Secrecy by typing in the computational model

Stéphanie Delaune  Clément Hérouard  Joseph Lallemand

IRISA, CNRS & Univ. Rennes, France
Part 1: Squirrel
Verification of protocols: two families of models

80’s

**Symbolic model**
- Abstract terms
  - Perfect primitives
    \[ \text{dec}(\{m\}_k, k) = m \]
  - Automation

**Computational model**
- Turing machines
  - Cryptographic games
    - IND-CPA and INT-CTXT
  - Automation

**Tools**
- ProVerif
- Deepsec
- TypeQ
- Tamarin
- CryptoVerif
- EasyCrypt
- OWL
- Squirrel

Delaune, Hérouard, Lallemand
Secrecy by typing
Verification of protocols: two families of models

80's

Symbolic model

Computational model

Computationally Complete Symbolic Attacker
CCSA

Term \( t \) → Machine \([t]\)

Squirrel

2014
Squirrel’s logic

Wide Mouthed Frog protocol:

\[ A \rightarrow S : a, \{ b, k_{ab} \}_{k_a} \]
\[ S \rightarrow B : \{ a, k_{ab} \}_{k_b} \]

3 actions:

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Server</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I[i, j, k] )</td>
<td>( S[i, j, k] )</td>
<td>( R[i, j, k] )</td>
</tr>
</tbody>
</table>
Squirrel’s logic

Wide Mouthed Frog protocol:

\[ A \rightarrow S : a, \{ b, k_{ab} \}_{k_a} \]
\[ S \rightarrow B : \{ a, k_{ab} \}_{k_b} \]

3 actions:

Initiator \( I[i, j, k] \)
Server \( S[i, j, k] \)
Responder \( R[i, j, k] \)

Indices:

\( i \): Initiator
\( j \): Responder
\( k \): Session
**Squirrel’s logic**

**Wide Mouthed Frog protocol:**

\[ A \rightarrow S : a, \{ b, k_{ab} \}_{k_a} \]
\[ S \rightarrow B : \{ a, k_{ab} \}_{k_b} \]

3 actions:

- **Initiator** \( I[i, j, k] \)
- **Server** \( S[i, j, k] \)
- **Responder** \( R[i, j, k] \)

In each action:
- Output
- Condition
- States’ updates

Output:
\[ \text{senc}((\\text{fst}(\text{input}@S[i, j, k]), \text{snd}(\text{sdec}(\text{snd}(\text{input}@S[i, j, k]), k[i])))\), k[j], r[i, j, k]) \]
Different notions of secrecy

**Secrecy:**
The attacker cannot find the value $s$.

$$\forall f, f(frame@\tau) = s$$

**Strong secrecy:**
The attacker cannot distinguish the value $s$ and a fresh nonce $n$

$$frame@\tau, s \sim frame@\tau, n$$
Different notions of secrecy

**Secrecy:**
The attacker cannot find the value $s$.

$$\not\exists f, f(frame@\tau) = s$$

**Strong secrecy:**
The attacker cannot distinguish the value $s$ and a fresh nonce $n$

$$frame@\tau, s \sim frame@\tau, n$$
Part 2: Typing for security
**Types for security**

**Principle:** Over-approximate a value by a type

\[
x : \text{Msg} \quad y : \text{Msg} \\
\langle x, y \rangle : \text{Msg}
\]
Types for security

**Principle:** Over-approximate a value by a type

\[
x : \text{Msg} \quad y : \text{Msg} \quad \langle x, y \rangle : \text{Msg}
\]

Types for secrecy (with symmetric encryption):

- **Low:** Public
- **High:** Secret
- **SK[T]:** Symmetric key for type \( T \)
- ...
Related Work: Type systems have been used
- In many symbolic models (Focardi & Maffei, 2011)
- In the computational model in OWL (Gancher et al., 2023)
Types for security

**Related Work:** Type systems have been used
- In many symbolic models (Focardi & Maffei, 2011)
- In the computational model in OWL (Gancher et al., 2023)

**Goal**
Design a type system for secrecy for Squirrel’s logic (CCSA)
Part 3: Contributions
Contributions

1. Design of the type system

2. Soundness result

3. Case studies

4. Asymmetric encryption
Contributions

1 Design of the type system
   \[ \Gamma \vdash m : T \]

2 Soundness result

3 Case studies

4 Asymmetric encryption
Typing rules

Γ; R ⊢ t : T

Types of names/variables/states

Set of randoms

Term

Type
Typing rules

Types of
\[ \Gamma; R \vdash t : T \]

names/variables/states

Set of randoms

Type

Term

Types:

