A Closer Look at Multiple Forking: Leveraging (In)dependence for a Tighter Bound

Sanjit Chatterjee and Chethan Kamath

Indian Institute of Science, Bangalore

November 3, 2013

Table of contents

BACKGROUND

Schnorr Signature: Features

- Derived from Schnorr identification through FS Transform
- Uses one hash function
- Security:
  - Based on the *discrete-log* assumption
  - Hash function modelled as a *random oracle* (RO)
  - Security argued using (random) oracle replay attacks

Schnorr Signature: Construction

The Setting:

1. We work in group $\mathbb{G} = \langle g \rangle$ of prime order p.
2. A hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p$ is used.

Key Generation:

1. Select $z \in_R \mathbb{Z}_p$ as the sk
2. Set $Z := g^z$ as the pk

Signing:

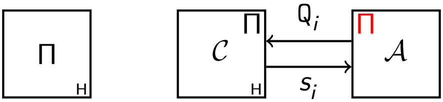
1. Select $r \in_R \mathbb{Z}_p$, set $R := g^r$ and $c := \mathsf{H}(m, R)$.
2. The signature on m is $\sigma := (y, R)$ where $y := r + zc$

Verification:

1. Let $\sigma = (y, R)$ and $c = \mathsf{H}(m, R)$.
2. σ is valid if $g^y = RZ^c$

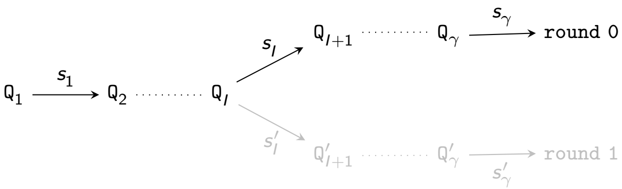
The Oracle Replay Attack

- Random oracle H – i^{th} RO query $\mathtt{Q}_i$ replied with s_i.



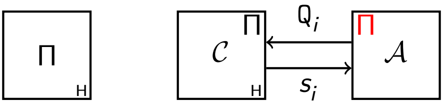
Adversary re-wound to $\mathtt{Q}_I$

Simulation in `round 1` from $\mathtt{Q}_I$ using a *different* random function



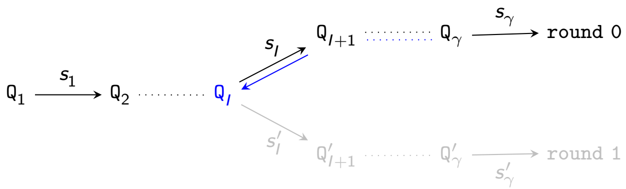
The Oracle Replay Attack

- Random oracle H – i^{th} RO query $\mathtt{Q}_i$ replied with s_i.



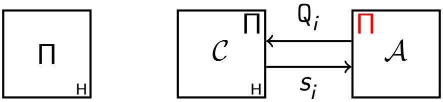
1. Adversary re-wound to $\mathtt{Q}_I$

   Simulation in `round 1` from $\mathtt{Q}_I$ using a *different* random function

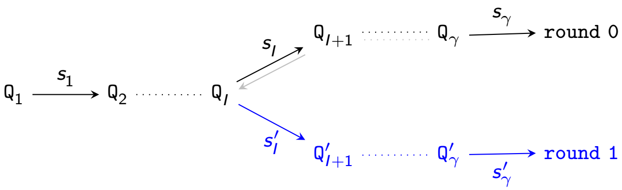


The Oracle Replay Attack

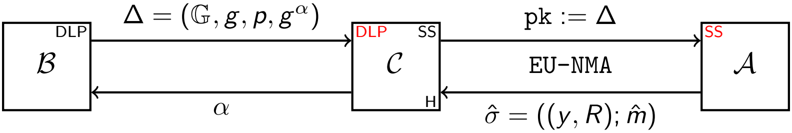
- Random oracle H – i^{th} RO query $\mathtt{Q}_i$ replied with s_i.

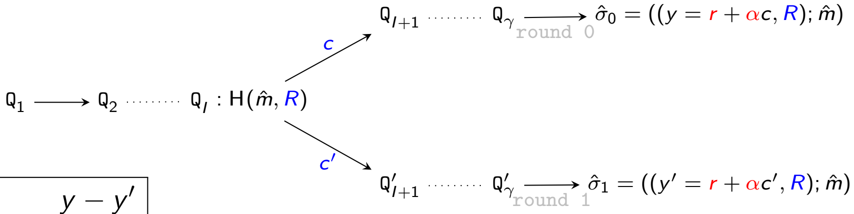


1. Adversary re-wound to $\mathtt{Q}_I$
2. Simulation in `round 1` from $\mathtt{Q}_I$ using a *different* random function



Security of Schnorr Signature, In Brief





$$\alpha = \frac{y - y'}{c - c'}$$

Cost of Oracle Replay Attack

The Forking Lemma [PS00] gives a bound on the success probability of the oracle replay attack in terms of

