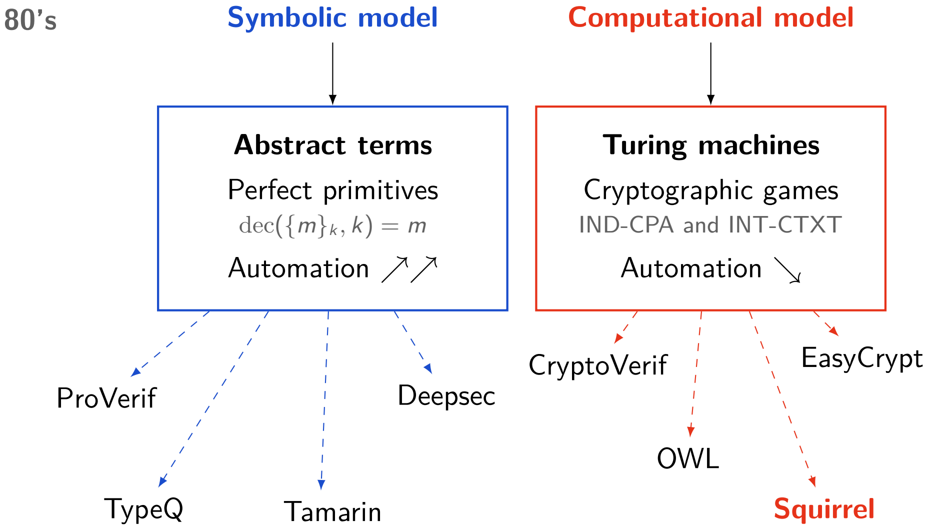
Secrecy by typing in the computational model

Stéphanie Delaune **Clément Hérouard** Joseph Lallemand

IRISA, CNRS & Univ. Rennes, France

Part 1: Squirrel

Verification of protocols: two families of models



Verification of protocols: two families of models

80's **Symbolic model** **Computational model**

2014 **Computationally Complete Symbolic Attacker**

CCSA

Term t $\rightarrow$ Machine $[\![t]\!]$

Squirrel

Squirrel's logic

Wide Mouthed Frog protocol:

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

3 actions:

Initiator	Server	Responder
$I[i, j, k]$	$S[i, j, k]$	$R[i, j, k]$

Squirrel's logic

Wide Mouthed Frog protocol:

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

3 actions:

Initiator	Server	Responder
$I[i, j, k]$	$S[i, j, k]$	$R[i, j, k]$

Indices:
i: Initiator
j: Responder
k: Session

Squirrel's logic

Wide Mouthed Frog protocol:

$A \rightarrow S : a, \{b, k_{ab}\}_{k_a}$

$S \rightarrow B : \{a, k_{ab}\}_{k_b}$

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

3 actions:

Initiator	Server	Responder
$I[i,j,k]$	$S[i,j,k]$	$R[i,j,k]$

Output:

$\mathsf{senc}(\langle \mathsf{fst}(\mathsf{input}@S[i,j,k]), \mathsf{snd}(\mathsf{sdec}(\mathsf{snd}(\mathsf{input}@S[i,j,k]), k[i]))\rangle, k[j], r[i,j,k])$

Different notions of secrecy

Secrecy:
The attacker cannot find the value s.

$$\nexists f, f(\text{frame@}\tau) = s$$

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

$$\text{frame@}\tau, s \sim \text{frame@}\tau, n$$

Different notions of secrecy

Secrecy:
The attacker cannot find the value s.

$$\nexists f, f(\text{frame@}\tau) = s$$

✗

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

$$\text{frame@}\tau, s \sim \text{frame@}\tau, n$$

✓

Part 2: Typing for security

Types for security

Principle: Over-approximate a value by a type

$$\frac{x : \mathrm{Msg} \qquad y : \mathrm{Msg}}{\langle x, y \rangle : \mathrm{Msg}}$$

Types for security

Principle: Over-approximate a value by a type

$$\frac{x : \mathrm{Msg} \qquad y : \mathrm{Msg}}{\langle x, y \rangle : \mathrm{Msg}}$$

Types for secrecy (with symmetric encryption):

- Low: Public
- High: Secret
- $\mathrm{SK[T]}$: Symmetric key for type T
- ...