- Msg
- High; Low
- Bool; Cte(c)
- T + T
- T × T
- SK[T]
Typing rules

Types of names/variables/states

Set of randoms

Type

Term

Zeros:

\[ \Gamma; R \vdash t : \text{Msg} \]

\[ \Gamma; R \vdash \text{zeros}(t) : \text{Low} \]

Pair:

\[ \Gamma; R_1 \vdash t_1 : T_1 \]

\[ \Gamma; R_2 \vdash t_2 : T_2 \]

\[ \Gamma; R_1 \uplus R_2 \vdash \langle t_1, t_2 \rangle : T_1 \times T_2 \]
Typing rules

\[
\begin{align*}
\Gamma; R \vdash t : T & \quad \Gamma(k) = SK[T] \\
\Gamma; R \sqcup \{r\} \vdash \text{senc}(t, k[j], r[i]) : \text{Low}
\end{align*}
\]

Encryption:

\[
\begin{align*}
\Gamma; R \vdash t : T & \quad \Gamma(k) = SK[T] \\
\Gamma; R \vdash \text{sdec}(t, k[j]) : T + \text{Cte(fail)}
\end{align*}
\]

Decryption:
Contributions

1 Design of the type system

2 Soundness result
   Soundness
   If $\Gamma \vdash t : \text{Low}$ and $\Gamma \vdash s : \text{High}$
   Then a computational attacker cannot deduce $[s]$ from $[t]$

3 Case studies

4 Asymmetric encryption
Proof sketch

\[
\begin{align*}
S_{\text{dec}} & \\
S_{\text{enc}} & \text{Pair} \\
\text{Zeros} & 
\end{align*}
\]
Proof sketch

Out  Frame  Cond  In  Exec
State  Eq-Ind
Assign  Var  Sdec  Break-Sum
If-False  If-True  Fst  Snd
Name  Pair  Senc  If
Fun-Msg  Fun-Low  Sub-Typing  Zeros
Cst-∞  Cst-0
Eq

Problems of the base system:
- Decryption
- Some rules modify the environment
- Some rules do not type all subterms

Properties of the restricted system:
- No decryption rule
- If a term types, all subterms type in the same environment, keys and randoms are well-used, its value is computable by a PPTM with oracles
- In a Low term, if a subterm is High, it is in a boolean, an encryption, or a zeros
Proof sketch

Meta-logic system

- Out
- Frame
- Eq-Ind
- Cond
- In
- Exec

- Assign
- Var
- Sdec
- Break-Sum
- Fst
- Fst
- Snd

- Name
- Pair
- If
- Senc
- Fun-Msg
- Fun-Low
- Sub-Typing
- Zeros
- Cst-∞
- Cst-0
- Eq

Problems of the base system:
- Decryption
- Some rules modify the environment
- Some rules do not type all subterms

Properties of the restricted system:
- No decryption rule
- If a term types, all subterms type in the same environment,
  keys and randoms are well-used,
  its value is computable by a PPTM with oracles
- In a Low term, if a subterm is High, it is in a boolean, an encryption, or a zeros
Proof sketch

Meta-logic system

Macros and indices

Assign  Var  Sdec  Break-Sum  Fst
If-False  If-True  Snd

If  Name  Pair  Senc  If
Fun-Msg  Fun-Low  Sub-Typing  Zeros  Cst-∞
Cst-0  Eq

Delaune, Hérouard, Lallemand  Secrecy by typing  14 / 22
Proof sketch

Meta-logic system

Macros and indices

Base logic system

Assign
If-False
Sdec
break-Sum
Fst
Snd
Name
Pair

Senc
If
Fun-Msg
Fun-Low
Sub-Typing
Zeros
Cst-∞
Cst-0
Eq

Properties of the restricted system:
- No decryption rule
- If a term types, all subterms type in the same environment,
- keys and randoms are well-used,
- its value is computable by a PPTM with oracles

Induction on the trace

Problems of the base system:
- Decryption
- Some rules modify the environment
- Some rules do not type all subterms

Delaune, Hérouard, Lallemand

Secrecy by typing

14 / 22
Proof sketch

Meta-logic system

Macros and indices

Base logic system

Assign Var Sdec Break-Sum Fst
If-False If-True Snd

Name Pair Senc If
Fun-Msg Fun-Low Sub-Typing
Zeros Cst-∞ Cst-0 Eq

Problems of the base system:
- Decryption
- Some rules modify the environment
- Some rules do not type all subterms
Proof sketch