1. success probability of the adversary (ϵ)
2. bound on RO queries (q)

$$\text{DLP} \;\leq_{O(q/\epsilon^2)}\; \text{Schnorr Signature}$$

Cost of Oracle Replay Attack

The Forking Lemma [PS00] gives a bound on the success probability of the oracle replay attack in terms of

1. success probability of the adversary (ϵ)
2. bound on RO queries (q)

$$\text{DLP} \ \leq_{\mathrm{O}(q/\epsilon^2)} \ \text{Schnorr Signature}$$

The cost: security *degrades* by $\mathrm{O}(q)$

- More or less optimal [Seu12]

General Forking Lemma

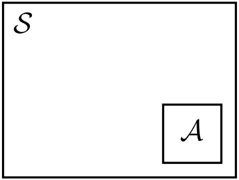
"Forking Lemma is something purely probabilistic, not about signatures" [BN06]

- Abstract version of the Forking Lemma
- Separates out details of simulation (of adversary) from analysis
- A wrapper algorithm used as *intermediary*
    - Simulate the protocol environment to $\mathcal{A}$
    - Simulate the RO as specified by $\mathcal{S}$

General Forking Lemma

"Forking Lemma is something purely probabilistic, not about signatures" [BN06]

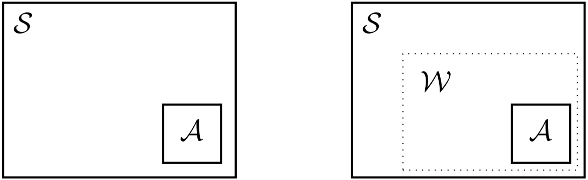
- Abstract version of the Forking Lemma
- Separates out details of simulation (of adversary) from analysis
- A wrapper algorithm used as *intermediary*
    - Simulate the protocol environment to $\mathcal{A}$
    - Simulate the RO as specified by $\mathcal{S}$



General Forking Lemma

"Forking Lemma is something purely probabilistic, not about signatures" [BN06]

- Abstract version of the Forking Lemma
- Separates out details of simulation (of adversary) from analysis
- A wrapper algorithm used as *intermediary*
  - Simulate the protocol environment to $\mathcal{A}$
  - Simulate the RO as specified by $\mathcal{S}$



- Structure of a wrapper call:

$$(I, \sigma) \leftarrow \mathcal{W}(x, s_1, \ldots, s_q; \rho)$$

...General Forking Lemma...

General-Forking Algorithm $\mathcal{F}_{\mathcal{W}}(x)$

Pick coins ρ for $\mathcal{W}$ at random

$\{s_1, \ldots, s_q\} \in_R \mathbb{S}$; $(I, \sigma) \leftarrow \mathcal{W}(x, s_1, \ldots, s_q; \rho)$ //round 0

if $(I = 0)$ **then return** $(0, \perp, \perp)$

$\{s'_{I_0}, \ldots, s'_q\} \in_R \mathbb{S}$; $(I', \sigma') \leftarrow \mathcal{W}(x, s_1, \ldots, s_{I-1}, s'_I, \ldots, s'_q; \rho)$ //round 1

if $(I' = I \wedge s'_I \neq s_I)$ **then return** $(1, \sigma, \sigma')$

else return $(0, \perp, \perp)$

...General Forking Lemma...

General-Forking Algorithm $\mathcal{F}_{\mathcal{W}}(x)$

Pick coins ρ for $\mathcal{W}$ at random

$\{s_1, \ldots, s_q\} \in_R \mathbb{S}$; $(I, \sigma) \leftarrow \mathcal{W}(x, s_1, \ldots, s_q; \rho)$ //round 0

if $(I = 0)$ **then return** $(0, \perp, \perp)$

$\{s'_{I_0}, \ldots, s'_q\} \in_R \mathbb{S}$; $(I', \sigma') \leftarrow \mathcal{W}(x, s_1, \ldots, s_{I-1}, s'_I, \ldots, s'_q; \rho)$ //round 1

if $(I' = I \wedge s'_I \neq s_I)$ **then return** $(1, \sigma, \sigma')$

else return $(0, \perp, \perp)$

The General Forking Lemma gives a bound on the success probability of the oracle replay attack (frk) in terms of

1. success probability of $\mathcal{W}$ (acc)
2. bound on RO queries (q)

$$frk \geq acc^2/q$$

MULTIPLE FORKING

Overview

- Introduced by Boldyreva *et al.* [BPW12]
- Motivation:
  - General Forking restricted to *one* RO and single replay attack
  - Multiple Forking considers two ROs and multiple replay attacks

Overview

- Introduced by Boldyreva *et al.* [BPW12]
- Motivation:
  - General Forking restricted to *one* RO and single replay attack
  - Multiple Forking considers two ROs and multiple replay attacks

- Used originally to argue security of a DL-based proxy signature scheme
- Used further in
  1. Galindo-Garcia IBS [GG09]
  2. Chow *et al.* Zero-Knowledge Argument [CMW12]

GALINDO-GARCIA IBS

Galindo-Garcia IBS: Features

- Derived from Schnorr signature scheme – *nesting*
  - Based on the *discrete-log* (DL) assumption
- Efficient, simple and *does not* use pairing
- Uses two hash functions
- Security argued using nested replay attacks

Galindo-Garcia IBS: Construction

Setting:

1. We work in a group $\mathbb{G} = \langle g \rangle$ of prime order p.
2. Two hash functions $\mathsf{H}, \mathsf{G} : \{0,1\}^* \to \mathbb{Z}_p$ are used.

