Sécurité des protocoles cryptographiques : aspects logiques et calculatoires

Mathieu Baudet

Laboratoire Spécification et Vérification
(INRIA Futurs, CNRS, ENS Cachan)

Soutenance de thèse – 16 jan. 2007
New technologies (Internet, Wifi, cell phones) allow cheap worldwide communications.

Many services now available on the Internet:
- shopping,
- online auction (eBay, ...),
- account management (bank, phone company, ...),
- e-administration (tax payment, ...),
- <your favorite e-Business here>

Unfortunately, Internet was not designed for security.
... hence big efforts required to secure websites
• The attacker can
  – eavesdrop messages,
  – delete some of them,
  – send fake ones.
Modeling insecure networks

- The attacker can
  - eavesdrop messages,
  - delete some of them,
  - send fake ones.

→ How to securely communicate anyway?

In brief:

attacker \approx \text{network}
Cryptographic protocols

... are concurrent programs

- which communicate with the network
- and use cryptography:

- **symmetric** encryption

  $\begin{align*}
  K \\
  M \rightarrow \text{Enc.} \rightarrow \{M\}_K \rightarrow \text{Dec.} \rightarrow M
  \end{align*}$

- **asymmetric** encryption

  $\begin{align*}
  pk = pub(sk) \\
  M \rightarrow \text{Enc.} \rightarrow \{M\}_{pk} \rightarrow \text{Dec.} \rightarrow M
  \end{align*}$

- **signatures**

  $\begin{align*}
  sk \\
  M \rightarrow \text{Sign} \rightarrow [M]_{sk} \rightarrow \text{Check} \rightarrow \text{ok}
  \end{align*}$

- ... 

Unfortunately, designing secure protocols is not an easy task...
An example of logical attack

Denning-Sacco protocol:

0. $A \rightarrow B : A, \{ [k_{AB}]_{sk_A} \}_{pk_B}$
1. $B \rightarrow A : \{ secr_{AB} \}_{k_{AB}}$

Active attacker:

- **chooses** the interleaving of sessions,
- **controls** the network (may intercept, analyze, forge messages).
An example of logical attack

Denning-Sacco protocol:

0. $A \rightarrow B : A, \{ [k_{AB}]_{sk_A} \}_{pk_B}$
1. $B \rightarrow A : \{ secr_{AB} \}_{k_{AB}}$

An attack with 2 sessions:

0. $A \rightarrow I : A, \{ [k_{AI}]_{sk_A} \}_{pk_I}$
0’. $I(A) \rightarrow B : A, \{ [k_{AI}]_{sk_A} \}_{pk_B}$
1. $B \rightarrow I(A) : \{ secr_{AB} \}_{k_{AI}}$
Automatic analysis of protocols

- Based on symbolic (logical) models
  → introduced by Needham-Schroeder (1978) and Dolev-Yao (1983)

- Messages represented by terms of unbounded size

- Now highly automatized tools
  - bounded number of sessions (exact, typically co-NP)
    → constraint solving & symbolic model-checking
  - unbounded number of sessions (approximate)
    → tree automata, Horn clauses, typing systems...
Challenges in automatic verification

Not surprisingly, difficulties come from

• message abstraction, and
• the need for effective procedures.

We would like to handle

(1) more protocols
(2) more properties
(3) more attacks
Challenges in automatic verification (1)

Handling more protocols

- Free term algebras are OK for constructors & destructors, e.g. pairing, encryption (with integrity checking), signature.
- Other primitives require equational theories.
Challenges in automatic verification (1)

Handling more protocols

- Free term algebras are OK for constructors & destructors, e.g. pairing, encryption (with integrity checking), signature.

- Other primitives require equational theories. E.g.:
  - Exclusive OR: \((\text{Comon et al., Chevalier et al. in 2003})\)
    \[
    x \oplus y = y \oplus x \quad x \oplus x = 0 \\
    (x \oplus y) \oplus z = x \oplus (y \oplus z) \\
    x \oplus 0 = x
    \]
  - Surjective encryption (ciphers):
    \[
    \text{dec(}\text{enc(x, y), y)} = x \quad \text{enc(}\text{dec(x, y), y)} = x
    \]
    (Delaune-Jacquemard, among other primitives, in 2004)
Handling more security properties

- Most of existing results concern trace properties, e.g. simple secrecy and authentication.

