Security Protocols IX:
The Computational Soundness of
Formal Encryption

Christoph Sprenger and Sebastian Mödersheim
Department of Computer Science
ETH Zurich

FMSEC Module 10, v.2
November 23, 2009
Outline

1. The Formal View
2. The Computational View
3. The Soundness Theorem
Two Views of Cryptography

<table>
<thead>
<tr>
<th></th>
<th>Formal View</th>
<th>Computational View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages</td>
<td>Symbolic terms.</td>
<td>Strings of bits.</td>
</tr>
<tr>
<td>Adversary</td>
<td>Dolev-Yao deduction.</td>
<td>Probabilistic polynomial-time Turing machine.</td>
</tr>
<tr>
<td>Assumes</td>
<td>Perfect cryptography.</td>
<td>Hard problems (e.g., factoring, DL).</td>
</tr>
<tr>
<td></td>
<td>Two separate research communities!</td>
<td></td>
</tr>
</tbody>
</table>

Abadí and Rogaway started to bridge the gap and relate the two views:

\[ M \cong N \implies [M]_\Pi \cong [N]_\Pi \]

\[\implies \text{“Statements of the formal model also hold in the real world”}\]
Computational Soundness

• This lecture discusses the approach by Abadí and Rogaway (early 2000’s) covering passive adversaries.

• This work has paved the way for a very active area of research linking the formal and cryptographic views: The computational soundness of formal protocol models.

Benefits

• Conceptual clarification of the relation between these two views, including their assumptions.

• Enables combination of advantages: we may use symbolic tools (efficiency) to obtain cryptographic guarantees (precision).
Outline

1. The Formal View
2. The Computational View
3. The Soundness Theorem
Messages as Terms

Definition (Terms)

\[ M, N ::= \text{terms} \]
\[ K \quad \text{key (for } K \in \text{Keys)} \]
\[ b \quad \text{bit (for } b \in \text{Bool} = \{0, 1\}) \]
\[ (M, N) \quad \text{pair} \]
\[ \{M\}_K \quad \text{encryption (for } K \in \text{Keys)} \]

- This work only considers symmetric encryption and atomic keys.
- Interpreted in free algebra, e.g.,

\[ \{M\}_K = \{M'\}_{K'} \quad \text{iff} \quad M = M' \text{ and } K = K' \]
Key cycles

**Definition (Cyclic messages)**

- We say that \( K \) encrypts \( K' \) in \( M \) if there is a subterm \( \{N\}_K \) of \( M \) such that \( N \) contains an occurrence of \( K' \).
- We say that \( M \) is cyclic if the relation “encrypts in \( M \)” on keys is cyclic.

**Examples**

- \( K \) encrypts both \( K_1 \) and \( K_2 \) in \( \{ \{ K_2 \}_K \}_K \).
- \( (1, \{K\}_K) \) is cyclic.
- \( (\{K_1\}_K, \{K_2\}_K) \) is cyclic.
- \( (\{K_1\}_K, \{0\}_K) \) is acyclic.

The soundness theorem is restricted to acyclic messages.
Message Deduction by Adversary

Definition (Adversary deduction)

\[
\begin{align*}
M \in H & \quad \text{Inject} & M \in \mathcal{D}\mathcal{Y}(H) \\
M_1 \in \mathcal{D}\mathcal{Y}(H) & \quad \text{Pair} & (M_1, M_2) \in \mathcal{D}\mathcal{Y}(H) \\
K \in \mathcal{D}\mathcal{Y}(H) & \quad \text{Enc} & \{M\}_K \in \mathcal{D}\mathcal{Y}(H) \\
\{M\}_K \in \mathcal{D}\mathcal{Y}(H) & \quad \text{Dec} & M \in \mathcal{D}\mathcal{Y}(H) \\
\end{align*}
\]

Example

For \( H = \{K_3, \{0, \{K_1\}_{K_2}\}_K\}_K \) we have:

\[
(\{K_1\}_{K_2}, \{1\}_{K_3}) \in \mathcal{D}\mathcal{Y}(H)
\]
Intuitively, a pattern is a term that may have some parts that the intruder cannot decrypt.