Set-up:

1. Select $z \in_R \mathbb{Z}_p$ as the $\mathtt{msk}$; set $Z := g^z$ as the $\mathtt{mpk}$

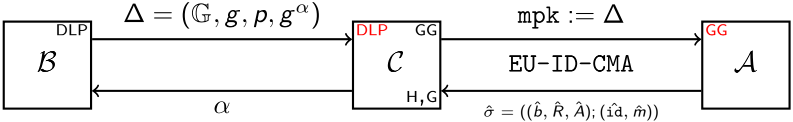
Key Extraction:

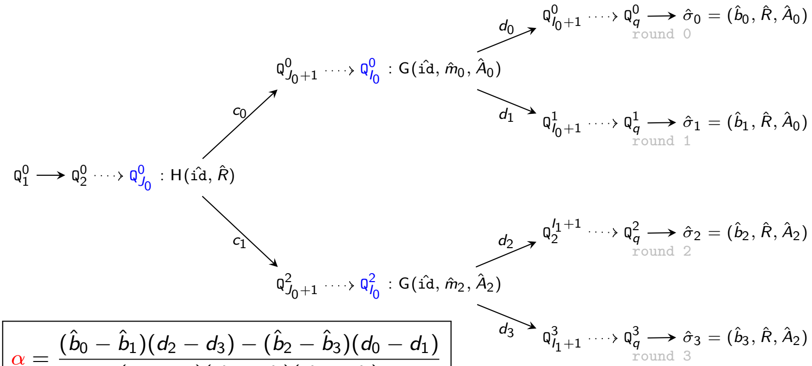
1. Select $r \in_R \mathbb{Z}_p$ and set $R := g^r$.
2. Return $\mathtt{usk} := (y, R)$ as the $\mathtt{usk}$, where $y := r + zc$ and $c := \mathsf{H}(\mathtt{id}, R)$.

Signing:

1. Select $a \in_R \mathbb{Z}_p$ and set $A := g^a$.
2. Return $\sigma := (b, R, A)$ as the signature, where $b := a + yd$ and $d := \mathsf{G}(\mathtt{id}, m, A)$.

Security, In Brief/The Nested Replay Attack





$$\alpha = \frac{(\hat{b}_0 - \hat{b}_1)(d_2 - d_3) - (\hat{b}_2 - \hat{b}_3)(d_0 - d_1)}{(c_0 - c_1)(d_0 - d_1)(d_2 - d_3)}$$

Extending General Forking: Multiple-Forking

Multiple-Forking Algorithm $\mathcal{M}_{\mathcal{W},3}$

Pick coins ρ for $\mathcal{W}$ at random

$\{s_1^0, \ldots, s_q^0\} \in_R \mathbb{S}$;
$(I_0, J_0, \sigma_0) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_q^0; \rho)$ //round 0
if $((I_0 = 0) \vee (J_0 = 0))$ **then return** $(0, \perp)$

$\{s_{I_0}^1, \ldots, s_q^1\} \in_R \mathbb{S}$;
$(I_1, J_1, \sigma_1) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_{I_0-1}^0, s_{I_0}^1, \ldots, s_q^1; \rho)$ //round 1
if $\left((I_1, J_1) \neq (I_0, J_0) \vee (s_{I_0}^1 = s_{I_0}^0)\right)$ **then return** $(0, \perp)$

$\{s_{J_0}^2, \ldots, s_q^2\} \in_R \mathbb{S}$;
$(I_2, J_2, \sigma_2) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_{J_0-1}^0, s_{J_0}^2, \ldots, s_q^2; \rho)$ //round 2
if $\left((I_2, J_2) \neq (I_0, J_0) \vee (s_{J_0}^2 = s_{J_0}^1)\right)$ **then return** $(0, \perp)$

$\{s_{I_2}^3, \ldots, s_q^3\} \in_R \mathbb{S}$;
$(I_3, J_3, \sigma_3) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_{J_0-1}^0, s_{J_0}^2, \ldots, s_{I_2-1}^2, s_{I_2}^3, \ldots, s_q^3; \rho)$ //round 3
if $\left((I_3, J_3) \neq (I_0, J_0) \vee (s_{I_0}^3 = s_{I_0}^2)\right)$ **then return** $(0, \perp)$

return $(1, \{\sigma_0, \ldots, \sigma_3\})$

Multiple-Forking Lemma

The Multiple-Forking Lemma gives a bound on the success probability of the nested replay attack ($mfrk$) in terms of

