Modern Public Key Cryptography

Provable Security

Lukas Helminger partially based on slides by S. Ramacher

April 14th, 2021

SCIENCE
PASSION
TECHNOLOGY

Outline

Sequences of Games

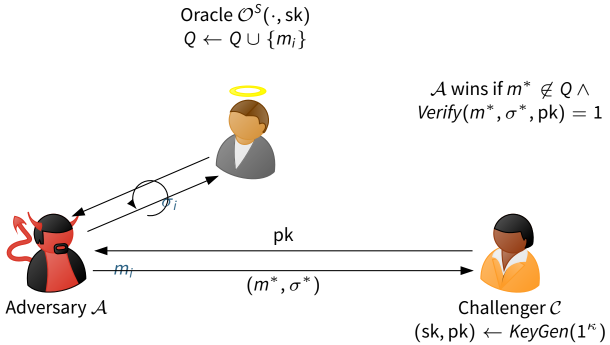
Hybrid Encryption

Game-based Security

- Models security as game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ (which takes on role of all honest parties)
- Interactions between $\mathcal{A}$ and $\mathcal{C}$ well-defined
  - Modeled as oracles that $\mathcal{A}$ can query
  - e.g. $\mathcal{A}$ can query oracle for signatures on arbitrary messages
- At the end, $\mathcal{A}$ required to output "something" (e.g. a message-signature pair)
  - Winning condition specifies what $\mathcal{A}$ must output to win game (e.g. unqueried, valid message-signature pair)

Game-based Security: Example

Experiment $\mathbf{Exp}_{\Sigma}^{\text{EUF-CMA}}(\cdot)$:



Why another proof technique?

- Reductionist proofs are often very complex
  $\rightsquigarrow$ hard to verify
- Idea: What if we slowly "converge" to our solution?
  - We start with original game $G = G_0$, (i.e. security definition)
  - modify it in series of small steps ($G_0 \rightarrow G_1 \rightarrow G_2 \rightarrow$ ...)
  - until we end up in game G_n, which allows to prove the statement
- For each game hop, we have to justify distribution changes of values visible to $\mathcal{A}$!

Sequences of Games (ctd)

- Let S_i be event that $\mathcal{A}$ wins game G_i
  - e.g. outputs signature forgery in game G_i
- We relate $Pr[S_i]$ and $Pr[S_{i+1}]$ for $i = 0, \dots, n-1$
- If $Pr[S_n]$ is (negligibly close to) "target probability" c, then scheme secure
  - Proof gives bound on success probability of $\mathcal{A}$:
    - Bound on $Pr[S_n]$ gives bound on $Pr[S_0]$

    $\Rightarrow$ If $Pr[S_n]$ negligible, then $Pr[S_0]$ negligible as well!

Game Hopping

Three different ways to justify game change:

1. Indistinguishability
   - Computational: If an efficient algorithm can distinguishing G_i from G_{i+1}, then contradiction to underlying hardness assumption.
   - Statistical distance negligible
2. Failure Event: G_i and G_{i+1} identical unless some failure event F occurs
   - $Pr[S_{i+1}] = Pr[S_i] \Pr[\neg F]$
   - if $Pr[F]$ negligible $\Rightarrow$ $Pr[S_{i+1}] \approx Pr[S_i]$
   - but $Pr[F]$ can also be non-negligible
3. Bridging: "Equivalent transformation" to prepare next hop (improves readability) $\Rightarrow$ $Pr[S_i] = Pr[S_{i+1}]$

Sequence of Games Proof of RSA-FDH: Outline

- We will prove RSA-FDH secure using a game series, using
  - bridging steps, and
  - failure events
- Basically, same as before but slower and better readable

Sequence of Games Proof of RSA-FDH: G_0

Game G_0 (original EUF-CMA game)

$(\mathrm{sk}, \mathrm{pk}) = (d, (N, e)) \leftarrow KeyGen(1^\kappa)$

$m_0 \leftarrow \mathcal{A}(\emptyset, \mathrm{pk})$

$h_0 \xleftarrow{R} \mathbb{Z}_N^*$

$\sigma_i \leftarrow h_i^d \mod N$

return $(m^*, \sigma^*) \leftarrow \mathcal{A}((m_0, h_0, \sigma_0), \mathrm{pk})$

Let S_0 be event that $m^* \neq m_0$ and $\sigma^e = H(m)$.