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)

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

$$\Gamma \vdash m : \mathrm{T}$$

2 Soundness result

3 Case studies

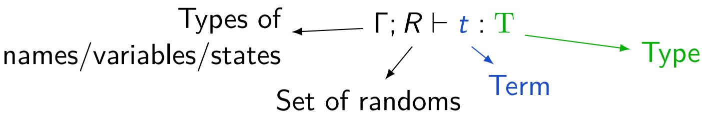
4 Asymmetric encryption

Typing rules

$$\Gamma; R \vdash t : \mathrm{T}$$

- Γ: Types of names/variables/states
- R: Set of randoms
- t: Term
- T: Type

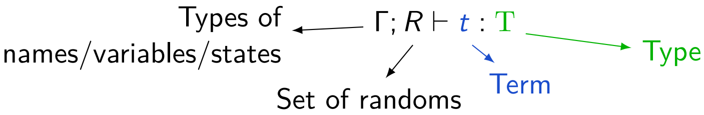
Typing rules



Types:

- Msg
- High; Low
- Bool; $\mathrm{Cte(c)}$
- $\mathrm{T} + \mathrm{T}$
- $\mathrm{T} \times \mathrm{T}$
- $\mathrm{SK[T]}$

Typing rules



Zeros: $$\frac{\Gamma; R \vdash t : \mathrm{Msg}}{\Gamma; R \vdash \mathsf{zeros}(t) : \mathrm{Low}}$$

Pair: $$\frac{\Gamma; R_1 \vdash t_1 : \mathrm{T}_1 \qquad \Gamma; R_2 \vdash t_2 : \mathrm{T}_2}{\Gamma; R_1 \sqcup R_2 \vdash \langle t_1, t_2 \rangle : \mathrm{T}_1 \times \mathrm{T}_2}$$

Typing rules

$$\Gamma; R \vdash t : \mathrm{T}$$

Types of names/variables/states

Set of randoms

Term

Type

Encryption: $$\frac{\Gamma; R \vdash t : \mathrm{T} \qquad \Gamma(k) = \mathrm{SK}[\mathrm{T}]}{\Gamma; R \sqcup \{r\} \vdash \mathsf{senc}(t, k[\vec{j}], r[\vec{i}]) : \mathrm{Low}}$$

Decryption: $$\frac{\Gamma; R \vdash t : \mathrm{Low} \qquad \Gamma(k) = \mathrm{SK}[\mathrm{T}]}{\Gamma; R \vdash \mathsf{sdec}(t, k[\vec{j}]) : \mathrm{T} + \mathrm{Cte}(\mathrm{fail})}$$

Contributions

1. Design of the type system

2. Soundness result

Soundness

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

3. Case studies

4. Asymmetric encryption

Proof sketch

Sdec

Senc Pair

Zeros

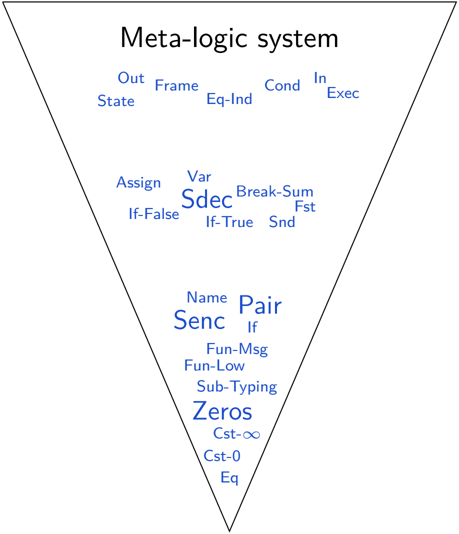
Proof sketch

Out Frame Cond In
State Eq-Ind Exec

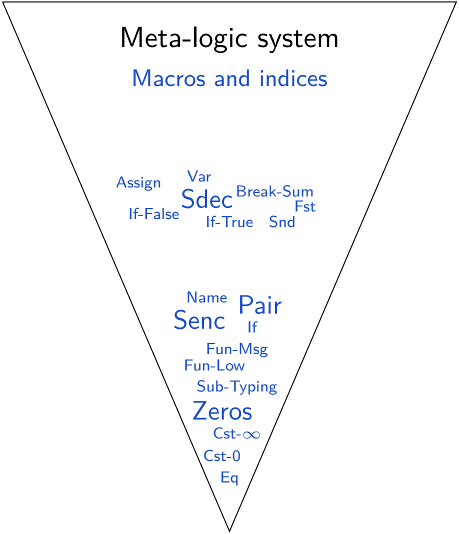
Var
Assign Sdec Break-Sum
If-False Fst
If-True Snd