- Modeling indistinguishability properties require an observational equivalence in a language of processes.
Handling more security properties

- Most of existing results concern trace properties, e.g. simple secrecy and authentication.

- Modeling indistinguishability properties require an observational equivalence in a language of processes.

- The applied pi-calculus, proposed in 2001 by M. Abadi and C. Fournet, is such a language, also featuring equational theories.
  → First decidability result for the passive case (i.e. static equivalence) in 2004 by M. Abadi and V. Cortier.
Handling more attacks

- **Symbolic models** automatized but *a priori* restricted to logical attacks

- **Computational (cryptographic) models** deal with arbitrary (efficient) adversary but require *a priori* hand-made, complex reduction proofs

Ideally, symbolic tools should provide cryptographic proofs.
Handling more attacks

- **Symbolic models** automatized but *a priori* restricted to logical attacks

- **Computational (cryptographic) models** deal with arbitrary (efficient) adversary but require *a priori* hand-made, complex reduction proofs

Ideally, symbolic tools should provide cryptographic proofs.

→ First **computationally sound** symbolic models:

- **Data indistinguishability** for symmetric encryption in 2000 (Abadi and Rogaway)

- **Active case** started in 2003 with Backes, Pfitzmann and Waidner’s cryptographic library.
Contributions of this thesis

- (1-2) First **decidability result** for an equivalence of processes in presence of equational theories.

- (3) First results of **computational soundness** for static equivalence.

Both results apply to **dictionary attacks** and contribute to clarify the “right” symbolic definition for it.

- (1) more protocols
- (2) more properties
- (3) more attacks
Outline

1. Introduction
2. Symbolic analysis of protocols
3. Constraint solving
4. Computational justification for a passive adversary
5. Conclusion
Dictionary attacks (a.k.a. guessing attacks)

http://www.thc.org/thc-hydra/
Dictionary attacks (a.k.a. guessing attacks)

**Definition (Lowe WITS’02)**

Dictionary attacks =

- weak secret (password) → exhaustive search feasible
- off-line verification test → “is this the right value?”

where off-line = no interaction with the network

On-line tests do not *undermine* security, but off-line ones do.

→ c.f. Unix’s shadow passwords
**Examples of dictionary attacks (1)**

**Handshake Protocol**

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$A \rightarrow B : {n}<em>{w</em>{AB}}$</td>
<td>Aims to authenticate principal $B$ from $A$’s viewpoint.</td>
</tr>
<tr>
<td>1</td>
<td>$B \rightarrow A : {n + 1}<em>{w</em>{AB}}$</td>
<td></td>
</tr>
</tbody>
</table>
Examples of dictionary attacks (1)

Handshake Protocol

0. \( A \rightarrow B : \{n\}_w^{AB} \) as \( m_1 \)
1. \( B \rightarrow A : \{n + 1\}_w^{AB} \) as \( m_2 \)

Aims to authenticate principal \( B \) from \( A \)'s viewpoint.

An off-line verif. test for shared password \( w_{AB} \):

\[ \text{dec}(m_1, x) + 1 \equiv \text{dec}(m_2, x) \]

Note:

- this case only requires a passive attacker (eavesdropper)
- password-based encryption impl. by keyed permutations
Examples of dictionary attacks (2)

“Enhanced” Kerberos Protocol, Gong SAC’93

0. \( A \rightarrow S : \left\{ A, B, n_1, n_2, \{ t_A \}_{wAS} \right\}_{pks}^a \)

1. \( S \rightarrow A : \left\{ n_1, k \oplus n_2 \right\}_{wAS}, \left\{ A, k, t_S \right\}_{wBS} \)

2. \( A \rightarrow B : \left\{ A, k, t_S \right\}_{wBS} \)
Examples of dictionary attacks (2)