1. success probability of $\mathcal{W}$ (acc)
2. bound on RO queries (q)
3. number of rounds of forking (n)

$$mfrk \geq acc^{n+1}/q^{2n}$$

Follows from condition: $\mathrm{F} : (I_n, J_n) = (I_{n-1}, J_{n-1}) = \ldots = (I_0, J_0)$

Degradation: $\mathrm{O}\left(q^{2n}\right)$

Multiple-Forking Lemma

The Multiple-Forking Lemma gives a bound on the success probability of the nested replay attack (*mfrk*) in terms of

1. success probability of $\mathcal{W}$ (*acc*)
2. bound on RO queries (q)
3. number of rounds of forking (n)

$$mfrk \geq acc^{n+1}/q^{2n}$$

Follows from condition: $\mathsf{F} : (I_n, J_n) = (I_{n-1}, J_{n-1}) = \ldots = (I_0, J_0)$

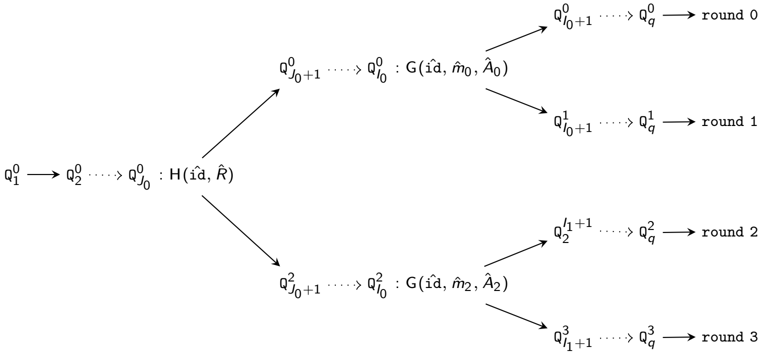
Degradation: $\mathrm{O}\left(q^{2n}\right)$

- Can we improve?

IMPROVING ON MULTIPLE FORKING

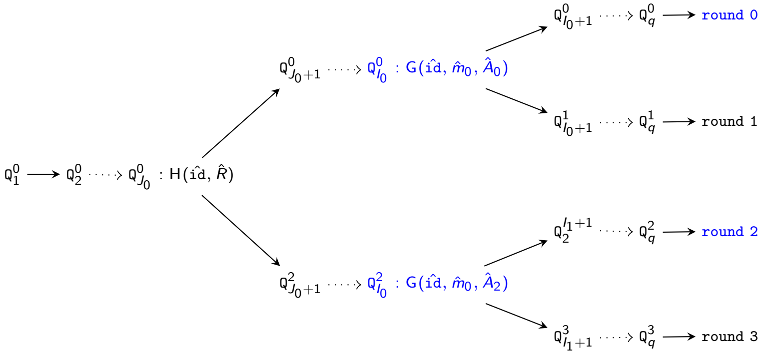
The Intuition

- Recall, condition F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$



The Intuition

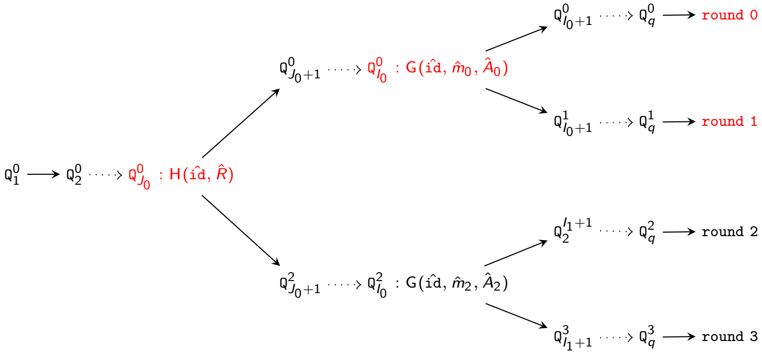
- Recall, condition F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$



- Observations:
  1. *Independency* condition O_1: I_2 *need not* equal I_0

The Intuition

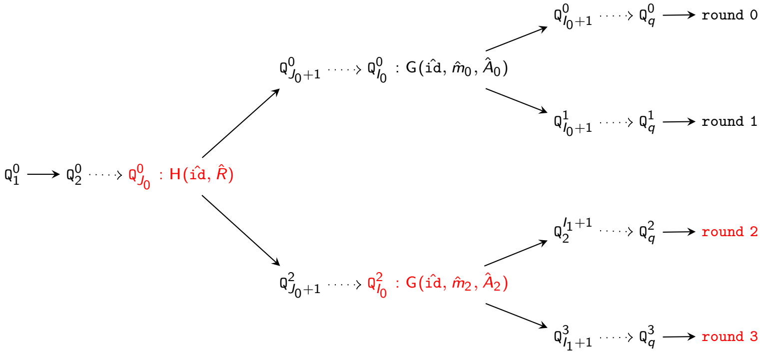
- Recall, condition F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$



- Observations:
  1. *Independency* condition O_1: I_2 *need not* equal I_0
  2. *Dependency* condition O_2: $(I_1 = I_0)$ can *imply* $(J_1 = J_0)$

The Intuition

- Recall, condition F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$



- Observations:
  1. *Independency* condition O_1: I_2 *need not* equal I_0
  2. *Dependency* condition O_2: $(I_1 = I_0)$ can *imply* $(J_1 = J_0)$ (similarly $(I_3 = I_2)$ can *imply* $(J_3 = J_2)$)

...The Intuition...

