SYSTEM SECURITY 2

This whole course is solely directed to following idea.

Reduction:
- Assume $\mathcal{A}$ exists.
- Wrap $\mathcal{A}$ with your own code.
- Initialize $\mathcal{A}$.
- Serve all its queries.
- Wrap the hard problem $\mathcal{P}$ as input to $\mathcal{A}$.
- Unwrap the answer from $\mathcal{A}$ as the solution to $\mathcal{P}$.

Conclusion:
- We have solved $\mathcal{P}$ using $\mathcal{A}$. Therefore $\mathcal{A}$ doesn't exist!

Security experiment for passive $\mathcal{A}$ identification scheme (IS).

$Exp_{IS}^{PA,\mathcal{A},\lambda,L}$:

1° Init stage
$PAR \leftarrow Init(\lambda)$
$a, A \leftarrow KeyGen(PAR)$

2° Query stage $(\pi(P(a), V(A)))$
$\pi(P(a), V(A)) \rightarrow T_1$
$\vdots \quad \vdots$
$\pi(P(a), V(A)) \rightarrow T_L \rightarrow V^{P,V,L} = \{T_i\}_i^L$

3° Impersonation stage
$\pi(\mathcal{A}(PAR, A, V^{P,V,L}), V(A))$

Advantage

The advantage of $\mathcal{A}$ in $Exp_{IS}^{PA,\mathcal{A},\lambda,L}$ is probability of acceptance in Impersonation stage.

$$Adv(\mathcal{A}, Exp_{IS}^{PA,\mathcal{A},\lambda,L}) = Pr[\pi(\mathcal{A}(PAR, A, V^{P,V,L}), V(A)) \rightarrow 1]$$

The IS scheme is secure when $Adv(\ldots) < \varepsilon(\lambda, L)$

Security experiment for active $\mathcal{A}$ identification scheme (IS)

$Exp_{IS}^{AA,\mathcal{A},\lambda,L}$:

1° Init stage
$PAR \leftarrow Init(\lambda)$
$a, A \leftarrow KeyGen(PAR)$

2° Query stage $(\pi(P(a), \mathcal{A}(A)))$
$\pi(P(a), \mathcal{A}(A)) \rightarrow T_1$
$\vdots$
$\pi(P(a), \mathcal{A}(A)) \rightarrow T_L \rightarrow V^{P,\mathcal{A},L} = \{T_i\}_i^L$

3° Impersonation stage
$\pi(\mathcal{A}(PAR, A, V^{P,\mathcal{A},L}), V(A))$

Advantage

$$Adv(\mathcal{A}, Exp_{IS}^{AA,\mathcal{A},\lambda,L}) = Pr[\pi(\mathcal{A}(PAR, A, V^{P,\mathcal{A},L}), V(A)) \rightarrow 1]$$

Chosen Prover's Ephemeral (IS)

$Exp_{IS}^{CPE,\lambda,L}$:

1° Init stage
$PAR \leftarrow Init(\lambda)$
$a, A \leftarrow KeyGen(PAR)$

2° Query stage ($\mathcal{A}$ is $\bar{V}$)
$\pi(P^{\overline{eph_1}}(a,A), \bar{V}(A, \overline{eph_1}))$
$\vdots \qquad \vdots$
$\pi(P^{\overline{eph_L}}(a,A), \bar{V}(A, \overline{eph_L}))$
$\rightarrow V^{P,\bar{V},\overline{eph}(L)}$

3° Impersonation stage
$\pi(\mathcal{A}(A, V^{P,\bar{V},\overline{eph}(L)}), V(A))$

Advantage
$Adv(\mathcal{A}, Exp_{IS}^{CPE,\lambda,L}) = Pr[\pi(\mathcal{A}^{A, V^{P,\bar{V},\overline{eph}(L)}}), V(A)) = 1]$

Note: Schnorr IS is not secure in that model.

Computational Diffie-Hellman (CDH):

For any PPT $\mathcal{A}_{CDH}$:

$$Pr[\mathcal{A}_{CDH}(G, g^x, g^y) = g^{xy} \mid G \leftarrow_\$ \mathcal{G}(\lambda), x \leftarrow_\$ \mathbb{Z}_q^*, y \leftarrow_\$ \mathbb{Z}_q^*] \leq \epsilon_{CDH}(\lambda)$$

where $\epsilon_{CDH}(\lambda)$ is negligible

Decisional Diffie-Hellman (DDH):

Let $G \leftarrow_\$ \mathcal{G}(\lambda)$, $x, y, z \leftarrow_\$ \mathbb{Z}_q^*$ then $D_0 = (G, g^x, g^y, g^{xy})$, $D_1 = (G, g^x, g^y, g^z)$

For any PPT $\mathcal{A}_{DDH}$:

$$|Pr[\mathcal{A}_{DDH}(D_0) = 0] - Pr[\mathcal{A}_{DDH}(D_1) = 0]| \leq \epsilon_{DDH}(\lambda)$$

where $\epsilon_{DDH}(\lambda)$ is negligible

Gap Computational Diffie-Hellman (GDH):

Let $\mathcal{O}_{DDH}$ be a DDH oracle, $\mathcal{O}_{DDH}(G, g^x, g^y, g^z) \rightarrow 1$ if and only if $z = xy$

then GDH states that

$$Pr[\mathcal{A}_{GDH}^{\mathcal{O}_{DDH}}(G, g^x, g^y) = g^{xy} \mid G \leftarrow_\$ \mathcal{G}(\lambda), x \leftarrow_\$ \mathbb{Z}_q^*, y \leftarrow_\$ \mathbb{Z}_q^*] \leq \epsilon_{GDH}(\lambda)$$

where $\epsilon_{GDH}(\lambda)$

Note: Obligatory assumption for groups with pairing. Because pairing is $\mathcal{O}_{DDH}$.

