S \text{ queries to } S \text{ and } Q \text{ queries to RO}}] 
\leq P[\bigcup_{r \in \mathbb{Z}_q^*} \{r \text{ was part of at least one of the } Q_S \text{ queries to } S \text{ and } Q \text{ queries to RO}}] 
\leq \sum_{r \in \mathbb{Z}_q^*} P[\bigcup_{(j=1, \ldots, Q_S), (k=1, \ldots, Q)} \{r \text{ was part of at least the } j^{th} \text{ query to } S \text{ and } k^{th} \text{ queries to RO}}] 
\leq \sum_{r \in \mathbb{Z}_q^*} \sum_{j=1}^{Q_S} \sum_{k=1}^{Q} P[r \text{ was part of at least the } j^{th} \text{ query to } S \text{ and } k^{th} \text{ queries to RO}}] 
\leq \sum_{r \in \mathbb{Z}_q^*} \sum_{j=1}^{Q_S} \sum_{k=1}^{Q} \left(\frac{1}{q-1}\right)^2 = (q - 1) \times \frac{Q_S Q}{(q-1)^2} = \frac{Q_S Q}{q-1} \leq \frac{Q S Q}{2^{k+1}}
\]

Since \( Q_S \) and \( Q \) are polynomial in \( k \), we conclude that \( P[\text{ColType 1}] \) is negligible in \( k \).

Next, we compute \( P[\text{ColType 2}] \):
\[ P[\text{ColType } 2] = P[\cup_{(m,r)} \{(m,r) \text{ appeared at least twice in some of the queries to } S\}] \]
\[ \leq P[\cup_{r \in \mathbb{Z}_q^*} \{r \text{ was part of at least 2 queries to } S\}] \]
\[ \leq \sum_{r \in \mathbb{Z}_q^*} \left( \frac{Q_s}{2} \right) \times \frac{1}{(q-1)^2} = \left( \frac{Q_s}{2} \right) \times \frac{q^{-1}}{(q-1)^2} = \frac{Q_s(Qs-1)}{2(q-1)} \leq \frac{Q_s^2}{2 \times 2^{q-1}} \]

And so \( P[\text{ColType } 2] \) is also negligible in \( k \).

We then have:

\[ P[\text{Col}] = P[\text{ColType } 1 \cup \text{ColType } 1] \leq P[\text{ColType } 1] + P[\text{ColType } 2] \leq \frac{Q_sQ + Q_s^2}{2^{q-1}} \equiv \delta(k) \]

which is negligible in \( k \). We can finally conclude (as was shown in section 6 of part 1), that:

\[ P_{\omega,r',H}[\mathcal{A}(\omega)H.S(r') succeeds in EFACM \cap \overline{\text{Col}}] \geq \epsilon(k) - \delta(k), \text{ (non-negligible in } k) \]

**Step 4**: In this step, our objective is to show that if \((\omega^*, r^*, H^*)\) is a successful tuple that generated a first EFACM forgery, then the following quantity is non-negligible in \( k \):

\[ P_{H}[\mathcal{A}(\omega^*)H.S(r^*) succeeds in EFACM \cap \overline{\text{Col}} \text{ with } (\rho_{\beta} \neq \rho_0^*), \text{ and } (\rho_i = \rho_i^*) \text{ for } i \in \{1, ..., \beta - 1\}] \]

Here \( \beta \) is an appropriate index that we will define in the proof. And to further simplify the notation, we let \( \rho_i^* \equiv H^*(q_i^*) \) and \( \rho_i \equiv H(q_i) \). \((q_i^*, q_i)\) denote respectively the \( i^{th} \) query to \( H^* \) and \( H \) for all \( i \in 1, ... \beta \).

Let’s take a closer look at \( P_{\omega,r',H}[\mathcal{A}(\omega)H.S(r') succeeds in EFACM \cap \overline{\text{Col}}] \).