Sequence of Games Proof of RSA-FDH: G_0

Game G_0 (original EUF-CMA game)

$(\mathrm{sk}, \mathrm{pk}) = (d, (N, e)) \leftarrow \mathit{KeyGen}(1^\kappa)$

for $i = 1, \ldots, q$ do

$\quad m_i \leftarrow \mathcal{A}((m_j, h_j, \sigma_j)_{j=1}^{i-1}, \mathrm{pk})$

$\quad h_i \xleftarrow{R} \mathbb{Z}_N^*$

$\quad \sigma_i \leftarrow h_i^d \mod N$

return $(m^*, \sigma^*) \leftarrow \mathcal{A}((m_i, h_i, \sigma_i)_{i=1}^{q}, \mathrm{pk})$

Let S_0 be event that $m^* \neq m_i$ for $i = 1, \ldots, q$ and $\mathit{Verify}(m^*, \sigma^*, \mathrm{pk}) = 1$ in G_0

Sequence of Games Proof of RSA-FDH: G_1

Now, we change game to work without access to sk.

Game G_1

$(\cdot, \mathrm{pk}) = (\cdot, (N, e)) \leftarrow KeyGen(1^\kappa)$

for $i = 1, \ldots, q$ do

$\quad m_i \leftarrow \mathcal{A}((m_j, h_j, \sigma_j)_{j=1}^{i-1}, \mathrm{pk})$

$\quad r_i \xleftarrow{R} \mathbb{Z}_N^*$

$\quad h_i \leftarrow r_i^e \mod N$

$\quad \sigma_i \leftarrow r_i$

return $(m^*, \sigma^*) \leftarrow \mathcal{A}((m_i, h_i, \sigma_i)_{i=1}^{q}, \mathrm{pk})$

From $\mathcal{A}$'s view G_0 and G_1 identical (bridging step): $Pr[S_0] = Pr[S_1]$

Sequence of Games Proof of RSA-FDH: G_2

Include RSA instance (N, e, c) with some probability $1 - p$

Game G_2 (simplified: sim. + game combined)

$\mathsf{pk} \leftarrow (N, e), L \leftarrow \emptyset$

for $i = 1, \ldots, q$ do

$$m_i \leftarrow \mathcal{A}((m_j, h_j, \sigma_j)_{j=1}^{i-1}, \mathsf{pk})$$

$$r_i \xleftarrow{R} \mathbb{Z}_N^*$$

$$h_i \leftarrow \begin{cases} r_i^e \mod N & \text{with probability } p \\ c \cdot r_i^e \mod N & \text{with probability } (1-p) \end{cases}$$

$$\sigma_i \leftarrow \begin{cases} r_i & \text{if } h_i = r_i^e \mod N \\ \text{abort} & \text{otherwise} \end{cases}$$

$$L[m_i] \leftarrow (h_i, r_i)$$

$(m^*, \sigma^*) \leftarrow \mathcal{A}((m_i, h_i, \sigma_i)_{i=1}^{q}, \mathsf{pk}), (h^*, r^*) \leftarrow L[m^*]$

return (m^*, σ^*) if $h^* \neq (r^*)^e \mod N,$ else abort $=0$

Sequence of Games Proof of RSA-FDH: Remarks 2

Remarks

- L is just a list (not visible to $\mathcal{A}$) to store important values
- Experiment aborts if
  - simulation impossible
    - in such cases, reduction would already have to break RSA problem by itself
  - result of "no value"
    - in this case, result is value that reduction can compute itself

Sequence of Games Proof of RSA-FDH: $G_1 \rightarrow G_2$

Transition $G_1 \rightarrow G_2$

Let F be failure event that an abort happens in G_2.

$$Pr[F] = 1 - Pr[Forgery\ good \wedge Simulation\ ok] =$$
$$1 - Pr[Forgery\ good \mid Simulation\ ok] \cdot Pr[Simulation\ ok] =$$
$$1 - (1-p) \cdot p^q$$

Thus, we have $Pr[F] = 1 - (1-p) \cdot p^q$ and get

$$Pr[S_2] = Pr[\neg F] \cdot Pr[S_1] = (1-p)p^q \cdot Pr[S_1]$$

Sequence of Games Proof of RSA-FDH: G_3

Here, we assume that no abort will happen

Game G_3 (simplified: sim. + game combined)

$\mathrm{pk} \leftarrow (N, e), \rho \xleftarrow{R} R$

for $i = 1, \dots, q$ do

$\quad m_i \leftarrow \mathcal{A}((m_j, h_j, \sigma_j)_{j=1}^{i-1}, \mathrm{pk}; \rho)$

$\quad r_i \xleftarrow{R} \mathbb{Z}_N^*$

$$h_i \leftarrow \begin{cases} r_i^e \mod N & \text{with probability } p \\ c \cdot r_i^e \mod N & \text{with probability } (1-p) \end{cases}$$

$\quad \sigma_i \leftarrow r_i$

return $(m^*, c^d \cdot r^*) \leftarrow \mathcal{A}((m_i, h_i, \sigma_i)_{i=1}^{q}, \mathrm{pk}; \rho)$