These undecryptable parts are marked by □.

We want to define a function \( p(M, T) \) that, given a message \( M \) and a set of known keys \( T \), replaces the undecryptable parts of \( M \) by □.
Patterns from Terms

**Definition (Pattern of a term \( M \))**

With respect to a set of keys \( T \):

\[
\begin{align*}
p(K, T) &= K \quad \text{for } K \in \text{Keys} \\
p(b, T) &= b \quad \text{for } b \in \text{Bool} \\
p((M, N), T) &= (p(M, T), p(N, T)) \\
p(\{ M \}_K , T) &= \begin{cases} 
\{ p(M, T) \}_K & \text{if } K \in T \\
\Box & \text{otherwise}
\end{cases}
\end{align*}
\]

With respect to the keys that are derivable from the message \( M \):

\[
\text{pattern}(M) = p(M, \text{Keys} \cap \text{DY}(\{ M \}))
\]

**Example**

\[
\text{pattern}\left( (\{ \{ K_1 \}_K \} K_2 , 0 \} K_3 , K_3 ) \right) = p((\{ \{ K_1 \}_K \} K_2 , 0 \} K_3 , K_3 ), \{ K_3 \}) = (\{ \Box , 0 \} K_3 , K_3 )
\]
Equivalece of Messages

Idea: The adversary cannot distinguish messages with the same pattern.

Definition (Equivalence)

Two terms are equivalent if they yield the same pattern:

\[ M \equiv N \text{ if and only if } \text{pattern}(M) = \text{pattern}(N) \]

Examples

- \((\{\{ K_1 \} K_2 \} K_3, K_3) \equiv (\{\{0\} K_2 \} K_3, K_3)\)
  Both terms yield the same pattern: \((\{\square\} K_3, K_3)\).

- \((\{0\} K, K) \neq (\{0\} K', K')\)
  These terms differ only by a renaming of keys.

Definition (Equivalence up to renaming)

\[ M \sim N \text{ if and only if } \text{if there is a bijection } \sigma \text{ on Keys such that } M \equiv N\sigma. \]
Exploring the Definition of Equivalence

**Examples (basic)**

- $0 \equiv 0, 0 \not\equiv 1$
- $(K, \{0\}_K) \not\equiv (K, \{1\}_K)$  
  Decryptable messages
- $\{0\}_K \equiv \{1\}_K$  
  Undecryptable messages
- $\{0\}_K \equiv \{1\}_{K'}$ and even $\{0\}_K \equiv \{1\}_{K'}$

**Examples (Properties of encryption)**

- $\{(((1, 1), (1, 1)), (1, 1))\}_K \equiv \{0\}_K$
  Cannot deduce the size of a plaintext from ciphertext.
- $(\{0\}_K, \{0\}_K) \equiv (\{0\}_K, \{1\}_K)$
  Cannot distinguish two encryptions of a plaintext under given key.
- $(\{0\}_K, \{1\}_K) \equiv (\{0\}_K, \{1\}_{K'})$
  Cannot detect whether two ciphertexts use the same key.
Outline

1. The Formal View
2. The Computational View
3. The Soundness Theorem
## Preliminaries: Indistinguishability & Co.

### Definitions

- A probability **ensemble** is a collection of distributions on strings, \( D = \{D_\eta\} \), one for each \( \eta \in \mathbb{N} \).

- We write \( x \xleftarrow{\$} D_\eta \) to indicate that \( x \) is sampled from \( D_\eta \).

- We write \( \Pr[x \xleftarrow{\$} D_\eta : E] \) for the probability of event \( E \) when \( x \) is sampled from \( D_\eta \).

- A function \( \varepsilon : \mathbb{N} \to \mathbb{R} \) is **negligible** if for all \( c > 0 \) there exists a \( N_c \) such that \( \varepsilon(\eta) \leq \eta^{-c} \) for all \( \eta \geq N_c \).