"Enhanced" Kerberos Protocol, Gong SAC’93

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>$A \rightarrow S$ :</td>
<td>${A, B, n_1, n_2, {t_A}<em>{w</em>{AS}}}_{pks}$ as $m_1$</td>
</tr>
<tr>
<td>1.</td>
<td>$S \rightarrow A$ :</td>
<td>${n_1, k \oplus n_2}<em>{w</em>{AS}}, {A, k, t_S}<em>{w</em>{BS}}$ as $m_2$</td>
</tr>
<tr>
<td>2.</td>
<td>$A \rightarrow B$ :</td>
<td>${A, k, t_S}<em>{w</em>{BS}}$</td>
</tr>
</tbody>
</table>

Off-line test for $w_{AS}$: $\pi_1(\text{dec}(\pi_1(m_2), x)) = ? \pi_1(\text{dec}(\pi_1(m'_2), x))$

Off-line test for $w_{BS}$: $\pi_1(\text{dec}(\pi_2(m_2), y)) = ? A$
Modeling dictionary attacks

- Which data are weak? → given by the protocol
- Verification test?
  → distinguishes between two scenarios: wrong / right guess
- The general definition from Corin et al. [WISP’04] and Blanchet et al. [LICS’05] uses the observational equivalence of the applied pi-calculus.
- We proved that a stricter equivalence based on bi-processes suffices to characterize guessing attacks.
Symbolic analysis by example

### Handshake Protocol

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A → B</td>
<td>({n}<em>{k</em>{AB}})</td>
</tr>
<tr>
<td>1</td>
<td>B → A</td>
<td>({n+1}<em>{k</em>{AB}})</td>
</tr>
</tbody>
</table>

Is the following trace a **feasible** one?

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A → I(B)</td>
<td>({n}<em>{k</em>{AB}})</td>
</tr>
<tr>
<td>1</td>
<td>I(B) → A</td>
<td>({n+1}<em>{k</em>{AB}})</td>
</tr>
</tbody>
</table>

If yes → **attack** on authentication
### Symbolic analysis by example

<table>
<thead>
<tr>
<th>Step</th>
<th>Rule</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>$A \rightarrow I(B)$:</td>
<td>${n}<em>{k</em>{AB}}$</td>
</tr>
<tr>
<td>1.</td>
<td>$I(B) \rightarrow A$:</td>
<td>${n+1}<em>{k</em>{AB}}$</td>
</tr>
</tbody>
</table>

→ $I$’s computation $X_1$ must satisfy the constraints:

$$\exists x_1, \quad X_1[\{n\}_{k_{AB}}] = ? x_1 \quad \text{and} \quad \text{dec}(x_1, k_{AB}) = ? n + 1$$

without $I$’s “knowing” $n$ nor $k_{AB}$.
Symbolic analysis by example

0. \( A \rightarrow I(B) : \{ n \}^{k_{AB}} \)
1. \( I(B) \rightarrow A : \{ n + 1 \}^{k_{AB}} \)

\( I \)'s computation \( X_1 \) must satisfy the constraints:

\[ \exists x_1, \quad X_1[\{ n \}^{k_{AB}}] = \overline{?} x_1 \quad \text{and} \quad \text{dec}(x_1, k_{AB}) = \overline{?}_E n + 1 \]

where \( n, k_{AB} \) cannot occur in \( X_1 \).

Equations interpreted modulo the theory \( E \) of ciphers:

\[
\text{dec}(\{x\}_y, y) = x \quad \{ \text{dec}(x, y) \}_y = x
\]

The previous constraint system is unsatisfiable (i.e. has no solutions) \( \Rightarrow \) this interleaving infeasible \( \Rightarrow \) no attack.
More generally, a finite number of sessions entails a finite number of interleavings, thus of constraint systems to verify.

Trace properties correspond to (un)satisfiability problems on constraint systems.

This works for any equational theory — as long as we can solve the constraint systems...