Effect of $\mathtt{O}_1$ and $\mathtt{O}_2$ on F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$

- $\mathtt{O}_1$: I_2 *need not* equal I_0

$$(I_3, J_3) = (I_2, J_2) \wedge (J_2 = J_0) \wedge (I_1, J_1) = (I_0, J_0)$$

- $\mathtt{O}_2$: $(I_1 = I_0) \implies (J_1 = J_0)$ and $(I_3 = I_2) \implies (J_3 = J_2)$

$$(I_3 = I_2 = I_1 = I_0) \wedge (J_2 = J_0)$$

...The Intuition...

Effect of $\mathtt{O}_1$ and $\mathtt{O}_2$ on F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$

- $\mathtt{O}_1$: I_2 *need not* equal I_0

$$(I_3, J_3) = (I_2, J_2) \wedge (J_2 = J_0) \wedge (I_1, J_1) = (I_0, J_0)$$

- $\mathtt{O}_2$: $(I_1 = I_0) \implies (J_1 = J_0)$ and $(I_3 = I_2) \implies (J_3 = J_2)$

$$(I_3 = I_2 = I_1 = I_0) \wedge (J_2 = J_0)$$

- Together, $\mathtt{O}_1$&$\mathtt{O}_2$:

$$(I_3 = I_2) \wedge (I_1 = I_0) \wedge (J_2 = J_0)$$

...The Intuition...

Effect of $\mathtt{O}_1$ and $\mathtt{O}_2$ on F : $(I_3, J_3) = (I_2, J_2) = (I_1, J_1) = (I_0, J_0)$

- $\mathtt{O}_1$: I_2 *need not* equal I_0

$$(I_3, J_3) = (I_2, J_2) \wedge (J_2 = J_0) \wedge (I_1, J_1) = (I_0, J_0)$$

- $\mathtt{O}_2$: $(I_1 = I_0) \implies (J_1 = J_0)$ and $(I_3 = I_2) \implies (J_3 = J_2)$

$$(I_3 = I_2 = I_1 = I_0) \wedge (J_2 = J_0)$$

- Together, $\mathtt{O}_1$&$\mathtt{O}_2$:

$$(I_3 = I_2) \wedge (I_1 = I_0) \wedge (J_2 = J_0)$$

Intuitively, degradation reduced to $O\left(q^3\right)$

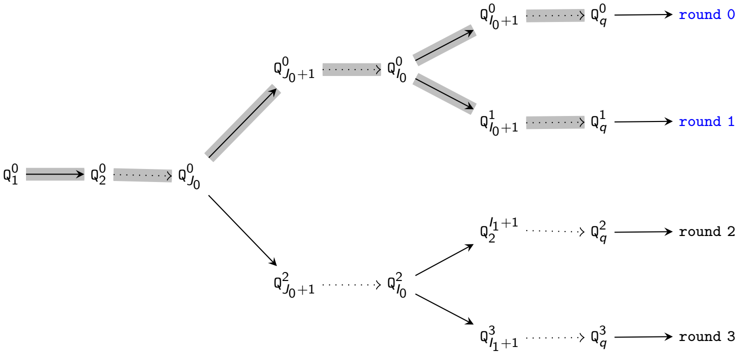
- In general, degradation reduced to $O(q^n)$

MORE ON (IN)DEPENDENCY

The Conceptual Wrapper

- Observations *better* formulated using a conceptual wrapper
  - Clubs two (consecutive) executions of the original wrapper
  - Denoted by $\mathcal{Z}$

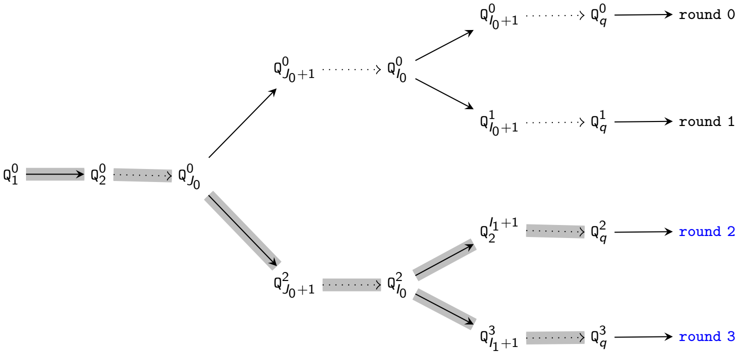
$$(I_k, J_k, \sigma_k), (I_{k+1}, J_{k+1}, \sigma_{k+1})) \leftarrow \mathcal{Z}\left(x, \mathtt{S}^k, \mathtt{S}^{k+1}; \rho\right)$$