Unforgeability for signing scheme (SS):

Unforgeability:

1° Init stage

$PAR \leftarrow Init(\lambda)$

$(a, A) \leftarrow KeyGen(PAR)$

2° Query stage ($\mathcal{F}(A)$)

$O_{sign}(m_1) \rightarrow \sigma_1$ ⋮ ⋮ $O_{sign}(m_i) \rightarrow \sigma_i$ } = $\mathcal{F}$ issues M

3° Forgery stage

$m^*, \sigma^* \leftarrow \mathcal{F}^{O_{sign}}(A)$

Advantage

The advantage of $\mathcal{F}$ in unforgeability experiment

$Pr[(m^*, \sigma^*) \leftarrow \mathcal{F}^{O_{sign}}(A),\ m^* \notin M,\ Vonify(\sigma^*, m^*, A) \rightarrow 1] = Adv(\mathcal{F}^{O_{sign}})$

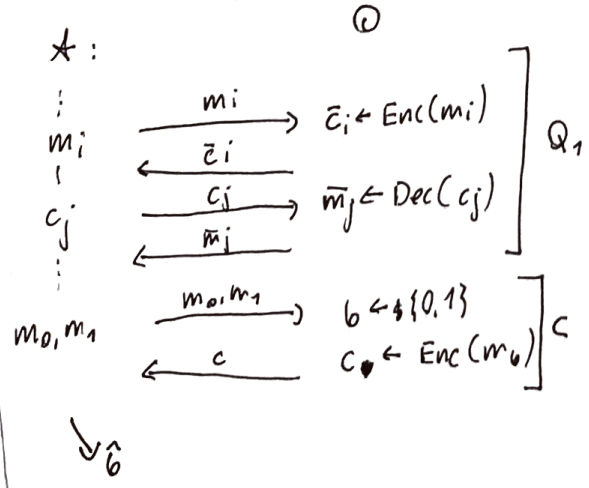
Note: In Random Oracle model we have to model (serve) every Oracle present in the scheme

RSA assumption:

- easy: $f_{RSA}(x, (N, e)) = x^e \bmod N \rightarrow y$
- difficult: $f_{RSA}^{-1}(y, \_) \rightarrow x$ (it can be viewed as $y^d \bmod N \rightarrow x$, without d)

Chosen-Ciphertext-Attack Experiment (Semantic Security)

$Exp_{CCA}^{\mathcal{A}, O_{Enc}, O_{Dec}}$:



Advantage

$Adv(\mathcal{A}^{O_{Enc}, O_{Dec}}) = \left| Pr[b = \hat{b}] - \frac{1}{2} \right| \leq \varepsilon(\lambda)$

Sequence of Games Approach

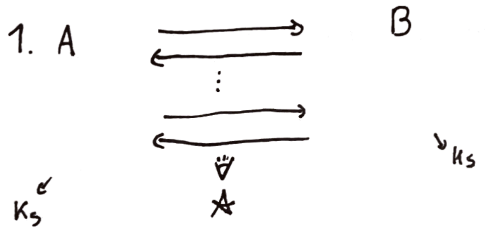
1. Initial game (G_0) is the original game - defined in security experiment.
2. Subsequent games $G_1, G_2, \ldots G_n$ are defined introducing minor changes (upon underlying security assumptions)
3. A should not be able to distinguish between G_i and G_{i+1}.
   If he can distinguish, he can break the underlying security assumption.
4. Final game G_n is such that it is easy to conclude initial security of G_0.

Advantage(G_i) = advantage of A in G_i.

Security of Authenticated Key Exchange

- $K_A = K_B = K_S$
- Parties know their identities
- A $\overset{\text{knows}}{\rightsquigarrow}$ B
- B $\overset{\text{knows}}{\rightsquigarrow}$ A
- K_S is secret (secrecy property)

Real-or-Random Game

1. 

2. $b \leftarrow_\$ \{0,1\}$
3. if $b == 1$:
   $\mathcal{A} \leftarrow K_S$

   else

   $K_R \leftarrow_\$$ KeySpace

   $\mathcal{A} \leftarrow K_R$
4. $\mathcal{A} \rightarrow \hat{b}$

The scheme is secure if $|\Pr[\hat{b} = b] - \frac{1}{2}| \leq \varepsilon$, where ε is negligible

Canetti-Krawczyk Model

- $\mathcal{A}$ controls communication between $\{P_1, \dots, P_n\}$
- s is a session identifier (unique)
- $\mathcal{A}$ queries to Oracles controlling $\{P_1, \dots, P_n\}$
- ~~Matching session between P_i and P_j when:~~
  ~~they haven't completed yet~~
  ✓
  have completed with output (P_i, s, P_j)

Session Key secrecy

A protocol π is secure in CK model if $\forall$ PPT $\mathcal{A}$:

- P_i, P_j (uncorrupted) complete matching session with outputs (P_i, s, P_j) and (P_j, s, P_i) and $K_A = K_B$ ~~$=K_B$~~ accept with negligible probability
- $|\Pr[\hat{b} = b] - \frac{1}{2}| \leq \varepsilon$

oracle's methods:

- Send(P_i, P_j, s, m)
  - if K_s established: return accept
  - elif P_j stopped execution of s: return $\perp$
  - else: return resp $\leftarrow P_j(m)$
- Corrupt(P_i, s)
  - return entire internal state P_i
- SessionKey Reveal(P_i, P_j, s)
  - if K established: return K
  - else: return $\perp$
- Expire Session(P_i, s)
  - delete complete session state P_i
- StateReveal(P_i, s)
  - if K established: return $\perp$
  - else return: 'state' except LT keys
- Test(P_i, P_j, s) ← called only once - ends the experiment
  - if matchSession$(P_i, P_j) \wedge K$ established:
    - $b \leftarrow_\$ \{0,1\}$
    - if $b = 0$: return K
    - if $b = 1$: return $R \leftarrow_\$$ KeySpace
  - else: return $\perp$

Canetti-Krawczyk-Refined Model

CK model + 2 additional Oracle's methods

- Session is exposed if one of:
  1. SessionKey Reveal
  2. State Reveal
  3. Corrupt
  4. LongTerm Key Reveal

- Ephemeral Key Reveal (P_i, s)
  | return: eph key of P_i
- LongTerm KeyReveal (P_i, s)
  | return: Lt key of P_i

- Once parties P_i, P_j are in matching session we allow
  * to issue combination of LongTermKey Reveal and EphKeyReveal
  but not to the same party (attack would be trivial)

Schnorr IS

Prover $(a, A = g^a)$		Verifier (A)
$x \leftarrow \$ \mathbb{Z}_q^*$		
$\mathcal{X} = g^x$	$\xrightarrow{\mathcal{X}}$	
		$c \leftarrow \$ \mathbb{Z}_q^*$
	$\xleftarrow{c}$	
$s = x + ac$	$\xrightarrow{s}$	accept iff $g^s = \mathcal{X} A^c$

Proof: We assume that there exists $\mathcal{A}$ such that can forge Schnorr IS. We will use it to break DLP assumption, namely $DLg\, g^a$.