→ Can we do the same for equivalence properties?
Intruder constraint systems

Each interleaving is mapped to a system $\Sigma(X_1 \ldots X_n)$:

$$\exists x_1 \ldots x_m, \begin{cases} 
X_1[t_1 \ldots t_{a_1}] = ? x_1 \quad u_1 =? _E u'_1 \\
\quad \cdots \cdots \cdots \cdots \\
X_m[t_1 \ldots t_{a_m}] = ? x_m \quad u_n =? _E u'_n 
\end{cases}$$

with several “cryptographic” regularity conditions:

- “The attacker’s knowledge $t_1, \ldots, t_{a_i}$ increases.”
- “Messages $t_j$ depend only on previous attacker outputs $x_i$.”
Standard Intruder constraint systems

Each **interleaving** is mapped to a system $\Sigma(X_1 \ldots X_n, X, Y)$:

$$
\exists x_1 \ldots x_m, x, y, \begin{cases}
X_1[t_1 \ldots t_{a_1}] = ? x_1 & u_1 = ? E u'_1 \\
\vdots & \vdots \\
X_m[t_1 \ldots t_{a_m}] = ? x_m & u_n = ? E u'_n \\
X[t_1 \ldots t_{a_m}] = ? x & x = ? E y \\
Y[t_1 \ldots t_{a_m}] = ? y 
\end{cases}
$$

Let $X, Y, x, y$ be fresh variables.

$\rightarrow$ the extra equation models an **off-line test** of the intruder.
Security against off-line dictionary attacks

Let $s_0$ and $s_1$ model the right and the wrong value of the weak secret.

For each interleaving, let $\Sigma_i(X_1 \ldots X_n, X, Y)$ be

$$
\exists \tilde{x}_n, x, y, \quad \begin{cases} 
X_1[t_1 \ldots t_{a_1}] = ? \quad x_1 \quad u_1 = ? \quad u_1' \\
\ldots \\
X_m[t_1 \ldots t_{a_m}] = ? \quad x_m \quad u_n = ? \quad u_n' \\
X[t_1 \ldots t_{a_m}, s_i] = ? \quad x \quad x = ? \quad y \\
Y[t_1 \ldots t_{a_m}, s_i] = ? \quad y
\end{cases}
$$
Let $s_0$ and $s_1$ model the right and the wrong value of the weak secret.

For each interleaving, we must check that the two augmented systems $\Sigma_i$ have the same sets of solutions.

\[ \rightarrow \text{Equivalence between two second-order } E\text{-unification problems.} \]
Convergent subterm theories

\( E \) is convergent subterm iff it is generated by a convergent rewriting system \( \mathcal{R} \) such that for every rule \( l \rightarrow r \) in \( \mathcal{R} \), either

1. \( r \) is a subterm of \( l \), or
2. \( r \) is an \( \mathcal{R} \)-reduced term (say a constant).

Examples:

- Encryption(s)  
  \( \text{pdec( penc(} x, \text{ pub(} y), z)\text{, y) = x} \)

- Signature(s)  
  \( \text{check( sig(} x, y, z)\text{, pub(} y)\text{)) = ok} \)  
  \( \text{(no equation)} \)

- Hash function
- Idempotency  
  \( \text{f(} f(x)\text{)} = f(x) \)

- Involution  
  \( \text{i(} i(x)\text{)} = x \)

- . . .
Our problem boils down to generalizing previous work of S. Delaune and F. Jacquemard [CCS’04] and M. Abadi and V. Cortier [ICALP’04].
Contributed solving procedure

- Based on a set of transformation rules on extended constraint systems.
- Variables instanciated “on demand”.
- Solves equality constraints by narrowing.
- Main task is to compute (generating) sets of deducible terms and visible equations to handle the 2nd-order part.
How to represent sets of solutions?

Example: (passive case and syntactic equality)

\[ X[k, h(0, k)] =? u \]

where \( k \) is a secret and \( u \) is any ground term.

- Deducible terms are built (here simply) upon: \( k, h(0, k) \).
- Let us label \( k \) as \( w_1 \) and \( h(0, k) \) as \( w_2 \).
- Visible equations are generated by \( \Psi = \{ h(0, w_1) = w_2 \} \).

Fact

Let \( \theta_0 \) be any solution:
\[
(X\theta)\{w_1 \mapsto k, \; w_2 \mapsto h(0, k)\} =? u.
\]

The set of all solutions is

\[ \{\theta \mid \theta =_{\Psi} \theta_0\} \]
Main theorem

Theorem (Baudet [CCS'05])

For every convergent subterm theory $E$, the satisfiability of intruder constraint systems is NP-decidable. So is the non-equivalence of standard intruder constraint systems.
Corollary

For a bounded number of sessions, the security of protocols modeled by a convergent subterm theory $E$, with respect to

- trace properties (simple secrecy, authentication) and
- off-line dictionary attacks,

is co-NP-decidable.