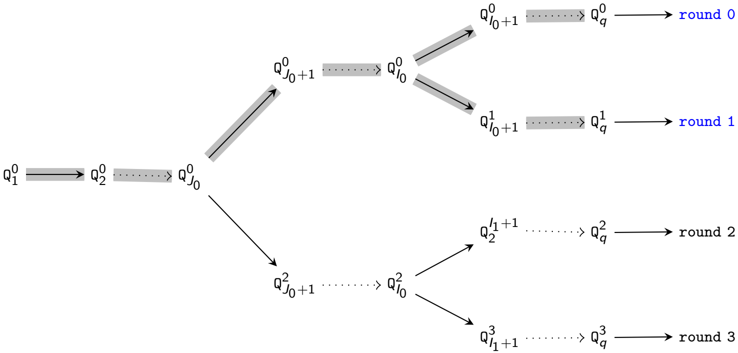
The Conceptual Wrapper

- Observations *better* formulated using a conceptual wrapper
  - Clubs two (consecutive) executions of the original wrapper
  - Denoted by $\mathcal{Z}$

$$(I_k, J_k, \sigma_k), (I_{k+1}, J_{k+1}, \sigma_{k+1})) \leftarrow \mathcal{Z}\left(x, \mathtt{S}^k, \mathtt{S}^{k+1}; \rho\right)$$

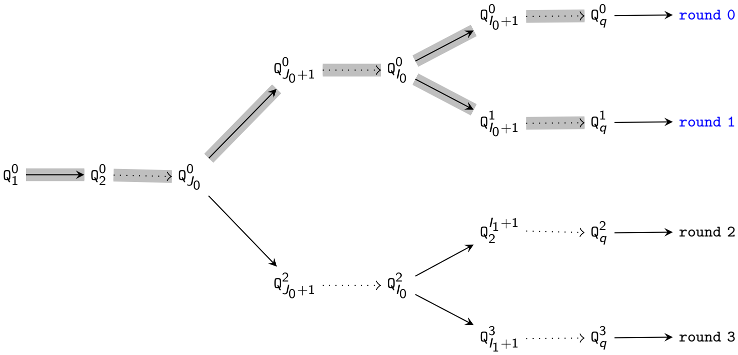


Index Independency



- It is not necessary for the I indices across $\mathcal{Z}$ to be the same
  - I_k need not be equal to $I_{k-2}, I_{k-4}, \ldots, I_0$ for $k = 2, 4, \ldots, n-1$

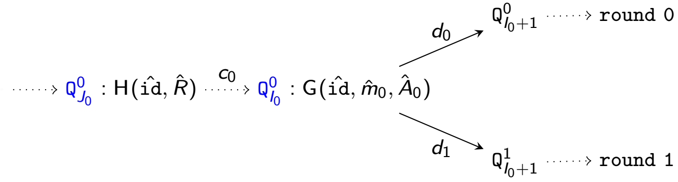
Random-Oracle Dependency



- It is possible to design protocols such that, for the k^{th} invocation of $\mathcal{Z}$, $(I_{k+1} = I_k) \implies (J_{k+1} = J_k)$.

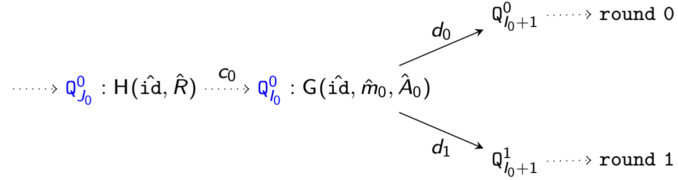
Inducing Random-Oracle Dependency

- Consider `round 0` and `round 1` of simulation for GG-IBS



Inducing Random-Oracle Dependency

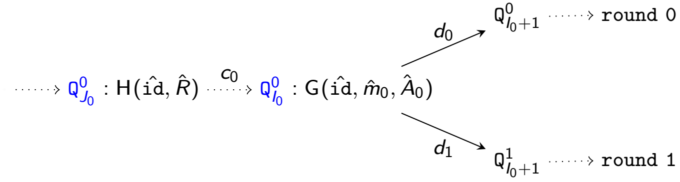
- Consider `round 0` and `round 1` of simulation for GG-IBS



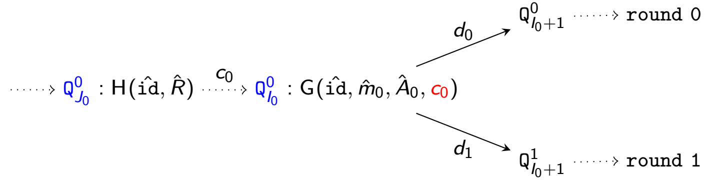
- Need to explicitly ensure that $(J_1 = J_0)$