1° Init Stage

$PAR \leftarrow Init(\lambda)$

$(g, g^a) \leftarrow GetInstanceDLP(PAR)$

set $A (= g^a) \rightarrow \mathcal{A}$

2° Simulation for Query stage

$\mathcal{A} \leftarrow GenView(l)$

```
GenView (l):
for i=1 to l:
    s' ←$ Z_q*
    c' ←$ Z_q*
    X' = g^s' / A^c'
    T' = (X', c', s')
    V := V ∪ T'
return V
```

// generating transcripts in such a way guarantees that $\mathcal{A}$ can not distinguish if T' from real T.

3° Impersonation Stage

1. $\mathcal{X} \leftarrow \mathcal{A}$
2. $c \leftarrow \$ \mathbb{Z}_q^*$ → 2'. $c' \leftarrow \$ \mathbb{Z}_q^*$
3. $c \rightarrow \mathcal{A}$ → 3'. $c' \rightarrow \mathcal{A}$
4. $s \leftarrow \mathcal{A}$ → 4'. $s' \leftarrow \mathcal{A}$

4° Reduction to DLP

if transcripts are accepted:

$$\begin{cases} s = x + ac \\ s' = x + ac' \end{cases} \rightarrow a = \frac{s - s'}{c - c'}$$

5° Contradiction

We used $\mathcal{A}$ to calculate $DLg\, g^a \rightarrow a$. But this is against DLP assumption, thus $\mathcal{A}$ can not exist.

Okamoto IS

Prover $(a_1, a_2, A = g_1^{a_1} g_2^{a_2})$		Verifier (A)
$x_1 \leftarrow_\$ \mathbb{Z}_q^*$		
$x_2 \leftarrow_\$ \mathbb{Z}_q^*$		
$\mathcal{X} = g_1^{x_1} \cdot g_2^{x_2}$	$\xrightarrow{\mathcal{X}}$	$c \leftarrow_\$ \mathbb{Z}_q^*$
	$\xleftarrow{c}$	
$s_1 = x_1 + a_1 c$		
$s_2 = x_2 + a_2 c$	$\xrightarrow{s_1, s_2}$	accept iff $g_1^{s_1} \cdot g_2^{s_2} = \mathcal{X} A^c$

Proof: We assume that there exists $\mathcal{A}$ such that can forge Okamoto IS. We will use it to break DLP, namely $DL_{g_1 g_2}$.

1° Init stage

$PAR \leftarrow Init(\lambda)$

$a_1, a_2 \leftarrow_\$ \mathbb{Z}_q^*$

$A = g_1^{a_1} \cdot g_2^{a_2} \rightarrow \mathcal{A}$

2° Playing in Query Stage

Play Query Stage $(\mathcal{A}, P(a_1, a_2), \mathcal{A}(A))$

Play Query Stage $(P, \mathcal{A})$:

for i=1 to q:

$V := V \cup \pi(P, \mathcal{A})$

return V

3° Impersonation Stage

1. $\mathcal{X} \leftarrow \mathcal{A}$
2. $c \leftarrow_\$ \mathbb{Z}_q^*$
3. $c \rightarrow \mathcal{A}$
4. $s_1, s_2 \leftarrow \mathcal{A}$

→ 2'. $c' \leftarrow_\$ \mathbb{Z}_q^*$

3'. $c' \rightarrow \mathcal{A}$

4'. $s_1', s_2' \leftarrow \mathcal{A}$

4° Reduction to DLP

if transcripts an acceptable (verifiable):

$$\begin{cases} s_1 = x_1 + a_1 c \\ s_2 = x_2 + a_2 c \end{cases} \qquad \begin{cases} s_1' = x_1 + a_1 c' \\ s_2' = x_2 + a_2 c' \end{cases}$$

note $g_2 = g_1^{\alpha} \rightarrow g_1^{s_1} g_2^{s_2} = g_1^{s_1} g_1^{\alpha s_2} = g_1^{s_1 + \alpha s_2}$

so:

$$\begin{cases} s_1 + \alpha s_2 = x_1 + a_1 c + \alpha(x_2 + a_2 c) \\ s_1' + \alpha s_2' = x_1 + a_1 c' + \alpha(x_2 + a_2 c') \end{cases}$$

then:

$$s_1 + \alpha s_2 - s_1' + \alpha s_2' = a_1 c + \alpha a_2 c - a_1 c' - \alpha a_2 c'$$

$$\alpha(s_2 - s_2') + \alpha(a_2 c' - a_2 c) = a_1 c - a_1 c' - s_1 + s_1'$$

$$\alpha = \frac{a_1 c - a_1 c' - s_1 + s_1'}{s_2 - s_2' + a_2 c' - a_2 c}$$

5° Contradiction

We used $\mathcal{A}$ to calculate $DL_{g_1 g_2} \rightarrow \alpha$.

This is against DLP assumption, thus $\mathcal{A}$ can not exist.

Mod Schnorr IS

Prover $(a, A = g^a)$		Verifier (A)
$x \leftarrow_\$ \mathbb{Z}_q^*$		
$\mathcal{X} = g^x$	$\xrightarrow{\mathcal{X}}$	
	$\xleftarrow{c}$	$c \leftarrow_\$ \mathbb{Z}_q^*$
$\hat{g} = \mathcal{H}(\mathcal{X}, c)$		
$s = x + ac$		
$\hat{S} = \hat{g}^s$	$\xrightarrow{\hat{S}}$	$\hat{g} = \mathcal{H}(\mathcal{X}, c)$
		accept iff:
		$e(\hat{S}, g) == e(\hat{g}, \mathcal{X} A^c)$

Proof: We assume that exists $\bigstar$ such that can win $Exp_{IS}^{CDE, \mathcal{R}, \mathcal{L}}$ for Mod Schnorr IS. We will use it to break GDH assumption.