Any successful forgery must pass the verification test. One of the verification equations is to check if \( h = H(m,r) \). So we distinguish between 2 scenarios (w.l.o.g. we assume that all \( \mathcal{A} \)-queries sent to \( RO \) are distinct from each-other since \( \mathcal{A} \) can keep a local copy of previous query results and avoid redundant calls):

- **Scenario 1**: \( \mathcal{A} \) was successful in its forgery, and no collisions occurred, and it never queried \( RO \) on input \((m,r)\).

- **Scenario 2**: \( \mathcal{A} \) was successful in its forgery, and no collisions occurred, and it queried \( RO \) on input \((m,r)\) during its execution.

The probability of scenario 1 is upperbounded by the probability that \( \mathcal{A} \) picks a value for \( h \) that matches the value of \( H(m,r) \). Here, \( H(m,r) \) is the value that \( RO \) returns to \( \mathcal{V} \) (the verification algorithm) when verifying the validity of the forged signature. (It is upper-bounded because at the very least, the constraint \( h = H(m,r) \) must be observed for a valid signature). And since \( h \) can be any value in \( \mathbb{Z}_q \), we get:
This implies that \( \exists \) Proof

By definition of the sets \( \{ \cup \) And clearly, \( A \) during execution. We let \( \text{Ind} \) Recall that, \( \text{Ind} \) We then define \( \text{Ind} \) So we introduce the set \( \text{Ind} \) We then define \( \text{Ind} \) So we assume that a successful forgery will likely be of the Scenario 2 type. We have:

\[
P[\text{Scenario } 1] \leq \frac{1}{q} \leq \frac{1}{2^k}, \text{ which is negligible in } k.
\]

So we assume that a successful forgery will likely be of the Scenario 2 type. We have:

\[
P[\text{Scenario } 2] = P_{\omega, r'; H}[A(\omega)^{H,S(r')} \text{succeeds in EFACM } \cap \text{Col}] - P[\text{Scenario } 1]
\]

\[
\geq \varepsilon(k) - \delta(k) - \frac{1}{2^k} \equiv \nu(k), \text{ which is non-negligible in } k
\]

We then define \( \text{Ind}(\omega, r', H) \) to be the index of the query \( (m, r) \) sent by \( A(\omega) \) to RO during execution. We let \( \text{Ind}(\omega, r', H) = \infty \) if the query \( (m, r) \) was never asked by \( A(\omega) \). This definition allows us to build the following sets:

- \( S = \{ (\omega, r', H) | A(\omega)^{H,S(r')} \text{succeeds in EFACM } \cap \text{Col} \cap \text{Ind}(\omega, r', H) \neq \infty \} \)

In other terms, \( S \) is the set of tuples \( (\omega, r', H) \) that yield a successful EFACM forgery when no collisions occur, and when \( A(\omega) \) queried RO on input \( (m, r) \) at some point during its execution (i.e., scenario 2).

- \( S_i = \{ (\omega, r', H) | A(\omega)^{H,S(r')} \text{succeeds in EFACM } \cap \text{Col} \cap \text{Ind}(\omega, r', H) = i \} \)

In other terms, \( S_i \) is the set of tuples \( (\omega, r', H) \) that yield a successful EFACM forgery when no collisions occur, and when the index of the \( A(\omega) \)-query on input \( (m, r) \) sent to RO is equal to \( i \).

Recall that, \( P_{\omega, r'; H}[(\omega, r', H) \in S] = P[\text{Scenario } 2] \geq \nu(k), \) which is non-negligible in \( k \).

And clearly, \( \{ \cup_{i=1}^{Q} S_i \} \) partitions \( S \). So \( \sum_{i=1}^{Q} P[(\omega, r', H) \in S_i | (\omega, r', H) \in S] = 1 \).

This implies that \( \exists i \in \{ 1, ...Q \} \) s.t. \( P[(\omega, r', H) \in S_i | (\omega, r', H) \in S] \geq \frac{1}{2^Q} \).