Inducing Random-Oracle Dependency

- Consider round 0 and round 1 of simulation for GG-IBS

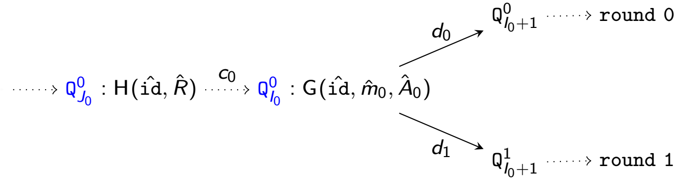


- Need to explicitly ensure that $(J_1 = J_0)$

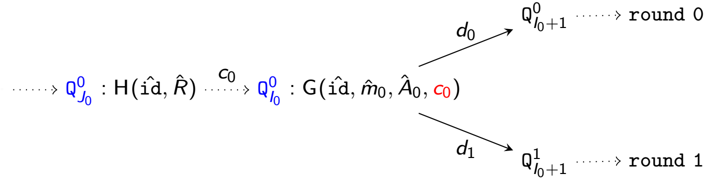


Inducing Random-Oracle Dependency

- Consider `round 0` and `round 1` of simulation for GG-IBS



- Need to explicitly ensure that $(J_1 = J_0)$



- Hence, $(I_1 = I_0) \implies (J_1 = J_0)$!

Galindo-Garcia IBS with Binding

Setting:

1. We work in a group $\mathbb{G} = \langle g \rangle$ of prime order p.
2. Two hash functions $\mathsf{H}, \mathsf{G} : \{0,1\}^* \to \mathbb{Z}_p$ are used.

Set-up:

1. Select $z \in_R \mathbb{Z}_p$ as the $\mathtt{msk}$; set $Z := g^z$ as the $\mathtt{mpk}$

Key Extraction:

1. Select $r \in_R \mathbb{Z}_p$ and set $R := g^r$.
2. Return $\mathtt{usk} := (y, R)$ as the $\mathtt{usk}$, where $y := r + zc$ and $c := \mathsf{H}(\mathtt{id}, R)$.

Signing:

1. Select $a \in_R \mathbb{Z}_p$ and set $A := g^a$.
2. Return $\sigma := (b, R, A)$ as the signature, where $b := a + yd$ and $d := \mathsf{G}(m, A, c)$.

Random Oracle Dependency...

Definition (Random-Oracle Dependency)

A random oracle H_2 is defined to be η-dependent on the random oracle H_1 ($\mathsf{H}_1 \prec \mathsf{H}_2$) if the following criteria are satisfied:

1. $(1 \leq J < I \leq q)$ and
2. $\Pr[(J' \neq J) \mid (I' = I)] \leq \eta$

Random Oracle Dependency...

Definition (Random-Oracle Dependency)

A random oracle H_2 is defined to be η-dependent on the random oracle H_1 ($\mathsf{H}_1 \prec \mathsf{H}_2$) if the following criteria are satisfied:

1. $(1 \leq J < I \leq q)$ and
2. $\Pr[(J' \neq J) \mid (I' = I)] \leq \eta$

Claim (Binding induces dependency)

Binding H_2 to H_1 induces a random-oracle dependency $\mathsf{H}_1 \prec \mathsf{H}_2$ with $\eta_b := q_1(q_1 - 1)/|\mathbb{R}_1|$.

- Here q_1 denotes the upper bound on the number of queries to the random oracle H_1; $\mathbb{R}_1$ denotes the range of H_1.

A UNIFIED TREATMENT

A Unified Model

- Depending on whether $\mathtt{O}_1$ and $\mathtt{O}_2$ is applicable, we get four different MF Algorithms and MF Lemmas
- To incorporate this, we add additional abstraction to the MF Algorithm
  - The condition itself is passed as a parameter