Adding disequality tests is harmless as far as trace properties are concerned.

We prove the whole biprocess-based equivalence decidable → useful for strong secrecy.
Constraint systems for pure eavesdropper

An execution of the protocol corresponds to a system $\Sigma(X, Y)$:

$$\exists x, y, \left\{ \begin{array}{c} X[t_1 \ldots t_{am}] =? x \\ Y[t_1 \ldots t_{am}] =? y \end{array} \right. \text{ and } x =?= E y$$

where for all $1 \leq j \leq a_i$, $\text{var}(t_j) = \emptyset$.

**Notation**

We call $[t_1 \ldots t_{am}]$ a frame. A (more) standard notation is:

$$\varphi = \nu k_1 \ldots k_p \cdot \{ x_1 = t_1, \ldots, x_{am} = t_{am} \}$$

where $k_1 \ldots k_p$ are the private constants ("names", modeling secret values) in $t_1 \ldots t_{am}$.
Static Equivalence

Definition (static equivalence)

\[ \varphi_1 \approx_E \varphi_2 \text{ if for every valid test } M =_E^? N, \]

\[ M\varphi_1 =_E N\varphi_1 \text{ iff } M\varphi_2 =_E N\varphi_2 \]

Two frames \( \varphi \) and \( \varphi' \) correspond to equivalent intruder constraint systems iff they are statically equivalent.

Example:

\[ \nu n. \{ x = \{ n \}_{c_0}, y = \{ n+1 \}_{c_0} \} \not\approx_E \nu n. \{ x = \{ n \}_{c_1}, y = \{ n+1 \}_{c_1} \} \]

because of \( \text{dec}(x, c_0) + 1 =_E^? \text{dec}(y, c_0) \).
Computational soundness of static equivalence

- Does it correspond to cryptographic indistinguishability?

- In \((Baudet, Cortier, Kremer [ICALP’05])\), we studied
  - a general soundness criterion,
  - deterministic surjective encryption, and
  - the case of pure exclusive Or.

- What about other kind of encryptions? passwords?
Concrete implementation

- Complexity parameter $\eta$
- Assume an (efficient) implementation for each function symbol, and random generators for names.
- Terms $t$ mapped to (distributions over) bit-strings $[t]_\eta$
- We may restrict terms to well-sorted ones

**Definition (indistinguishability)**

$[\varphi_1] \approx [\varphi_2]$ if $\text{Adv}^{\text{IND}}(A, [\varphi_1]_\eta, [\varphi_2]_\eta)(\eta) =$

$$\mathbb{P} [\phi_1 \leftarrow [\varphi_1]_\eta; A(\eta, \phi_1) = 1] - \mathbb{P} [\phi_2 \leftarrow [\varphi_2]_\eta; A(\eta, \phi_2) = 1]$$

is a negligible function of $\eta$. 
### Sorts, symbols and equational theory of interest

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>[::=]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SKey$</td>
<td>symmetric keys</td>
</tr>
<tr>
<td>$EKey$</td>
<td>(public) encryption keys</td>
</tr>
<tr>
<td>$DKey$</td>
<td>(private) decryption keys</td>
</tr>
<tr>
<td>$Data$</td>
<td>passwords and other data</td>
</tr>
<tr>
<td>$Coins$</td>
<td>coins for encryption</td>
</tr>
<tr>
<td>$Pair[\tau_1, \tau_2]$</td>
<td>pairs of messages</td>
</tr>
<tr>
<td>$SCipher[\tau]$</td>
<td>symmetric encryptions</td>
</tr>
<tr>
<td>$ACipher[\tau]$</td>
<td>asymmetric encryptions</td>
</tr>
</tbody>
</table>
Sorts, symbols and equational theory of interest