If this were not the case, then one would get the following contradiction:

\[
1 = \sum_{i=1}^{Q} P[(\omega, r', H) \in S_i | (\omega, r', H) \in S] < Q \times \frac{1}{2^Q} = \frac{1}{2} < 1.
\]

So we introduce the set \( I \) consisting of all indices that meet the \( \frac{1}{2^Q} \) threshold, i.e.

\[
I = \{ i \in \{ 1, ...Q \} | P[(\omega, r', H) \in S_i | (\omega, r', H) \in S] \geq \frac{1}{2^Q} \}
\]

We claim that \( P[\text{Ind}(\omega, r', H) \in I | (\omega, r', H) \in S] \geq \frac{1}{2} \).

**Proof** By definition of the sets \( S_i \) we have:

\[
P[\text{Ind}(\omega, r', H) \in I | (\omega, r', H) \in S] = \sum_{i \in I} P[(\omega, r', H) \in S_i | (\omega, r', H) \in S]
\]

\[
= 1 - \sum_{j \notin I} P[(\omega, r', H) \in S_j | (\omega, r', H) \in S] > 1 - \sum_{j \notin I} \frac{1}{2^Q} > 1 - \frac{Q}{2^Q} = \frac{1}{2}
\]
The next step is to apply the splitting lemma to each \( S_i \), \( i \in I \). First note that:

\[
P_{\omega,r',\mathcal{H}}((\omega, r', \mathcal{H}) \in S_i) = P((\omega, r', \mathcal{H}) \in S_i \mid (\omega, r', \mathcal{H}) \in S) \times P_{\omega,r',\mathcal{H}}((\omega, r', \mathcal{H}) \in S) \\
\geq \frac{1}{2Q} \times \nu(k)
\]

Referring to the notation used in the splitting lemma (section 7 of part 1), we let:

\[
A \equiv S_i, \quad X \equiv (\omega, r', \rho_1, \ldots, \rho_{i-1}), \quad Y \equiv (\rho_i, \ldots, \rho_Q), \quad \epsilon \equiv \frac{\nu(k)}{2Q}, \quad \alpha \equiv \frac{\nu(k)}{4Q} = \frac{\epsilon}{2}
\]

\( X \) is defined as the space of tuples of all random tapes \( \omega \), all random tapes \( r' \), and all possible RO answers to the first \( i - 1 \) queries sent by \( A(\omega) \). \( Y \) is defined as the space of all possible RO answers to the last \( (Q - i + 1) \) queries sent by \( A(\omega) \). (Recall that \( \rho_i \equiv \mathcal{H}(q_i) \)). The splitting lemma guarantees the existence of a subset \( \Omega_i \) of tuples \( (\omega, r', \mathcal{H}) \) such that:

\begin{itemize}
  \item \( P_{\omega,r'\mathcal{H}}((\omega, r', \mathcal{H}) \in \Omega_i) \geq \frac{\nu(k)}{4Q} \)
  \item \( \forall[(\omega^*, r^{\omega*}, \mathcal{H}^{\omega*}) \equiv (\omega^*, r^{\omega*}, \rho_1^{\omega*}, ..., \rho_{i-1}^{\omega*}, \rho_i^{\omega*} ... \rho_Q)] \in \Omega_i, \) we have

\[
P_{\mathcal{H}}((\omega^*, r^{\omega*}, \rho_1^{\omega*}, ..., \rho_{i-1}^{\omega*}, \rho_i ... \rho_Q) \in S_i \mid (\omega^*, r^{\omega*}, \mathcal{H}^{\omega*}) \in \Omega_i) \geq \frac{\nu(k)}{4Q}, \text{ and so}
\]

\[
P_{\mathcal{H}}((\omega^*, r^{\omega*}, \mathcal{H}) \in S_i \mid (\omega^*, r^{\omega*}, \mathcal{H}^{\omega*}) \in \Omega_i, \rho_1 = \rho_1^{\omega*}, ..., \rho_{i-1} = \rho_{i-1}^{\omega*}) \geq \frac{\nu(k)}{4Q}
\]