General Multiple-Forking Lemma

MF	Set of Conditions	Degradation
Original	$\mathbb{A}_0 = \begin{cases} \mathsf{B}: & (I_0 \geq 1) \wedge (J_0 \geq 1) \\ \mathsf{C}_k: & (I_{k+1}, J_{k+1}) = (I_k, J_k) \wedge (s_{I_k}^{k+1} \neq s_{I_k}^k) \\ \mathsf{D}_k: & (I_k, J_k) = (I_0, J_0) \wedge (s_{J_0}^k \neq s_{J_0}^l) \end{cases}$	$O(q^{2n})$
with $\mathtt{O}_1$	$\mathbb{A}_1 = \begin{cases} \mathsf{B}: & (I_0 \geq 1) \wedge (J_0 \geq 1) \\ \mathsf{C}_k: & (I_{k+1}, J_{k+1}) = (I_k, J_k) \wedge (s_{I_k}^{k+1} \neq s_{I_k}^k) \\ \mathsf{D}_k: & (J_k = J_0) \wedge (I_k \geq 1) \wedge (s_{J_0}^k \neq s_{J_0}^l) \end{cases}$	$O(q^{(3n+1)/2})$
with $\mathtt{O}_2$	$\mathbb{A}_2 = \begin{cases} \mathsf{B}: & (1 \leq J_0 < I_0 \leq q) \\ \mathsf{C}_k: & (I_{k+1} = I_k) \wedge (s_{I_k}^{k+1} \neq s_{I_k}^k) \\ \mathsf{D}_k: & (I_k, J_k) = (I_0, J_0) \wedge (s_{J_0}^k \neq s_{J_0}^l) \end{cases}$	$O(q^{(3n-1)/2})$
with $\mathtt{O}_1$&$\mathtt{O}_2$	$\mathbb{A}_3 = \begin{cases} \mathsf{B}: & (1 \leq J_0 < I_0 \leq q) \\ \mathsf{C}_k: & (I_{k+1} = I_k) \wedge (s_{I_k}^{k+1} \neq s_{I_k}^k) \\ \mathsf{D}_k: & (J_k = J_0) \wedge (J_k < I_k \leq q) \wedge (s_{J_0}^k \neq s_{J_0}^l) \end{cases}$	$O(q^n)$

- Condition F : $\wedge_{k=0,2,\ldots,n-1} \mathsf{C}_k \wedge \mathsf{D}_k$

General Multiple-Forking Algorithm

$\mathcal{N}_{\mathbb{A},\mathcal{W},n}$

Pick coins ρ for $\mathcal{W}$ at random

$\{s_1^0, \ldots, s_q^0\} \in_R \mathbb{S}$;

$(I_0, J_0, \sigma_0) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_q^0; \rho)$ //round 0

$\{s_{I_0}^1, \ldots, s_q^1\} \in_R \mathbb{S}$;

$(I_1, J_1, \sigma_1) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_{I_0-1}^0, s_{I_0}^1, \ldots, s_q^1; \rho)$ //round 1

if $\neg(\mathsf{B} \wedge \mathsf{C}_0)$ **then return** $(0, \perp)$

$k := 2$

while $(k < n)$ **do**

$\{s_{J_0}^k, \ldots, s_q^k\} \in_R \mathbb{S}$;

$(I_k, J_k, \sigma_k) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_{J_0-1}^0, s_{J_0}^k, \ldots, s_q^k; \rho)$ //round k

$\{s_{I_k}^{k+1}, \ldots, s_q^{k+1}\} \in_R \mathbb{S}$;

$(I_{k+1}, J_{k+1}, \sigma_{k+1}) \leftarrow \mathcal{W}(x, s_1^0, \ldots, s_{J_0-1}^0, s_{J_0}^k, \ldots, s_{I_k-1}^k, s_{I_k}^{k+1}, \ldots, s_q^{k+1}; \rho)$ //round k+1

if $\neg(\mathsf{C}_k \wedge \mathsf{D}_k)$ **then return** $(0, \perp)$

$k := k + 2$

end while

return $(1, \{\sigma_0, \ldots, \sigma_n\})$

CONCLUSION AND FUTURE WORK

Conclusion and Future Work

Conclusions:

- Identified the source of degradation for multiple forking and gave a tighter bound
- A unified model for multiple forking

Future directions:

- Is the bound optimal?
- *Other applications* for RO dependency?
  - Γ-protocols [YZ13]
  - Extended Forking Lemma [YADV+12]
- Other techniques to induces RO dependency

THANK YOU!

Bibliography

BN06 Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma – *CCS'06*

BPW12 Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. Secure proxy signature schemes for delegation of signing rights – *JoC*, 25

CMW12 Sherman Chow, Changshe Ma, and Jian Weng. Zero-knowledge argument for simultaneous discrete logarithms – *Algorithmica*, 64(2)

GG09 David Galindo and Flavio Garcia. A Schnorr-like lightweight identity-based signature scheme – *AFRICACRYPT'09*.

PS00 David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures – *JoC*, 13

Bibliography...

Seu12 Yannick Seurin. On the exact security of Schnorr-type signatures in the random oracle model – *EUROCRYPT'12*

YADV+ Sidi-Mohamed Yousfi-Alaoui, Özgür Dagdelen, Pascal Véron, David Galindo and Pierre-Louis Cayrel. Extended Security Arguments for Signature Schemes – *AFRICACRYPT'12*

YZ13 Andrew Chi-Chih Yao and Yunlei Zhao. Online/offline signatures for low-power devices – *IEEE Transactions on Information Forensics and Security*, 8(2)

Further Reading

CKK12 Sanjit Chatterjee, Chethan Kamath, and Vikas Kumar. Galindo-Garcia identity-based signature revisited – *ICISC'12*

CK13 Sanjit Chatterjee and Chethan Kamath. A Closer Look at Multiple-Forking: Leveraging (In)dependence for a Tighter Bound – *IACR eprint archive*, 2013/651