0° Preparing ROM table · $\mathcal{H}$ table (we use ':=' to denote that we assign some value to $\mathcal{H}$)

1° Init stage

$PAR \leftarrow Init(\lambda)$

$g, g^\alpha, g^\beta \leftarrow GetGDHInstance(PAR)$

$A = g^\alpha \rightarrow \bigstar$

2° Simulation for Passive and Active Query stage

$V^{P,V,\mathcal{L}_1} \leftarrow$ SimulationPassiveView$(\mathcal{L}_1)$

$V^{P,\bar{V},\mathcal{L}_2} \leftarrow$ SimulationActiveView$(\mathcal{L}_2)$

SimulationPassiveView$(\mathcal{L}_a)$:

for $i = 1$ to $\mathcal{L}_a$:

$s', c' \leftarrow_\$ \mathbb{Z}_q^*$

$\mathcal{X}' = \frac{g^{s'}}{A^{c'}}$

$\hat{g}' = \mathcal{H}(\mathcal{X}', c')$

$\hat{S}' = \hat{g}^{s'}$

$V := V \cup (\mathcal{X}', c', \hat{S}')$

↑ transcripts received as they are indistinguishable

3° Impersonation stage

1. $\mathcal{X} \leftarrow \bigstar$
2. $c, d \leftarrow_\$ \mathbb{Z}_q^*$
3. $\hat{g} \leftarrow \mathcal{H}(\mathcal{X}, c) := (g^\beta)^d$
4. $c \rightarrow \bigstar$
5. $\hat{S} \leftarrow \bigstar$

2'. $c', d' \leftarrow_\$ \mathbb{Z}_q^*$

3'. $\hat{g}' \leftarrow \mathcal{H}(\mathcal{X}, c') := (g^\beta)^{d'}$

4'. $c' \rightarrow \bigstar$

5'. $\hat{S}' \leftarrow \bigstar$

SimulationActiveView$(\mathcal{L})$:

for $i = 1$ to $\mathcal{L}$:

$\bar{x} \leftarrow \bigstar$

$\bar{\mathcal{X}} = g^{\bar{x}}$

$\bar{\mathcal{X}} \rightarrow \bigstar$

$c \leftarrow \bigstar$

$r \leftarrow_\$ \mathbb{Z}_q$

$\hat{g} \leftarrow \mathcal{H}(\bar{\mathcal{X}}, c) := g^r$

$\hat{S} = \bar{\mathcal{X}}^r A^{rc}$

$\hat{S} \rightarrow \bigstar$

4° Reduction to GDH

if transcripts were derivable:

$$\begin{cases} \hat{S} = g^{\beta d(x+ac)} \\ \hat{S}' = g^{\beta d(x+ac')} \end{cases}$$

$$\rightarrow \frac{\hat{S}^{\frac{1}{d}}}{\hat{S}'^{\frac{1}{d'}}} = \frac{g^{\beta(x+ac)}}{g^{\beta(x+ac')}} = g^{\alpha\beta(c-c')} \rightarrow$$

$$\rightarrow \left(\frac{\hat{S}^{\frac{1}{d}}}{\hat{S}'^{\frac{1}{d'}}} \right)^{\frac{1}{(c-c')}} = g^{\alpha\beta}$$

5° Conclusion

We used $\bigstar$ to solve $GDH(g, g^\alpha, g^\beta) \rightarrow g^{\alpha\beta}$.

This is against GDH assumption therefore $\bigstar$ can not exist.

Schnorr SS

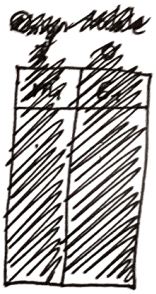
Signon(a, m)
- $x \leftarrow_{\$} \mathbb{Z}_q^*$
- $\mathcal{X} = g^x$
- $h = \mathcal{H}(\mathcal{X}, m)$
- $s = x + ah$
- $\sigma = (\mathcal{X}, s)$

Verifion(A, m, σ)
- $\mathcal{X}, s \leftarrow \sigma$
- $h = \mathcal{H}(\mathcal{X}, m)$
- accept iff $g^s = \mathcal{X} A^h$

Proof: We will show security of Schnorr SS in ROM by contradiction. We assume that exists PPT $\mathcal{F}$, then we will use it to break DLP assumption.

0° Prepening Oracles stage

- $\mathcal{O}_{sign}(m)$
- $\mathcal{O}_{\mathcal{H}}(\mathcal{X}, m)$



$\mathcal{O}_{sign}(m)$:
- $s, h \leftarrow_{\$} \mathbb{Z}_q^*$
- $\mathcal{X} = \frac{g^s}{A^h}$
- $\sigma = (\mathcal{X}, s)$
- register $\mathcal{O}_{\mathcal{H}}((\mathcal{X}, m), h)$
- return σ

$\mathcal{O}_{\mathcal{H}}(\mathcal{X}, m)$
- if $(\mathcal{X}, m)$ in $\mathcal{O}_{\mathcal{H}}$ table: return h
- else
  - $h \leftarrow_{\$} \mathbb{Z}_q^*$
  - register $\mathcal{O}_{\mathcal{H}}((\mathcal{X}, m), h)$
  - return h

$\mathcal{O}_{\mathcal{H}}$ table

I	O
$(\mathcal{X}_1, m_1)$	h_1
⋮	⋮

1° Init stage

$PAR \leftarrow Init(\lambda)$

~~$(q, g^a) \leftarrow$ GetDLPInstance(PAR)~~

2° Query stage

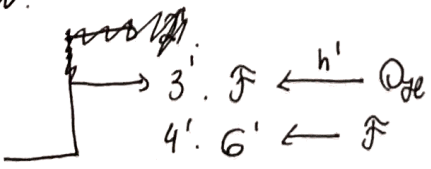
$\mathcal{F}$orgen has access to (simulated) $\mathcal{O}_{sign}, \mathcal{O}_{\mathcal{H}}$. All his queries are verifable:

$g^s = \frac{g^s}{A^h} \cdot A^h = \frac{g^s}{A^h} \cdot A^{\mathcal{O}_{\mathcal{H}}(\mathcal{X}, m)}$

$= \mathcal{X} A^{\mathcal{O}_{\mathcal{H}}(\mathcal{X}, m)}$

3° Forgery stage

1. $\mathcal{F}$ commits to $\mathcal{X}$.
2. $\mathcal{F} \xrightarrow{\mathcal{X}, m} \mathcal{O}_{\mathcal{H}}$
3. $\mathcal{F} \xleftarrow{h} \mathcal{O}_{\mathcal{H}}$
4. $\sigma \leftarrow \mathcal{F}$



4° Reduction to DLP

$$\begin{cases} s = x + ah \\ s' = x + ah' \end{cases} \rightarrow a = \frac{s - s'}{h - h'}$$

5° Contradiction

Using $\mathcal{F}$ we have calculated $DLg^a \rightarrow a$, what is against DLP assumption. Thus such $\mathcal{F}$ can not exist.