We have $Pr[S_2] = Pr[S_3]$ (bridging step) and can compute c^d

Sequence of Games Proof of RSA-FDH: Analysis

Analysis

Now, for S_3 (i.e. $\mathcal{A}$ outputs "useful" forgery (m^*, σ^*)) we have as "target probability"

$$Pr[S_3] = \mathbf{Adv}^{\mathsf{OW}}_{\mathsf{RSA}}(\mathcal{R})$$

Combined:

$$\begin{aligned}\mathbf{Adv}^{\mathsf{OW}}_{\mathsf{RSA}}(\mathcal{R}) &= Pr[S_3] = Pr[S_2] = (1-p)p^q \cdot Pr[S_1] = \\ &= (1-p)p^q \cdot Pr[S_0] = (1-p)p^q \cdot \mathbf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{RSA\text{-}FDH}}(\mathcal{A})\end{aligned}$$

Same result as before

Key Encapsulation Mechanism

Definition (KEM, [KL14])

A key-encapsulation mechanism (KEM) is a tuple of PPT algorithm (KGen, Encaps, Decaps) such that:

1. Algorithm KGen takes as input the security parameter 1^n and outputs the key public-/private-key pair (pk, sk).
2. Algorithm Encaps takes as input a public key pk and the security parameter 1^n. It outputs a ciphertext c and a key $k \in \{0, 1\}^{l(n)}$, where $l(n)$ is the key length.
3. Algorithm Decaps takes as input a private key sk and a ciphertext c, and outputs a key k or a special symbol $\perp$ denoting failure.

It is required that with all but negligible probability over (sk, pk) output by $\mathsf{KGen}(1^n)$, if $\mathsf{Encaps}_{\mathsf{pk}}(1^n)$ outputs (c, k), then $\mathsf{Decaps}_{sk}(c)$ outputs k.

KEM/DEM Paradigm

Let $\Pi = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$ be a KEM with key length n, and let $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ be a private-key encryption scheme. Construct a public-key encryption scheme $\Pi^{\mathsf{hy}} = (\mathsf{KGen}^{\mathsf{hy}}, \mathsf{Enc}^{\mathsf{hy}}, \mathsf{Dec}^{\mathsf{hy}})$ as follows:

$$\begin{array}{l}
\underline{\mathsf{KGen}^{\mathsf{hy}}(1^n)} \\
\textbf{return } (\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{KGen}(1^n)
\end{array}
\qquad
\begin{array}{l}
\underline{\mathsf{Enc}^{\mathsf{hy}}(\mathsf{pk}, m)} \\
(c, k) \leftarrow\$ \mathsf{Encaps}_{\mathsf{pk}}(1^n) \\
c' \leftarrow\$ \mathsf{Enc}'_k(m) \\
\textbf{return } (c, c')
\end{array}
\qquad
\begin{array}{l}
\underline{\mathsf{Dec}^{\mathsf{hy}}(\mathsf{sk}, (c, c'))} \\
(k) \leftarrow\$ \mathsf{Decaps}_{\mathsf{sk}}(c) \\
m \leftarrow\$ \mathsf{Dec}'_k(c') \\
\textbf{return } m
\end{array}$$

Efficiency

Fix n.

α... cost of encapsulating (Encaps) an n-bit key
β... cost of encryption (Enc$'$) per bit of plaintext
Assume $|m| > n$ (why?).

What is the cost per bit of plaintext using Π^{hy}?

$\beta \approx \alpha \cdot 10^{-5}, m = 10^6$

Ciphertext Length

Fix n.

L... length of ciphertext output by Encaps
Ciphertext $\mathsf{Enc}'(m)$ has length $n + |m|$.
Assume $|m| > n$ (why?).

What is the ciphertext length of Π^{hy}?

Security

Definition

(KEM Game)

1. $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{KGen}(1^n)$. Then $(c, k) \leftarrow \mathrm{Encaps}_{\mathrm{pk}}(1^n)$, with $k \in \{0,1\}^n$.
2. $b \xleftarrow{R} \{0,1\}$. $\hat{k} = k$ if $b = 0$, else $\hat{k} \xleftarrow{R} \{0,1\}^n$.
3. $b' \leftarrow \mathcal{A}(\mathrm{pk}, c, \hat{k})$. Winning game if $b = b'$.

A KEM is IND-CPA-secure if there exists no adversary that wins with more than $1/2 + negl(n)$ probability.

Further Reading I

[KL14] Jonathan Katz and Yehuda Lindell.
Introduction to Modern Cryptography, Second Edition.
CRC Press, 2014.

[Sho04] Victor Shoup.
Sequences of games: a tool for taming complexity in security proofs.
IACR Cryptology ePrint Archive, 2004:332, 2004.