\[
\begin{align*}
\text{enc}_\tau &: \tau \times \text{Data} \rightarrow \tau \\
\text{dec}_\tau &: \tau \times \text{Data} \rightarrow \tau \\
\text{penc}_\tau &: \tau \times \text{EKey} \times \text{Coins} \rightarrow \text{ACipher}[\tau] \\
\text{pdec}_\tau &: \text{ACipher}[\tau] \times \text{DKey} \rightarrow \tau \\
\text{pub} &: \text{DKey} \rightarrow \text{EKey} \\
\text{pdec\_success}_\tau &: \text{ACipher}[\tau] \times \text{DKey} \rightarrow \text{Data} \\
\text{senc}_\tau &: \tau \times \text{SKey} \times \text{Coins} \rightarrow \text{SCipher}[\tau] \\
\text{sdec}_\tau &: \text{SCipher}[\tau] \times \text{SKey} \rightarrow \tau \\
\text{sdec\_success}_\tau &: \text{SCipher}[\tau] \times \text{SKey} \rightarrow \text{Data} \\
\text{pair}_{\tau_1, \tau_2} &: \tau_1 \times \tau_2 \rightarrow \text{Pair}[\tau_1, \tau_2] \\
\text{fst}_{\tau_1, \tau_2}, \text{snd}_{\tau_1, \tau_2} &: \text{Pair}[\tau_1, \tau_2] \rightarrow \tau_2 \\
0, 1, w, c_0, c_1 \ldots &: \text{Data}
\end{align*}
\]
Sorts, symbols and **equational theory of interest**

\[
\begin{align*}
\text{dec}(\text{enc}(x, y), y) &= x \\
\text{enc}(\text{dec}(x, y), y) &= x \\
\text{pdec}(\text{penc}(x, \text{pub}(y), z), y) &= x \\
\text{pdec\_success}(\text{penc}(x, \text{pub}(y), z), y) &= 1 \\
\text{sdec}(\text{senc}(x, y, z), y) &= x \\
\text{sdec\_success}(\text{senc}(x, y, z), y) &= 1 \\
\text{fst}(\text{pair}(x, y)) &= x \\
\text{snd}(\text{pair}(x, y)) &= y \\
\text{pair}(\text{fst}(x), \text{snd}(x)) &= x
\end{align*}
\]

(sorts omitted)

(Note that this theory is subterm convergent.)
Computational soundness of $\approx_E$

A reduced frame $\varphi$ is well-formed if

- it contains no destructors (dec, pdec, ...),
- encryption keys are either names $k$, of the form $\text{pub}(k)$, or constants of sort $\text{Data}$,
- coins are “fresh” names,
- $\varphi$ has no key encryption cycles,
- for every subterm $\text{enc}(T, k)$, $T$ contains neither constants $w, c_0, \ldots$ nor subterms $\text{enc}(S, 0), \text{enc}(S, 1)$.

**Theorem (Abadi, Baudet, Warinschi [FOSSACS’06])**

*In any secure implementation, for every well-formed frames $\varphi_1$ and $\varphi_2$, $\varphi_1 \approx_E \varphi_2$ implies $[\varphi_1] \approx [\varphi_2]$.***
Application to dictionary attacks

**Corollary**

In any secure implementation, for every well-formed frame $\varphi$, $\varphi\{w \mapsto c_0\} \approx_E \varphi\{w \mapsto c_1\}$ implies that $w$ is computationally hidden in $\varphi$: for every (effective) sequences $\kappa_0$ and $\kappa_1$,

$$\llbracket \varphi \rrbracket_{w \mapsto \kappa_0} \approx \llbracket \varphi \rrbracket_{w \mapsto \kappa_1}$$

Generalizes to multiple passwords.
Outline

1. Introduction
2. Symbolic analysis of protocols
3. Constraint solving
4. Computational justification for a passive adversary
5. Conclusion
Summary

- Symbolic analysis of **trace properties** and **off-line dictionary attacks** based on constraint solving. → More generally we studied an equivalence of processes based on bi-processes.

- The procedure works for protocols specified by any convergent subterm theory $E$ (e.g. encryptions + pair + signatures + hash...).

- **Computational justification** in the case of data indistinguishability for several kinds of encryption.
Further work

• More equational theories (XOR, homomorphism, blind signatures...)

• More expressive observational equivalences: may-testing, barbed-congruence of the applied-pi calculus
  ⇒ on-going work by S. Delaune, S. Kremer and M. Ryan

• Computational justification in the active case?
Thanks!
Secure implementation: symmetric encryption

Let $\tau \in T_{senc}$, and $A = (A_1, A_2)$ be 2-stage adversary.