Meta-logic system
  Macros and indices

Base logic system
  Destructors and variables
  Other rules

Problems of the base system:
- Decryption
- Some rules modify the environment
- Some rules do not type all subterms
Proof sketch

Meta-logic system
  Macros and indices

Base logic system
  Destructors and variables

Restricted system
  Other rules

Problems of the base system:
- Decryption
- Some rules modify the environment
- Some rules do not type all subterms

Properties of the restricted system:
- No decryption rule
- If a term types,
  all subterms type in the same environment,
  keys and randoms are well-used,
  its value is computable by a PPTM with oracles
- In a Low term, if a subterm is High, it is in
  a boolean, an encryption, or a zeros
Proof sketch

Meta-logic system
  Macros and indices

Base logic system
  Destructors and variables

Restricted system
  Other rules

Soundness

Induction on the trace

INT-CTX

IND-CPA
Use of the theorem

**Soundness**

If $\Gamma \vdash t : \text{Low}$ and $\Gamma \vdash s : \text{High}$
Then a computational attacker cannot deduce $[s]$ from $[t]$

If a protocol is well typed in $\Gamma; R$
If a term $t$ type $\text{High}$
The attacker cannot find $[t]$ with the frame of the protocol
Use of the theorem

**Soundness**

If $\Gamma \vdash t : \text{Low}$ and $\Gamma \vdash s : \text{High}$
Then a computational attacker cannot deduce $[s]$ from $[t]$.

If a protocol is well typed in $\Gamma; R$
If a term $t$ type $\text{High}$
The attacker cannot find $[t]$ with the frame of the protocol.

In each action:
- Output types $\text{Low}$
- Condition types $\text{Bool}$
- States types as indicated in $\Gamma$
Contributions

1 Design of the type system

2 Soundness result

3 Case studies

4 Asymmetric encryption
# Case studies

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>no tag</th>
<th>tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wide Mouth Frog</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Denning Sacco</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Otways-Rees</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Needham-Schroeder*</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Yahalom*</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Yahalom-Paulson*</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Mechanism 6◊</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Mechanism 9◊</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Mechanism 13◊</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>

◊: ISO/IEC 11770 standard part II  
* : Without last message
Focus on Wide Mouth Frog

Protocol:

\[ A \to S : a, \{b, k_{ab}\}_{k_a} \]
\[ S \to B : \{a, k_{ab}\}_{k_b} \]

Scenario with **dishonest agents**:

7 actions $\to$ 7 outputs and conditions to type.
Focus on Wide Mouth Frog

Protocol:

\[ A \rightarrow S : a, \{b, k_{ab}\}_{k_a} \]
\[ S \rightarrow B : \{a, k_{ab}\}_{k_b} \]

Scenario with **dishonest agents**:
- 7 actions $\rightarrow$ 7 outputs and conditions to type.

Result:
- If A send $k_{ab}$ to an honest agent $k_{ab}$ is secret.
- If B receive $k_{ab}$ from an honest agent $k_{ab}$ is secret.
Contributions

1 Design of the type system

2 Soundness result

3 Case studies

4 Asymmetric encryption
New rules for IND-CCA2 asymmetric encryption

Public key: \( PK \)
\[
\Gamma(k) = AK[T]
\]
\[
\Gamma; R \vdash pk(k[j]) : Low
\]

Encryption: \( A_{enc} \)
\[
\Gamma; R \vdash t : T \quad \Gamma(k) = AK[T]
\]
\[
\Gamma; R \uplus \{r\} \vdash a_{enc}(t, pk(k[j]), r[i]) : Low
\]

Decryption: \( A_{dec} \)
\[
\Gamma; R \vdash t : Low \quad \Gamma(k) = AK[T]
\]
\[
\Gamma; R \vdash a_{dec}(t, k[j]) : T + Low
\]
New rules for IND-CCA2 asymmetric encryption

Public key: PK

Encryption: Aenc

Decryption: Adec

Meta-logic system
Macros and indices

Base logic system
Destructors
Adec

Restricted system
Other rules
PK
Aenc
Adec*

Soundness

No change
Proof without crypto reduction
IND-CCA2

Delaune, Hérouard, Lallemand
Case studies for asymmetric encryption

**Needham-Schroeder-Lowe:**

✔ (partial)

**ISO/IEC 11770 standard part II - Mechanism 6:**

✔ (partial)
Conclusion and ongoing work

Conclusion:

▶ A type system for secrecy in a computational model
  Symmetric/asymmetric encryption
▶ Soundness proof

Ongoing work:

▶ Add primitives
  hash function, signature...
▶ Key establishment protocol
  Key usability
▶ Integration in Squirrel