- Two ensembles \( D = \{D_\eta\} \) and \( D' = \{D'_\eta\} \) are (computationally) **indistinguishable**, written \( D \approx D' \), if for every probabilistic polynomial time adversary, the following function is negligible:

\[
\varepsilon(\eta) = \left| \Pr[x \xleftarrow{\$} D_\eta : A(\eta, x) = 1] - \Pr[x \xleftarrow{\$} D'_\eta : A(\eta, x) = 1] \right|
\]

Here, \( \eta \) is called the **security parameter**.
Encryption Scheme (1)

Domains
- Plaintext, Ciphertext, Key \( \subseteq \text{String} = \{0, 1\}^* \),
- Parameter \( = 1^* \).

Definition (Encryption scheme)
An encryption scheme is a triple \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) where
- \( \mathcal{K} \) is a probabilistic key generation algorithm with
  - \textbf{input}: a security parameter \( 1^n \in \text{Parameter} \), and
  - \textbf{output}: a key \( \mathcal{K}(1^n) \in \text{Key} \),
- \( \mathcal{E} \) is a probabilistic encryption algorithm with
  - \textbf{input}: a key \( k \in \text{Key} \) and a message \( m \in \text{String} \), and
  - \textbf{output}: a ciphertext \( \mathcal{E}_k(m) \in \text{Ciphertext} \).
- \( \mathcal{D} \) is a deterministic decryption algorithm with
  - \textbf{input}: a key \( k \in \text{Key} \) and a message \( c \in \text{String} \), and
  - \textbf{output}: a cleartext \( \mathcal{D}_k(c) \in \text{Plaintext} \).

Each algorithm runs in time polynomial in the size of its input.
Assumptions

- $\bot \in \text{Plaintext}$,
- For $k \in \mathcal{K}(1^n)$ and $\eta \in \text{Parameter}$:

$$\mathcal{D}_k(\mathcal{E}_k(m)) = \begin{cases} 
  m & \text{if } m \in \text{Plaintext} \\
  \bot & \text{otherwise}
\end{cases}$$

- $|\mathcal{E}_k(x)|$ depends only on $\eta$ and $|x|$ when $k \in \mathcal{K}(1^n)$. 
**Type-0 Security (1)**

**Repetition concealing** ($r$) Given ciphertexts $c$ and $c'$, are their underlying plaintexts equal?

Formal view: $(\{0\}_K, \{0\}_K) \approx (\{0\}_K, \{1\}_K)$

Implementation: requires randomized encryption.

**Which-key concealing** ($k$) Given ciphertexts $c$ and $c'$, are they encrypted with the same key?

Formal view: $(\{0\}_K, \{1\}_K) \approx (\{0\}_K, \{1\}_K')$

**Message-length concealing** ($l$) Does the length of ciphertext reveal the length of plaintext?

Formal view: $\{(((1, 1), (1, 1)), (1, 1))\}_K \approx \{0\}_K$

Implementation: requires maximal message size and padding.

**Type-0 security**: the encryption scheme does not reveal any of these informations. (There is a type-$rkl$ security for $r, k, l \in \text{Bool}$.)
Type-0 Security (2)

Definition (Type-0 security)

An encryption scheme \( \Pi = (K, E, D) \) is type-0 secure if for every probabilistic polynomial-time adversary \( A \) the advantage of \( A \),

\[
\text{Adv}^0_{\Pi[\eta]}(A) = \left| \Pr[k, k' \xleftarrow{R} \mathcal{K}(\eta) : A^{E_k(\cdot), E_k'(\cdot)}(\eta) = 1] - \Pr[k \xleftarrow{R} \mathcal{K}(\eta) : A^{E_k(\bot), E_k(\bot)}(\eta) = 1] \right|
\]

as a function of \( \eta \) is negligible.