Mod Schnorr SS

Signon(α, m)
- $x \leftarrow_\$ \mathbb{Z}_q^*$
- $\mathcal{X} = g^x$
- $h = \mathcal{H}(\mathcal{X}, m)$
- $\hat{g} = \mathcal{H}_g(\mathcal{X}, m)$
- $\hat{S} = \hat{g}^{x+\alpha h}$
- $\sigma = (\mathcal{X}, \hat{S})$

Verifier(A, m, σ)
- $\mathcal{X}, \hat{S} \leftarrow \sigma$
- $h = \mathcal{H}(\mathcal{X}, m)$
- $\hat{g} = \mathcal{H}_g(\mathcal{X}, m)$
- accept iff: $e(\hat{S}, g) == e(g, \mathcal{X} A^h)$

Proof: Assume that exists $\mathcal{F}$ such that can forge signature for Mod Schnorr SS. We will use him to breack GDH.

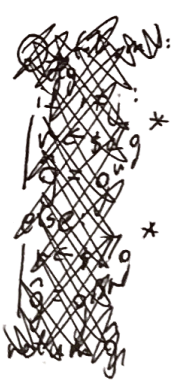
0° Preparing Oracles stage

- $\mathcal{O}_{\mathcal{H}}(\mathcal{X}, m)$
- $\mathcal{O}_{\mathcal{H}_g}(\mathcal{X}, m)$
- $\mathcal{O}^{EPH}_{sign}(m, x)$
- choose randomly $j \leftarrow_\$ \{1, \dots, z\}$

$\mathcal{O}_{\mathcal{H}_g}$ table

$\mathcal{X}_1, m_1$	r_1	g^{r_1}
$\mathcal{X}_j, m_j$	r_j	$g^{\beta r_j}$
⋮		

I r O



$\mathcal{O}_{\mathcal{H}}$ table

$\mathcal{X}_1, m_1$	h_1
⋮	⋮

I O

$\mathcal{O}_{\mathcal{H}_g}(\mathcal{X}, m)$:
- if $(\mathcal{X}, m)$ is in $\mathcal{O}_{\mathcal{H}_g}$ table: return $\hat{g}$
- $r \leftarrow_\$ \mathbb{Z}_q$
- if $i \neq j$: $\hat{g} = g^r$
- if $i = j$: $\hat{g} = g^{\beta r}$
- register $((\mathcal{X}, m), r, \hat{g})$
- return $\hat{g}$

$\mathcal{O}_{\mathcal{H}}(\mathcal{X}, m)$:
- if $(\mathcal{X}, m)$ in $\mathcal{O}_{\mathcal{H}}$ table: get h
- else: $h \leftarrow_\$ \mathbb{G}$; register $((\mathcal{X}, m), h)$
- return h

$\mathcal{O}^{EPH}_{sign}(m, \bar{x})$:
- $\bar{\mathcal{X}} = g^{\bar{x}}$
- $r, \hat{g} \leftarrow \mathcal{O}_{\mathcal{H}_g}(\bar{\mathcal{X}}, m)$
- $h \leftarrow \mathcal{O}_{\mathcal{H}}(\bar{\mathcal{X}}, m)$
- $\hat{S} = g^{r\bar{x}} A^{rh}$
- $\sigma = (\bar{\mathcal{X}}, \hat{S})$

1° Init stage

- $PAR \leftarrow Init(\lambda)$
- $g, g^\alpha, g^\beta \leftarrow GetGDHInstance(PAR)$
- $A = g^\alpha \rightarrow \mathcal{F}$

2° Simulation for Query stage

PlayQuerystage($\mathcal{F}^{\bar{x}}$, $\mathcal{O}^{EPH}_{sign}(m, \bar{x})$)

3° ~~Forced impersonation~~ stage

1. $\mathcal{F}$ commits to $\mathcal{X}$
2. $\mathcal{F} \xrightarrow{\mathcal{X}, m} \mathcal{O}_{\mathcal{H}_g}$
3. $\mathcal{F} \xleftarrow{\hat{g}} \mathcal{O}_{\mathcal{H}_g}$
4. $\mathcal{F} \xrightarrow{\mathcal{X}, m} \mathcal{O}_{\mathcal{H}}$
5. $\mathcal{F} \xleftarrow{h} \mathcal{O}_{\mathcal{H}}$
6. $\mathcal{F} \rightarrow \sigma$

if $\mathcal{F}$ doesn't ask for jth now, restart the experiment

5'. $\mathcal{F} \xleftarrow{h'} \mathcal{O}_{\mathcal{H}}$

6'. $\mathcal{F} \rightarrow \sigma'$

4° Reduction to GDH

if transcript is reusable:

$$\begin{cases} \hat{S} = g^{\beta r(x+\alpha h)} \\ \hat{S}' = g^{\beta r(x+\alpha h')} \end{cases} \rightarrow \frac{\hat{S}^{\frac{1}{r}}}{\hat{S}'^{\frac{1}{r}}} = g^{\beta(\alpha h - \alpha h')} \rightarrow \left(\frac{\hat{S}^{\frac{1}{r}}}{\hat{S}'^{\frac{1}{r}}}\right)^{\frac{1}{h-h'}} = g^{\alpha\beta}$$

5° Conclusion

We have used $\mathcal{F}$ to calculat $GDH(g, g^\alpha, g^\beta) \rightarrow g^{\alpha\beta}$, but that is against GDH assumption, thus such $\mathcal{F}$ can not exist.

Jarecki-Goth SS

Signer(a, m)
$r \leftarrow\$ \{0,1\}^n$
$\hat{g} = \mathcal{H}_g(m, r)$
$K \leftarrow\$ \mathbb{Z}_q^*$
$U = g^K$
$V = \hat{g}^K$
$\hat{A} = \hat{g}^a$
$h = \mathcal{H}(g, \hat{g}, A, \hat{A}, U, V)$
$S = k + ah$
$\sigma = (r, \hat{A}, S, h)$

Verifier(A, m, σ)
$r, \hat{A}, S, h \leftarrow \sigma$
$\hat{g} = \mathcal{H}_g(m, r)$
$U = \frac{g^S}{A^h}$
$V = \frac{\hat{g}^S}{\hat{A}^h}$
iff $h == \mathcal{H}(g, \hat{g}, A, \hat{A}, U, V)$
accept

Proof: We will prove unforgeability of Jarecki-Goth SS by contradiction. Assume that exists $\mathcal{F}_{opt}$, we will use it to break CDH assumption.