\item \( P((\omega, r', \mathcal{H}) \in \Omega_i \mid (\omega, r', \mathcal{H}) \in S_i) \geq \frac{(\nu(k))}{(4Q)} = \frac{1}{2}
\]
\end{itemize}

We would like to compute the probability of finding a 2\textsuperscript{nd} successful tuple \((\omega^*, r^{\omega*}, \mathcal{H}^{\omega*})\) given that \((\omega^*, r^{\omega*}, \mathcal{H}^{\omega*})\) was a successful 1\textsuperscript{st} tuple and s.t. \( \rho_j^{\omega*} = \rho_j^*, \ j = 1, \ldots, i - 1 \). That means finding the following probability:

\[
P_{\mathcal{H}}((\omega^*, r^{\omega*}, \mathcal{H}) \in S_i \mid (\omega^*, r^{\omega*}, \mathcal{H}^*) \in S_i, \rho_1 = \rho_1^*, ..., \rho_{i-1} = \rho_{i-1}^*)
\]

From the splitting lemma results, we have a (non-negligible in \( k \)) lower-bound on \( P_{\mathcal{H}}((\omega^*, r^{\omega*}, \mathcal{H}) \in S_i \mid (\omega^*, r^{\omega*}, \mathcal{H}^*) \in \Omega_i, \rho_1 = \rho_1^*, ..., \rho_{i-1} = \rho_{i-1}^*) \).

Note however, that \( \Omega_i \) and \( S_i \) are generally distinct sets. And so we cannot conclude that

\[
P_{\mathcal{H}}((\omega^*, r^{\omega*}, \mathcal{H}) \in S_i \mid (\omega^*, r^{\omega*}, \mathcal{H}^*) \in S_i, \rho_1 = \rho_1^*, ..., \rho_{i-1} = \rho_{i-1}^*)
\]

\[
= P_{\mathcal{H}}((\omega^*, r^{\omega*}, \mathcal{H}) \in S_i \mid (\omega^*, r^{\omega*}, \mathcal{H}^*) \in \Omega_i, \rho_1 = \rho_1^*, ..., \rho_{i-1} = \rho_{i-1}^*)
\]
and therefore we **cannot** conclude that the following quantity is non-negligible in $k$

$$P_H[(ω^*, r^{*s}, H) ∈ S_i | (ω^*, r^{*s}, H^*) ∈ S_i, ρ_1 = ρ_1^*, ..., ρ_{i-1} = ρ_{i-1}^*]$$

In order to show that the above quantity is non-negligible in $k$, we proceed differently. Suppose we can show that the following probability is non-negligible in $k$:

$$P(ω, r', H)[∃β ∈ I \text{ s.t. } (ω, r', H) ∈ (Ω_β ∩ S_β)]$$

This would imply that with non-negligible probability, we can find a tuple that belongs to $S_β$ (and hence corresponds to a successful forgery) and at the same time belongs to $Ω_β$. We can then invoke the splitting lemma result just mentioned, to find a second tuple corresponding to a second forgery and that has the desired properties.

To prove the above, we proceed as follows:

$$P[∃β ∈ I \text{ s.t. } (ω, r', H) ∈ (Ω_β ∩ S_β) | (ω, r', H) ∈ S] = P[∪_{i ∈ I} \{(ω, r', H) ∈ (Ω_i ∩ S_i) | (ω, r', H) ∈ S\}]$$

$$= \sum_{i ∈ I} P[(ω, r', H) ∈ (Ω_i ∩ S_i) | (ω, r', H) ∈ S], \text{ since the } S_i\text{'s are disjoint.}$$

$$\geq \frac{1}{2} \sum_{i ∈ I} P[(ω, r', H) ∈ S_i | (ω, r', H) ∈ S], \text{ (3rd result of splitting lemma above)}$$

$$\geq \frac{1}{2} \times \frac{1}{2} \text{ (by the claim proven earlier)} = \frac{1}{4}.$$ 