- The adversary \( A^{f,g}(\eta) \) has access to two oracles \( f \) and \( g \).
- First probability: Choose two keys \( k \) and \( k' \) by independently running the key generator. Then run the adversary \( A \) giving him access to the oracles \( f = E_k(\cdot) \) and \( g = E_k'(\cdot) \).
- Second probability: Choose a key \( k \) by running the key generator. Then run the adversary \( A \) with oracles \( f = g = \lambda m. E_k(\bot) \). These oracles return samples from \( E_k(\bot) \), ignoring the input \( m \) given.
Definition (Computational semantics of terms)

The distribution \( [M]_{\Pi[\eta]} \) on strings is induced by the following probabilistic algorithm, which defines how a bit string \( y \) is drawn from \( [M]_{\Pi[\eta]} \):

\[
\tau \overset{R}{\leftarrow} \text{Initialize}(\eta, M);
\]
\[
y \overset{R}{\leftarrow} \text{Convert}(\tau, M)
\]

where
\[
\text{Init}(\eta, M) :
\]
for each \( K \in \text{Keys}(M) \) do \( \tau(K) \overset{R}{\leftarrow} K(\eta) \) od;
return \( \tau \)

and
\[
\text{Convert}(\tau, K) = (\tau(K), "key")
\]
\[
\text{Convert}(\tau, b) = (b, "bool")
\]
\[
\text{Convert}(\tau, (M_1, M_2)) = (\text{Convert}(\tau, M_1), \text{Convert}(\tau, M_2), "pair")
\]
\[
\text{Convert}(\tau, \{[M]\}_K) = (E_{\tau(K)}(\text{Convert}(\tau, M)), "ciphertext")
\]
Outline

1. The Formal View
2. The Computational View
3. The Soundness Theorem
The Soundness Theorem

Let $M$ and $N$ be acyclic terms and let $\Pi$ be a type-0 secure encryption scheme. Then

$$M \equiv N \quad \text{implies} \quad \llbracket M \rrbracket_\Pi \approx \llbracket N \rrbracket_\Pi.$$

The rest of this module is devoted to the proof of this theorem.
### Overview of the Proof

<table>
<thead>
<tr>
<th>$M$</th>
<th>$\not\in$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_m$</td>
<td>$N_n$</td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td></td>
</tr>
<tr>
<td>$M_0$</td>
<td>$N_0$</td>
<td></td>
</tr>
</tbody>
</table>

- $[M]_\Pi \not\approx [N]_\Pi$

- We will construct a chain of **hybrid patterns** connecting $M$ and $N$ such that $M_0 = \text{pattern}(M) = \text{pattern}(N) = N_0$.

- Assume $[M]_\Pi \not\approx [N]_\Pi$.

- Show that there is a “large gap” in the chain: $[M_{i-1}]_\Pi \not\approx [M_i]_\Pi$ for some $i$ or $[N_{j-1}]_\Pi \not\approx [N_j]_\Pi$ for some $j$.

- From the adversary $A$ distinguishing the two ensembles at the gap, construct an adversary $A_0$ violating the type-0 security of the encryption scheme $\Pi$, contradicting the assumption of the Theorem.
**Definition (Recoverable and hidden keys)**

Let $\text{Keys}(M)$ be the set of keys occurring in $M$.

\[
\begin{align*}
\text{rec}(M) & = \text{Keys}(M) \cap \mathcal{D} \mathcal{Y} \{M\} \\
\text{hid}(M) & = \text{Keys}(M) - \text{rec}(M)
\end{align*}
\]

**Lemma (Key renaming)**

*It is possible to rename the keys in $M$ and $N$, obtaining $M'$ and $N'$, such that:*

- $\text{pattern}(M') = \text{pattern}(N')$,
- $\text{rec}(M') = \text{rec}(N') = \{J_1, \ldots, J_{\mu}\}$,
- $\text{hid}(M') = \{K_1, \ldots, K_m\}$,
- $\text{hid}(N') = \{K_1, \ldots, K_n\}$, and
- if $K_i$ encrypts $K_j$ in $M'$ or $N'$ then $i \geq j$.  