0° Preparing Oracles stage
- $\mathcal{O}_{\mathcal{H}}(g, \hat{g}, A, \hat{A}, U, V)$
- $\mathcal{O}_{\mathcal{H}_g}(m, r)$
- $\mathcal{O}_{sign}(m)$

1° Init stage
$PAR \leftarrow Init(\lambda)$
$(g, g^a, g^b) \leftarrow GetCDHInstance(PAR)$
$g^a \neq A \rightarrow \mathcal{F}$

2° Query stage
Forger has access to (simulated) $\mathcal{O}_{\mathcal{H}_g}$, $\mathcal{O}_{\mathcal{H}}$, $\mathcal{O}_{sign}$. All his queries are verifiable. Also $DLg A = DL_{\hat{g}} \hat{A}$ so the simulation is perfect.

3° Forgery stage
$m^*, \sigma^* \leftarrow \mathcal{F}$, $(r, \hat{A}, s, h) \leftarrow \sigma^*$

4° Reduction to CDH
$\hat{g} = \mathcal{H}(m^*, r)$ $(= B^d)$
$\hat{A}^{\frac{1}{a}} = \hat{g}^{a\frac{1}{a}} = B^{da\frac{1}{a}} = g^{bda\frac{1}{a}} = g^{ab}$

5° Conclusion
Using $\mathcal{F}$ we calculated $CDH(g, g^a, g^b) \rightarrow g^{ab}$, what is against CDH assumption. Thus such $\mathcal{F}$ can not exist.

$\mathcal{O}_{\mathcal{H}}(g, \hat{g}, A, \hat{A}, U, V)$:
if registered: return h
$h \leftarrow\$ \mathbb{Z}_q^*$
register $((g, \hat{g}, A, \hat{A}, U, V), h)$
return h

$\mathcal{O}_{\mathcal{H}_g}(m, r)$:
if registered: return d.
$d \leftarrow\$ \mathbb{Z}_q^*$
register $((m, r), B^d, d)$
return B^d

$\mathcal{O}_{sign}(m)$:
$r \leftarrow\$ \{0,1\}^n$
$K \leftarrow\$ \mathbb{Z}_q^*$
$\hat{g} = g^K$
if $\mathcal{O}_{\mathcal{H}_g}(m, r)$ registered:
redraw r or quit
register $((m, r), g^K)$
$\hat{A} = A^K$
$h \leftarrow\$ \mathbb{Z}_q^*$
$s \leftarrow\$ \mathbb{Z}_q^*$
$U = \frac{g^s}{A^h}$
$V = \frac{\hat{g}^s}{\hat{A}^h}$
if $\mathcal{O}_{\mathcal{H}}(g, \hat{g}, A, \hat{A}, U, V)$ registered:
quit
register $((g, \hat{g}, A, \hat{A}, U, V), h)$
return $(h, s, \hat{A}, r)$

Ring SS

RSignen $(m, a_j, \{A_1, \dots A_n\})$

for each $i \neq j$:
- $r_i \leftarrow \$ \mathbb{Z}_q^*$
- $R_i = g^{r_i}$
- $h_i = \mathcal{H}(m, R_i)$

$r_j \leftarrow \$ \mathbb{Z}_q^*$

$R_j = \dfrac{g^{r_j}}{\prod_{i \neq j} A_i^{h_i}}$

$h_j = \mathcal{H}(m, R_j)$

$s = r_j + \sum_{i \neq j} r_i + a_j h_j$

$\sigma = (s, \{R_1, \dots R_n\})$

RVerifion $(\sigma, m, \{A_1, \dots A_n\})$

$s, \{R_1, \dots R_n\} \leftarrow \sigma$

iff $g^s == \prod_i R_i \cdot A_i^{\mathcal{H}(m, R_i)}$

accept

Proof: We will prove unforgeability of Ring SS using contradiction. Assume that $\mathcal{F}_{PPT}$ exists, we will use it to break DLP assumption.

$\mathcal{O}_{\mathcal{H}}(m, R)$:
- if I registered: return h
- $h \leftarrow \$ \mathbb{Z}_q^*$
- register$((m, R), h)$
- return h

$\mathcal{O}_{sign}(m, \{A_1, \dots A_n\})$
- for $i = 1$ to $n-1$:
  - $r_i \leftarrow \$ \mathbb{Z}_q^*$
  - $R_i = g^{r_i}$
  - ~~register~~
  - $\mathcal{O}_{\mathcal{H}}(m, R_i) \rightarrow h_i$
- $s \leftarrow \$ \mathbb{Z}_q^*$
- $h_n \leftarrow \$ \mathbb{Z}_q^*$
- $R_n = \dfrac{g^s}{\prod_{i<n} R_i A_i^{h_i} \cdot A_n^{h_n}}$
- register$((m, R_n), h_n)$
- return $(\sigma = s, \{R_1, \dots R_n\})$

0° Preparing Oracles stage.
- $\mathcal{O}_{\mathcal{H}}(m, R)$
- $\mathcal{O}_{sign}(m, \{A_1, \dots A_n\})$

1° Init stage

$PAR \leftarrow Init(\lambda)$

$(g, g^\alpha) \leftarrow$ GetDLPInstance(PAR)

for $i = 1$ to n:
- $a_i \leftarrow \$ \mathbb{Z}_q^*$
- $g^{\alpha a_i} = A_i \rightarrow \mathcal{F}$

2° Query stage

$\mathcal{F}$orgen has access to (simulated) $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_{sign}$.

All his queries are verifiable.

3° Forgery stage

1. $\mathcal{F}$ commits to $\{r_1, \dots r_n\}$ and $\{R_1, \dots R_n\}$
2. $\mathcal{F} \xrightarrow{(m, R_j)} \mathcal{O}_{\mathcal{H}}$
3. $\mathcal{F} \xleftarrow{h} \mathcal{O}_{\mathcal{H}}$
4. $G \leftarrow \mathcal{F}$

3'. $\mathcal{F} \xleftarrow{h'} \mathcal{O}_{\mathcal{H}}$

4'. $G' \leftarrow \mathcal{F}$

4° Reduction to DLP

if the forgery is verifiable

$$\begin{cases} s = \sum r_i + a_j h_j \\ s' = \sum r_i + a_j h_j' \end{cases} \rightarrow s - s' = a_j (h_j - h_j') \rightarrow a_j = \frac{s - s'}{h_j - h_j'}$$

$$A_j = g^{\alpha \cdot a_j} = g^{a_j} \rightarrow \alpha = \frac{a_j}{a_j}$$

5° Contradiction

We have used $\mathcal{F}$ to calculate $DL_g g^\alpha \rightarrow \alpha$, what is against DLP assumption, thus such $\mathcal{F}$ can not exist.