And so we conclude that:

$$P(ω, r', H)[∃β ∈ I \text{ s.t. } (ω, r', H) ∈ (Ω_β ∩ S_β)]$$

$$= P[∃β ∈ I \text{ s.t. } (ω, r', H) ∈ (Ω_β ∩ S_β) | (ω, r', H) ∈ S] \times P(ω, r', H)[(ω, r', H) ∈ S]$$

$$\geq \frac{ν(k)}{4}, \text{ which is non-negligible in } k.$$ 

So let $β$ be such an index and $(ω^*, r^{*s}, H^*)$ such a tuple. From the result above, we know that finding such a $(ω^*, r^{*s}, H^*) ∈ (Ω_β ∩ S_β)$ can be done with non-negligible probability. And since $(Ω_β ∩ S_β) ⊂ Ω_β$, we must have $(ω^*, r^{*s}, H^*) ∈ Ω_β$. We can then invoke the $2^{nd}$ consequence of the splitting lemma, and write:

$$P_H[(ω^*, r^{*s}, H) ∈ S_β | (ω^*, r^{*s}, H^*) ∈ S_β, ρ_1 = ρ_1^*, ..., ρ_{β-1} = ρ_{β-1}^*] =$$

$$P_H[(ω^*, r^{*s}, H) ∈ S_β | (ω^*, r^{*s}, H^*) ∈ Ω_β, ρ_1 = ρ_1^*, ..., ρ_{β-1} = ρ_{β-1}^*] ≥ \frac{ν(k)}{4Q}$$

We still have one last constraint to impose and that is that $ρ_β^* ≠ ρ_β$. We show that the
following quantity is non-negligible:

\[ P_H[\{(\omega^*, r^{*\omega}, \mathcal{H}) \in S_\beta \} \cap (\rho_\beta \neq \rho_\beta^*) | (\omega^*, r^{*\omega}, \mathcal{H}^*) \in S_\beta, \; \rho_1 = \rho_1^*, ..., \rho_{\beta-1} = \rho_{\beta-1}^*] \]

To prove this, note that if \( B \) and \( C \) are independent events, then we can write:


And so we get \( P[A \cap B|C] \geq P[A|C] - P[B]. \) This results allows us to write:

\[ P_H[\{(\omega^*, r^{*\omega}, \mathcal{H}) \in S_\beta \} \cap (\rho_\beta \neq \rho_\beta^*) | (\omega^*, r^{*\omega}, \mathcal{H}^*) \in S_\beta, \; \rho_1 = \rho_1^*, ..., \rho_{\beta-1} = \rho_{\beta-1}^*] \]
\[ \geq P_H[\{(\omega^*, r^{*\omega}, \mathcal{H}) \in S_\beta | (\omega^*, r^{*\omega}, \mathcal{H}^*) \in S_\beta, \; \rho_1 = \rho_1^*, ..., \rho_{\beta-1} = \rho_{\beta-1}^*] - P_H[\rho_\beta = \rho_\beta^*] \]
\[ = P_H[\{(\omega^*, r^{*\omega}, \mathcal{H}) \in S_\beta | (\omega^*, r^{*\omega}, \mathcal{H}^*) \in \Omega_\beta, \; \rho_1 = \rho_1^*, ..., \rho_{\beta-1} = \rho_{\beta-1}^*] - P_H[\rho_\beta = \rho_\beta^*] \]
\[ (\because \text{we chose } (\omega^*, r^{*\omega}, \mathcal{H}^*) \in \Omega_\beta \cap S_\beta) \]
\[ \geq \frac{\omega(k)}{4^k} - \frac{1}{2^k}, \text{ which is non-negligible in } k. \]