Key Renaming (2)

Example (Key renaming)

Original terms: \( M \equiv N \)

\[
M = ( \{1, 1\}_{K_4}, K_6, \{K_5\}_{K_2}, \{K_3, K_5\}_{K_4}, \{0, K_3\}_{K_5}, \{K_1\}_{K_6} ) \\
N = ( \{K_3\}_{K_5}, K_1, \{K_4, 1\}_{K_2}, \{0\}_{K_4}, \{K_4\}_{K_5}, \{K_3\}_{K_1} )
\]

Renamed terms: \( M' \equiv N' \)

\[
M' = ( \{1, 1\}_{K_4}, J_1, \{K_2\}_{K_3}, \{K_1, K_2\}_{K_4}, \{0, K_1\}_{K_2}, \{J_2\}_{J_1} ) \\
N' = ( \{J_2\}_{K_3}, J_1, \{K_1, 1\}_{K_2}, \{0\}_{K_1}, \{K_1\}_{K_3}, \{J_2\}_{J_1} )
\]

The resulting terms satisfy the conclusion of Key renaming lemma:

- \( \text{pattern}(M') = \text{pattern}(N') = (\Box, J_1, \Box, \Box, \Box, \{J_2\}_J_1) \)
- \( \text{rec}(M') = \text{rec}(N') = \{J_1, J_2\} \)
- \( \text{hid}(M') = \{K_1, K_2, K_3, K_4\} \)
- \( \text{hid}(N') = \{K_1, K_2, K_3\} \)
- \( \text{encrypts}_\text{in}(M') = \{(K_4, K_2), (K_4, K_1), (K_3, K_2), (K_2, K_1)\} \)
- \( \text{encrypts}_\text{in}(N') = \{(K_3, K_1), (K_2, K_1)\} \)
The Hybrid Patterns (1)

We introduce pattern $M_0, M_1, \ldots, M_m$ and $N_0, N_1, \ldots, N_n$ so that these patterns form a chain from $M$ to $N$.

**Definition (Hybrid patterns)**

For $i \in \{0, \ldots, m\}$ and $j \in \{0, \ldots, n\}$:

$$M_i = p(M', \text{rec}(M') \cup \{K_1, \ldots, K_i\})$$

$$N_j = p(N', \text{rec}(N') \cup \{K_1, \ldots, K_j\})$$

- $M_m = M'$ and $N_n = N'$
- $M_0 = \text{pattern}(M') = \text{pattern}(N') = N_0$
- Recall: $M_0 = N_0$ as a result of renaming.
- Intuitively, $M_i$ and $N_i$ are the patterns that the adversary sees if he has a priori knowledge of the otherwise hidden keys $K_1, \ldots, K_i$.
- The ordering of the keys guarantees that this knowledge does not permit the discovery of other hidden keys.
### Example

\[ M' \]

\[ M_4 = ( \{1, 1\} K_4, J_1, \{K_2\} K_3, \{K_1, K_2\} K_4, \{0, K_1\} K_2, \{J_2\} J_1 ) \]

\[ M_3 = ( □, J_1, \{K_2\} K_3, □, \{0, K_1\} K_2, \{J_2\} J_1 ) \]

\[ M_2 = ( □, J_1, □, □, \{0, K_1\} K_2, \{J_2\} J_1 ) \]

\[ M_1 = ( □, J_1, □, □, □, \{J_2\} J_1 ) \]

\[ M_0 = ( □, J_1, □, □, □, □, \{J_2\} J_1 ) \]

\[ N_0 = ( □, J_1, □, □, □, □, \{J_2\} J_1 ) \]

\[ N_1 = ( □, J_1, □, \{0\} K_1, □, \{J_2\} J_1 ) \]

\[ N_2 = ( □, J_1, \{K_1, 1\} K_2, \{0\} K_1, □, \{J_2\} J_1 ) \]

\[ N_3 = ( \{J_2\} K_3, J_1, \{K_1, 1\} K_2, \{0\} K_1, \{K_1\} K_3, \{J_2\} J_1 ) \]
Defining Ensembles for the Hybrid Patterns