RSA-FDH SS

Signer$(m, (N, d))$:
  $h = \mathcal{H}(m)$
  $\sigma = h^d \mod N$

Verifier$(\sigma, m, (N, e))$:
  iff $\sigma^e \mod N == \mathcal{H}(m)$:
    accept

Proof: We will prove unforgeability of RSA-FDH SS using contradiction. Assume that PPT $\mathcal{F}$ exists. We will use it to break RSA assumption.

0° Preparing oracles stage
- $\mathcal{O}_{\mathcal{H}}(m)$
- $\mathcal{O}_{sign}(m)$

$\mathcal{O}_{\mathcal{H}}(m)$ and $\mathcal{O}_{sign}(m)$

if $i \neq j$:
  $\sigma \leftarrow_\$$ RSA-NUM
  register(m, σ, σ^e)
  return (σ_{sign} or $\sigma^e_{\mathcal{H}}$)
if $i = j$:
  register$(m, \perp, y)$
  return ($\perp_{sign}$ or $y_{\mathcal{H}}$)

I	$\mathcal{O}_{\mathcal{H}}$	$\mathcal{O}_{sign}$
m_1	σ_1^e	σ_1
⋮	⋮	⋮
m_j	y	$\perp$
⋮	⋮	⋮

j

$\sigma_1 \leftarrow_\$$ RSA-NUM

1° Init stage

$PAR \leftarrow Init(\lambda)$
$(N, e), y \leftarrow$ GetRSAInstance(PAR)
$(N, e) \rightarrow \mathcal{F}$
$j \leftarrow_\$ \{1, \dots z\}$

2° Queries stage

$\mathcal{F}$ has access to $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_{sign}$.
All his queries are verifiable. $h \leftarrow_\$ \mathcal{O}_{\mathcal{H}}(\bar{m})$ and $\sigma \leftarrow \mathcal{O}_{sign}(\bar{m})$
"σ_i^e" $\rightarrow (\sigma_i^e)^d = \sigma_i$ "σ_i"

3° Forgery stage

with probability $\frac{1}{z}$ <u>the forgery</u> will happen for the j-th run.
$m^*, \sigma^* \leftarrow \mathcal{F}$

4° Reduction to RSA assumption

if σ^* is proper signature for m^* ($\sigma^* = h^d \mod N = y^d \mod N$)
we compute $\sigma^{*e} \mod N = h^{de} \mod N = y^{de} \mod N = y$.
thus $\sigma^* = x$ (because $y^d = x \mod N$)

5° Contradiction

We used $\mathcal{F}$ to break $f_{RSA}^{-1}(y, -) \rightarrow x$ without d, what is against RSA assumption, thus such $\mathcal{F}$ doesn't exist.

ElGamal Encryption

$Encryption_y(m)$
- $r \leftarrow_\$ \mathbb{Z}_q^*$
- $\alpha = g^r$
- $\beta = y^r m$
- $c = (\alpha, \beta)$

$Decryption_x(c)$

$m = \frac{\beta}{\alpha^x}$

Proof: We will show semantic security (CCA1) for ElGamal encryption using sequence of games tech.

1° Init stage
- $PAR \leftarrow Init(\lambda)$
- $x, y \leftarrow KeyG\text{-}en(PAR)$

2° Define assumptions for the proof
- DLP
- DDH

3° Series of games

G_0:
- $Enc_y(m_b)$:
  - $r \leftarrow_\$ \mathbb{Z}_q^*$
  - $\alpha = g^r$
  - $\beta = y^r m_b$
  - $c = (\alpha, \beta)$
- $c \rightarrow *$
- $\hat{b} \leftarrow *$
- $p_0 = Pr[\hat{b} == b]$

$\rightarrow$

G_1:
- ...
  - $r \leftarrow_\$ \mathbb{Z}_q^*$
  - $\alpha = g^u$
  - $z \leftarrow_\$ \mathbb{Z}_q^*$
  - $\beta = g^z m_b$
  - $c = (\alpha, \beta)$
- ...
- ...
- $p_1 = ...$

4° Distinguishers

G_0: $c_0 = (g^r, g^{xr} m_b)$

G_1: $c_1 = (g^r, g^z m_b)$

- we observe answers of $*$ for G_0 and G_1:
  - "blue screen"
  - different answers
  - same answers

$\rightarrow$ $*$ can be used to break DDH assumption.
- $(g, g^u, g^v, g^w) \leftarrow$ DDH
- $y = g^u \rightarrow *$
- observe behavior

5° Conclusion

distance between G_0 and G_1: $|Adv(G_0) - Adv(G_1)| \leq \varepsilon_{DDH}$

but $Adv(G_1) = \frac{1}{2}$, because g^z is unrelated to m_b.

so $|Adv(G_0) - \frac{1}{2}| \leq \varepsilon_{DDH}$, where ε_{DDH} is negligible, thus ElGamal is secure in Semantic Security model.

Hashed ElGamal Encryption

Encryption$_g(m)$:
- $r \leftarrow_\$ \mathbb{Z}_q^*$
- $d = g^r$
- $\beta = \mathcal{H}(y^r) \oplus m$
- $c = (d, \beta)$

Decryption$_x(c)$:
- $m = \beta \oplus \mathcal{H}(d^x)$

Proof: We will prove semantic security of Hashed ElGamal encryption using sequence of games approach.

1° Init stage
- $PAR \leftarrow Init(\lambda)$
- $x, y \leftarrow KeyGen(PAR)$

2° Definition of hardness assumptions stage
- DDH
- HSA (Hash Smoothing Assumption)
  There is no algorithm $\mathcal{D}_{\mathcal{H}}$ that distinguishes with non negligible $\overset{P.b}{\mathcal{L}}$ distribution D_0 from D_1, where:
  $D_0 = \{\mathcal{H}(i) : i \leftarrow_\$ \{0,1\}^* \}$
  $D_1 = \{h : h \leftarrow_\$ \{0,1\}^L\}$
  then $Adv(\mathcal{D}_{\mathcal{H}}) = |Pr[\mathcal{D}_{\mathcal{H}}(D_1) = 1] - Pr[\mathcal{D}_{\mathcal{H}}(D_0) = 1]| \leq \varepsilon_{\mathcal{H}}$ for some negligible $\varepsilon_{\mathcal{H}}$.