**Step 5**: The final step uses the 2 forgeries obtained earlier to solve an instance of the Discrete Logarithm (DL) problem. Here is a recap of Step 4 results:

- With non-negligible probability of at least \( \frac{\omega(k)}{4^k} \) we get a successful tuple \((\omega^*, r^{*\omega}, \mathcal{H}^*)\), s.t. \((\omega^*, r^{*\omega}, \mathcal{H}^*) \in (\Omega_\beta \cap S_\beta)\) for some index \( \beta \in I \). So by running \( \mathcal{A} \) a number of times polynomial in \( k \), we can confidently find such a tuple.

- Once we find such a tuple, we’ve also shown that with non-negligible probability of at least \( \frac{\omega(k)}{4^k} - \frac{1}{2^k} \), we can find another successful tuple \((\omega^*, r^{*\omega}, \mathcal{H}^*)\) such that \((\omega^*, r^{*\omega}, \mathcal{H}^*) \in S_\beta\) and \((\gamma_1 = \rho_1^*), ..., (\gamma_{\beta-1} = \rho_{\beta-1}^*), \) but \((\rho_\beta \neq \rho_\beta^*)\).

W.l.o.g, let \((\omega^*, r^{*\omega}, \mathcal{H}^*)\) correspond to \( \sigma_{\text{forge}}(m_1) \equiv (r_1, h_1, \alpha_1), \) and \((\omega^*, r^{*\omega}, \mathcal{H}^*)\) correspond to \( \sigma_{\text{forge}}(m_2) \equiv (r_2, h_2, \alpha_2). \)

Recall that \( \beta \) is the index of the query \((m_1, r_1)\) that \( \mathcal{A} \) sends to the RO. Since the 2 experiments corresponding to the 2 successful tuples have the same random tapes \( \omega^* \) and \( r^{*\omega} \), and since the 2 corresponding ROs \( \mathcal{H}^* \) and \( \mathcal{H}^\sim \) behave the same way on the first \( \beta - 1 \) queries, we can be confident that the first \( \beta \) queries sent to the 2 ROs are identical. In particular the two \( \beta^{th} \) queries are the same (i.e., \((m_1, r_1) = (m_2, r_2)\)).

Moreover by design, \( h_1 = \mathcal{H}^\sim(m_1, r_1) = \rho_\beta^* \neq \rho_\beta^\sim = \mathcal{H}^\sim(m_1, r_1) = h_2. \)

So we have 2 successful forgeries \( \sigma_{\text{forge}}(m_1) \equiv (r_1, h_1, \alpha_1) \) and \( \sigma_{\text{forge}}(m_1) \equiv (r_1, h_2, \alpha_2) \), with \( h_1 \neq h_2. \) Since both are valid signatures, they must satisfy the verification equations. For the particular case of a Schnorr signature scheme, they must satisfy the following 2 equations (1 verification equation per signature):

- \( r_1 = g^{\alpha_1} y^{h_1}, \) where \( y \) is the public key of the signer whose signature \( \mathcal{A} \) is forging.
- \( r_1 = g^{\alpha_2} y^{h_2}, \) where \( y \) is the public key of the signer whose signature \( \mathcal{A} \) is forging.
Writing \( y = g^x \) (\( x \) is the secret key of the signer whose signature \( A \) is forging), we get:

\[
g^{\alpha_1 + x h_1} = g^{\alpha_2 + x h_2} \iff \alpha_1 + x h_1 = \alpha_2 + x h_2 \rightarrow x = \frac{\alpha_1 - \alpha_2}{h_2 - h_1}.
\]

Since, \( h_1 \neq h_2 \), we can solve for \( x \) (the DL of \( y \)) in polynomial time. This contradicts the intractability of DL on multiplicative cyclic groups and we conclude that our signature scheme (in this case the Schnorr’s scheme) is secure against EFACM in the RO model.

References