Definition (Computational semantics of patterns)

The distribution $\mathbb{P}_{\Pi[\eta]}$ on strings is induced by the following probabilistic algorithm, which defines how a bit string $y$ is drawn from $\mathbb{P}_{\Pi[\eta]}$:

$$
\begin{align*}
\tau & \leftarrow^R \text{Init}(\eta, M); \\
y & \leftarrow^R \text{Convert}(\tau, M)
\end{align*}
$$

where for some new, fixed key $K_0$ not occurring elsewhere:

$$
\text{Init}(\tau, M) :
$$

for each $K \in \text{Keys}(M) \cup \{K_0\}$ do $\tau(K) \leftarrow^R K(\eta)$ od;

return $\tau$

and

$$
\begin{align*}
\text{Convert}(\tau, K) &= (\tau(K), \text{"key"}) \\
\text{Convert}(\tau, i) &= (i, \text{"bool"}) \\
\text{Convert}(\tau, (M_1, M_2)) &= (\text{Convert}(\tau, M_1), \text{Convert}(\tau, M_2), \text{"pair"}) \\
\text{Convert}(\tau, \{M\}_K) &= (E_{\tau(K)}(\text{Convert}(\tau, M)), \text{"ciphertext"}) \\
\text{Convert}(\tau, \Box) &= (E_{\tau(K_0)}(\bot), \text{"ciphertext"})
\end{align*}
$$
Finding a Large Gap (1)

- Recall: We assume $[[M]]_\Pi \not\approx [[N]]_\Pi$ to derive a contradiction with type-0 security of $\Pi$.

- Clearly $[[M]]_\Pi = [[M']]_\Pi$, since $M$ and $M'$ differ only by key renaming. Similarly, we have $[[N]]_\Pi = [[N']]_\Pi$.

- Therefore, we have $[[M']]_\Pi \not\approx [[N']]_\Pi$, i.e., there is a probabilistic polynomial-time adversary $A$ such that

$$
\lambda(\eta) = \left| Pr[x \leftarrow [[M']]_\Pi[\eta] : A(\eta, x) = 1] - Pr[x \leftarrow [[N']]_\Pi[\eta] : A(\eta, x) = 1] \right|
$$

is not negligible.

- This means that there is a constant $c \in \mathbb{N}$ and an infinite set $\mathcal{F} \subseteq \mathbb{N}$ such that $\lambda(\eta) > \eta^{-c}$ for all $\eta \in \mathcal{F}$. 

Finding a Large Gap (2)

**Definition (Functions $p_i$ and $q_j$)**

\[
p_i(\eta) = \Pr[x \overset{R}{\leftarrow} [M_i]_{\Pi[\eta]} : A(\eta, x) = 1] \quad \text{(for } 0 \leq i \leq n)\]

\[
q_j(\eta) = \Pr[x \overset{R}{\leftarrow} [N_j]_{\Pi[\eta]} : A(\eta, x) = 1] \quad \text{(for } 0 \leq j \leq m)\]

- Hence, since $M' = M_m$ and $N' = N_n$, we have

  \[
  \lambda(\eta) = |p_m(\eta) - q_n(\eta)|.
  \]

- We also have $p_0 = q_0$, since $M'$ and $N'$ yield the same pattern, hence

  \[
  \lambda = |(p_m - p_{m-1}) + (p_{m-1} - p_{m-2}) + \ldots + (p_1 - p_0) + (q_0 - q_1) + (q_1 - q_2) + \ldots + (q_{n-1} - q_n)|
  \]