3° Sequence of games

G_0:
- $Enc_g(m_b)$
  - $r \leftarrow_\$ \mathbb{Z}_q^*$
  - $d = g^r$
  - $\beta = \mathcal{H}(g^r) \oplus m_b$
  - $c = (d, \beta)$
- $c \rightarrow \mathcal{A}$
- $\hat{b} \leftarrow \mathcal{A}$
- $p_0 = Pr[\hat{b} = b]$

G_1:
- ...
- $r \leftarrow_\$ \mathbb{Z}_q^*$
- $d = g^r$
- $z \leftarrow_\$ \mathbb{Z}_q^*$
- $\beta = z \oplus m_b$
- $c = (d, \beta)$
- $c \rightarrow \mathcal{A}$
- $\hat{b} \leftarrow \mathcal{A}$
- $p_1 = Pr[\hat{b} = b]$

G_2:
- ...
- $r \leftarrow_\$ \mathbb{Z}_q^*$
- $d = g^r$
- $z \leftarrow_\$ \{0,1\}^L$
- $\beta = z \oplus m_b$
- $c = (d, \beta)$
- ...
- ...
- $p_2 = ...$

G_0: $c_0 = (g^r, \mathcal{H}(g^{xr}) \oplus m_b)$ $\quad G_1$: $c_1 = (g^r, \mathcal{H}(g^z) \oplus m_b)$

G_1: $c_1 = (g^r, \mathcal{H}(g^z) \oplus m_b)$ $\quad G_2$: $c_2 = (g^r, h \oplus m_b)$

4° Distinguishers: 3 things can happen
- $\mathcal{A}$ outputs $\rightarrow$ $\mathcal{A}$ is a prob. distinguisher
- $\mathcal{A}$ provides diff output
- $\mathcal{A}$ provides same output

We assume that $\mathcal{A}$ is a distinguisher and we will show experiment towards contradiction.

G_0/G_1
- $(g, g^u, g^v, g^w) \leftarrow$ DDHInstance()
- $pk = g^u$
- $c = (g^v, \mathcal{H}(g^w) \oplus m_b) \rightarrow \mathcal{A}$
- if $\mathcal{A}$ behaves like in:
  - $G_0 \rightarrow w = uv$
  - $G_1 \rightarrow w \neq uv$
- but DDH is hard thus such distinguisher can not exist.

G_1/G_2
- $(\mathcal{H}, h) \leftarrow$ HSAInstance()
- $c = (g^r, h \oplus m_b) \rightarrow \mathcal{A}$
- if $\mathcal{A}$ behaves like in:
  - $G_1 \rightarrow h = \mathcal{H}(i)$ for some i
  - $G_2 \rightarrow h \leftarrow_\$ \{0,1\}^L$
- but HSA is hard, thus such distinguisher can not exist

5° Conclusion
- $|Adv(G_0) - Adv(G_1)| \in \varepsilon_{DDH}$
- $|Adv(G_1) - Adv(G_2)| \in \varepsilon_{HSA}$
- $Adv(G_2) = \frac{1}{2}$ (because h is unrelated to pk, $\mathcal{A}$ can only guess)

therefore

$$|Adv(G_0) - \tfrac{1}{2}| \leq \varepsilon_{DDH} + \varepsilon_{HSA}$$

SigMa AKE

$\hat{A}(a, A, B)$ | $\hat{B}(b, A, B)$

$s \leftarrow$ session_id
$x \leftarrow_\$ \mathbb{Z}_q^*$
$X = g^x$

$\xrightarrow{s, X}$

$y \leftarrow_\$ \mathbb{Z}_q^*$
$K = X^y$
$K_0 = PRF(K, 0)$
$K_1 = PRF(K, 1)$
$\sigma_B = SIG_b(s, 1, X, Y)$
$Mac_B = MAC_{K_0}(s, Cont_{1B})$

$\xleftarrow{s, Y, Mac_B, Cont_B}$

$K = Y^x$
$K_0 = PRF(K, 0)$
$K_1 = PRF(K, 1)$
if $MACVer(Mac_B, K_0) \wedge SignVer(\sigma_B, B, \{s, 1, X, Y\})$:
  $\sigma_A = Sig(s, 0, X, Y)$
  $Mac_A = MAC_{K_0}(s, Cont_A)$
  $K_S = K_1$
else:
  reject

$\xrightarrow{s, \sigma_A, Mac_A, Cont_A}$ if $MACVer(Mac_A, K_0) \wedge SignVer(\sigma_A, A, \{s, 0, X, Y\})$:
  $K = K_1$
else
  reject

1° Init stage
$PAR \leftarrow Init(\lambda)$
$g, a, b, A, B \leftarrow KeyGen(PAR)$

2° Definition of hardness assumptions stage
- DDH
- PRF (like HSA)

3° Sequence of games (ROR)

G_0:
...
original game
...

$\xrightarrow{DDH}$

G_1:
...
$z \leftarrow_\$ \mathbb{Z}_q^*$
$K = g^z$
...

$\xrightarrow{PRF}$

G_2:
...
$K_0 \leftarrow_\$ K_{space}$
...

$\xrightarrow{PRF}$

G_3:
...
$K_1 \leftarrow_\$ K_{space}$
...
In this game everything is random.

$\xrightarrow{PRF}$

G_4:
...
$K_0 = PRF(K, 0)$
...

$\xrightarrow{DDH}$

G_5:
...
$K = g^{xy}$
...

4° Distinguishers
For every of above game we can produce anguments that Distinguishes for transition $G_i \rightarrow G_{i+1}$ for every of above game we can produce anguments that Distinguishes for transition $G_i \rightarrow G_{i+1}$ doesn't exist because if he exists he would break hardness assamption.

5° Conclusion
- $|Adv(G_0) - Adv(G_3)| \leq \varepsilon_{DDH} + 2\varepsilon_{PRF}$, but $Adv(G_3) = \frac{1}{2}$ (because it's completely random), so
  $|Adv(G_0) - \frac{1}{2}| \leq \varepsilon_{DDH} + 2\varepsilon_{PRF}$
- $|Adv(G_0) - Adv(G_5)| \leq 2\varepsilon_{DDH} + 3\varepsilon_{PRF}$ what is negligible.