- $k \leftarrow R \mathcal{K}^s(\eta)$;
- $A_1$ is provided access to an oracle $\mathcal{E}^s(\cdot, k)$;
- then $A_1$ outputs a challenge message $m^* \in [\tau]_\eta$ together with some state information $st$;
- a bit $b \leftarrow R \{0, 1\}$ is selected at random; if $b = 0$, we let $c \leftarrow R \text{“SCipher”} \| \tau \| \mathcal{E}^s(m^*, k)$; otherwise, we let $c \leftarrow R [\text{SCipher}[\tau]]_\eta$;
- $A_2$ is given $c$ and $st$, and outputs a bit $b'$.
- $A$ is successful if $b' = b$.

$$\text{Adv}_{\Pi^s, A}^\tau(\eta) = \Pr[A \text{ is successful}] - \frac{1}{2}$$
Secure implementation: asymmetric encryption

Let \( \tau \in T_{\text{penc}} \), and \( A = (A_1, A_2) \) be 2-stage adversary.

- \((pk, sk) \xleftarrow{R} K^a(\eta)\);
- \(A_1\) is given \( pk\);
- then \( A_1\) outputs a challenge message \( m^* \in \llbracket \tau \rrbracket_\eta \) together with some state information \( st\);
- a bit \( b \xleftarrow{R} \{0, 1\} \) is selected at random; if \( b = 0 \), we let \( c \xleftarrow{R} \text{"ACipher" } \| \tau \| E^a(m^*, pk)\); otherwise, we let \( c \xleftarrow{R} \llbracket \text{ACipher}[\tau] \rrbracket_\eta\);
- \(A_2\) is given \( c\) and \( st\), and outputs a bit \( b'\).
- \(A\) is successful if \( b' = b\).

\[
\text{Adv}^\tau_{\Pi^a, A}(\eta) = \text{Pr}[A\text{ is successful}] - \frac{1}{2}
\]
Secure implementation: password encryption (1)

We require $T_{\text{enc}} \cap \{Pair[\tau_1, \tau_2]\} = \emptyset$. Let $\tau \in T_{\text{enc}}$, and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be 2-stage adversary.

- $k \leftarrow R \mathcal{K}(\eta)$;
- $\mathcal{A}_1$ is provided access to an oracle $\mathcal{E}(\cdot, k)$;
- then $\mathcal{A}_1$ outputs a challenge message $m^* \in \llbracket \tau \rrbracket_\eta$ together with some state information $st$;
- a bit $b \leftarrow R \{0, 1\}$ is selected at random; if $b = 0$, we let $c \leftarrow R \mathcal{E}(m^*, k)$; otherwise, we let $c \leftarrow R \llbracket \tau \rrbracket_\eta$;
- $\mathcal{A}_2$ is given $c$ and $st$, and outputs a bit $b'$.
- $\mathcal{A}$ is successful if $b' = b$, and the challenge message $m^*$ is different from all the messages $m$ submitted by $\mathcal{A}$ to the encryption oracle.

$$\text{Adv}^\tau_{\text{RoR}, \Pi, \mathcal{A}}(\eta) = \Pr[\text{$\mathcal{A}$ is successful}] - \frac{1}{2}$$
Secure implementation: password encryption (2)

Let $\tau \in T_{enc}$, and $A = (A_1, A_2)$ be 2-stage adversary.

- $A_1$ outputs a key $k \in \{0, 1\}^{\alpha_1(\eta)}$ and some state information $st$;
- a bit $b \leftarrow \{0, 1\}$ is selected at random; if $b = 0$, we let $m \leftarrow \mathbb{R}^{[\tau]} \eta$ and $c = \mathcal{E}(m, k)$; otherwise, we let $c \leftarrow \mathbb{R}^{[\tau]} \eta$;
- $A_2$ is given $c$ and $st$, and outputs a bit $b'$.
- $A$ is successful if $b' = b$.

$$\text{Adv}^\tau_{pwd, \pi, A}(\eta) = \Pr[A \text{ is successful}] - \frac{1}{2}.$$