- By the triangle inequality for $m + n$ summands, for all $\eta \in \mathcal{F}$:

  \[
  \exists i \in \{1, \ldots, n\}. \quad |p_i(\eta) - p_{i-1}(\eta)| \geq \lambda(\eta)/(m + n)
  \]

  \[
  \lor \quad \exists j \in \{1, \ldots, m\}. \quad |q_j(\eta) - q_{j-1}(\eta)| \geq \lambda(\eta)/(m + n)
  \]
Finding the Large Gap (3)

- By the triangle inequality for $m + n$ summands, for all $\eta \in \mathcal{F}$:

  \[
  \exists i \in \{1, \ldots, n\}. \quad |p_i(\eta) - p_{i-1}(\eta)| \geq \frac{\lambda(\eta)}{m + n}
  \]

  \[
  \lor \quad \exists j \in \{1, \ldots, m\}. \quad |q_j(\eta) - q_{j-1}(\eta)| \geq \frac{\lambda(\eta)}{m + n}
  \]

- Hence, since there is a finite number $(m + n)$ possible indices, there is an index $i \in \{1, \ldots, n\}$ such that for infinitely many $\eta \in \mathcal{F}$:

  \[
  |p_i(\eta) - p_{i-1}(\eta)| \geq \frac{\lambda(\eta)}{m + n}
  \]

  or there is an index $j \in \{1, \ldots, m\}$ such that for infinitely many $\eta \in \mathcal{F}$:

  \[
  |q_j(\eta) - q_{j-1}(\eta)| \geq \frac{\lambda(\eta)}{m + n}
  \]

- Let $i$ be such an index (the case for $j$ is analogous). This means that

  \[
  \varepsilon(\eta) = |p_i(\eta) - p_{i-1}(\eta)|
  \]

  is non-negligible (since $\lambda$ is non-negligible and $m + n$ is a constant).
Contradicting the Security of $\Pi$ (1)

Next, we construct an adversary $A_0$ from the adversary $A$ such that the following lemma holds (proved shortly).

Lemma (Relating the adversaries)

\[ p_i(\eta) = \Pr[y \overset{R}{\leftarrow} [M_i]_{\Pi[\eta]} : A(\eta, y) = 1] \]  
\[ = \Pr[k_0, k_i \overset{R}{\leftarrow} \mathcal{K}(\eta) : A_0^{\mathcal{E}_{k_i}(\cdot), \mathcal{E}_{k_0}(\cdot)}(\eta) = 1] \]  
\[ p_{i-1}(\eta) = \Pr[y \overset{R}{\leftarrow} [M_{i-1}]_{\Pi[\eta]} : A(\eta, y) = 1] \]  
\[ = \Pr[k_0 \overset{R}{\leftarrow} \mathcal{K}(\eta) : A_0^{\mathcal{E}_{k_0}(\bot), \mathcal{E}_{k_0}(\bot)}(\eta) = 1] \]

From the Lemma it follows that

\[ |(4) - (2)| = \text{Adv}_\Pi^0(A_0) = \varepsilon(\eta) = |(3) - (1)| \]

which is non-negligible.

Hence, we have derived a contradiction with the type-0 security of $\Pi$. 

Ch. Sprenger
The Soundness Theorem

Contradicting the Security of $\Pi$ (2)

Definition (Adversary $A^f,g(\eta)$; part 1)

The adversary $A^f,g(\eta)$ is defined by the following probabilistic algorithm:

\[
\begin{align*}
\tau & \xleftarrow{R} \text{Init}(\eta, M'); \\
y & \xleftarrow{R} \text{Conv}^f,g(\tau, M'); \\
b & \xleftarrow{R} A(\eta, y); \text{return } b
\end{align*}
\]

where $\text{Conv}^f,g(\tau, M')$ is defined on the next slide.

The Lemma requires that

\[
\begin{align*}
\Pr[y \xleftarrow{R} [M_i]_{\Pi[\eta]} : A(\eta, y) = 1] &= \Pr[k_0, k_i \xleftarrow{R} \mathcal{K}(\eta) : A^0_{\mathcal{E}_{k_i}(\cdot), \mathcal{E}_{k_0}(\cdot)}(\eta) = 1] \\
\Pr[y \xleftarrow{R} [M_{i-1}]_{\Pi[\eta]} : A(\eta, y) = 1] &= \Pr[k_0 \xleftarrow{R} \mathcal{K}(\eta) : A^0_{\mathcal{E}_{k_0}(\bot), \mathcal{E}_{k_0}(\bot)}(\eta) = 1]
\end{align*}
\]

We derive the following requirements on $\text{Conv}^f,g(\tau, M')$:

\[
\begin{align*}
\text{Convert}(\tau, M_i) &= \text{Conv}^{\mathcal{E}_{k_i}(\cdot), \mathcal{E}_{k_0}(\cdot)}(\tau, M') \\
\text{Convert}(\tau, M_{i-1}) &= \text{Conv}^{\mathcal{E}_{k_0}(\bot), \mathcal{E}_{k_0}(\bot)}(\tau, M')
\end{align*}
\]
Contradicting the Security of $\Pi$ (3)

**Definition (Adversary $A_{0}^{f,g}(\eta)$; part 2)**

- $Conv^{f,g}(\tau, K) = (\tau(K), ''key'')$
- $Conv^{f,g}(\tau, i) = (i, ''bool'')$
- $Conv^{f,g}(\tau, (M_1, M_2)) = (Conv^{f,g}(\tau, M_1), Conv^{f,g}(\tau, M_2), ''pair'')$
- $Conv^{f,g}(\tau, \{M_1\} K) = \begin{cases} (Conv^{f,g}(\tau, M_1), Conv^{f,g}(\tau, M_2), ''pair''), & \text{if } K \in \text{rec}(M') \cup \{K_1, \ldots, K_{i-1}\} \text{ then} \\ (E_{\tau(K)}(Conv^{f,g}(\tau, M_1)), ''ciphertext''), & \text{else if } K = K_i \text{ then} \\ (f(Conv^{f,g}(\tau, M_1)), ''ciphertext''), & \text{else}\end{cases}$
- $Conv^{f,g}(\tau, \{M_1\} K) = \begin{cases} (Conv^{f,g}(\tau, M_1), Conv^{f,g}(\tau, M_2), ''pair''), & \text{if } K \in \text{rec}(M') \cup \{K_1, \ldots, K_{i-1}\} \text{ then} \\ (E_{\tau(K)}(Conv^{f,g}(\tau, M_1)), ''ciphertext''), & \text{else if } K = K_i \text{ then} \\ (f(Conv^{f,g}(\tau, M_1)), ''ciphertext''), & \text{else}\end{cases}$

**Requirements on $Conv^{f,g}(\tau, M')$:**

- $Convert(\tau, M_i) = Conv^{E_{k_i}(\cdot), E_{k_0}(\cdot)}(\tau, M')$ \checkmark
- $Convert(\tau, M_{i-1}) = Conv^{E_{k_0}(\perp), E_{k_0}(\perp)}(\tau, M')$ \checkmark
Conclusion (Summary)

First step towards bridging the gap between the two views:

- Suitable design of the formal and computational models, so that a relation is possible.

- Soundness result: if messages in the formal view are indistinguishable, then so are their representations in the computational model.

⇒ Verification in the formal model carries over to verification in the computational model.

⇒ In this setting, there can’t be any “attacks” in the cryptographic model, that would not be possible in the formal model.
Conclusion (Extensions)

Still lots of challenging problems to solve:

- Active adversary, in particular interaction with honest agents.
  - “trace-mapping” approach [Micciancio, Warinschi & Cortier]
  - based on universal composability [Canetti & Herzog]
  - cryptographic library based on reactive simulatability framework [Backes, Pfitzmann & Waidner]

- More cryptographic primitives like Diffie-Hellman.

- Cryptographic systems for which tagging/typing is not possible.

- Weaker restrictions on the encryption scheme.
**Bibliography**