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

Proof sketch



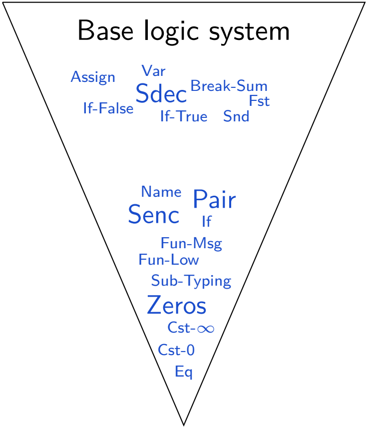
Proof sketch



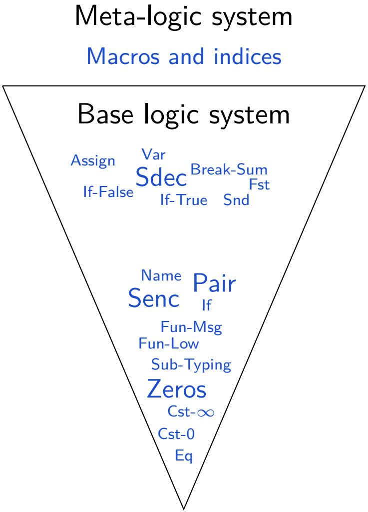
Proof sketch

Meta-logic system

Macros and indices



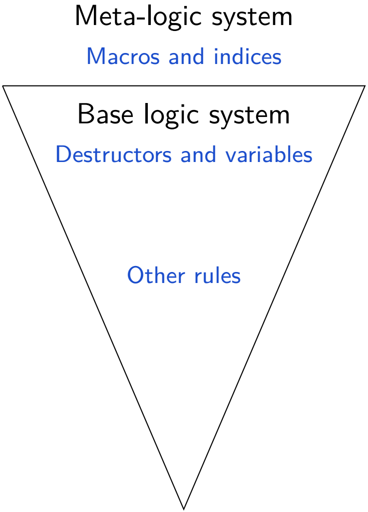
Proof sketch



Problems of the base system:

- Decryption
- Some rules modify the environment
- Some rules do not type all subterms

Proof sketch



Problems of the base system:

- Decryption
- Some rules modify the environment
- Some rules do not type all subterms

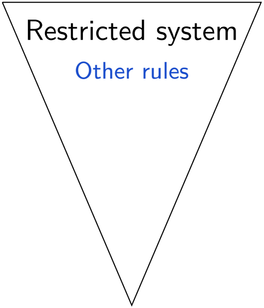
Proof sketch

Meta-logic system

Macros and indices

Base logic system

Destructors and variables



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,

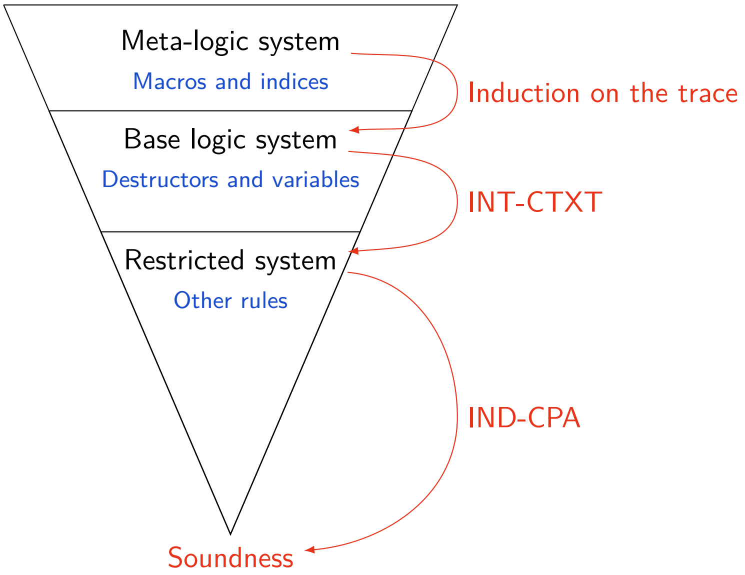
  all subterms type in the same environment,

  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



Use of the theorem

Soundness

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

If a protocol is well typed in $\Gamma; R$

If a term t type High

The attacker cannot find $[\![t]\!]$ with the frame of the protocol

Use of the theorem

Soundness

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

If a protocol is well typed in $\Gamma; R$ →

In each action:
- Output types Low
- Condition types Bool
- States types as indicated in Γ

If a term t type High

The attacker cannot find $[\![t]\!]$ with the frame of the protocol

Contributions

1 Design of the type system

2 Soundness result

3 Case studies

4 Asymmetric encryption

Case studies

	no tag	tags
Wide Mouth Frog	✓	✓
Denning Sacco	✗	✓
Otways-Rees	✗	✓
Needham-Schroeder*	✗	✓
Yahalom*	✗	✓
Yahalom-Paulson*	✗	✓
Mechanism 6$^{\diamond}$	-	✓
Mechanism 9$^{\diamond}$	-	✓
Mechanism 13$^{\diamond}$	-	✓

$^{\diamond}$: ISO/IEC 11770 standard part II

* : Without last message

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.

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

$$\frac{\Gamma(k) = \mathrm{AK}[\mathrm{T}]}{\Gamma; R \vdash \mathsf{pk}(k[\vec{j}]) : \mathrm{Low}}$$

Encryption: Aenc

$$\frac{\Gamma; R \vdash t : \mathrm{T} \qquad \Gamma(k) = \mathrm{AK}[\mathrm{T}]}{\Gamma; R \sqcup \{r\} \vdash \mathsf{aenc}(t, \mathsf{pk}(k[\vec{j}]), r[\vec{i}]) : \mathrm{Low}}$$

Decryption: Adec

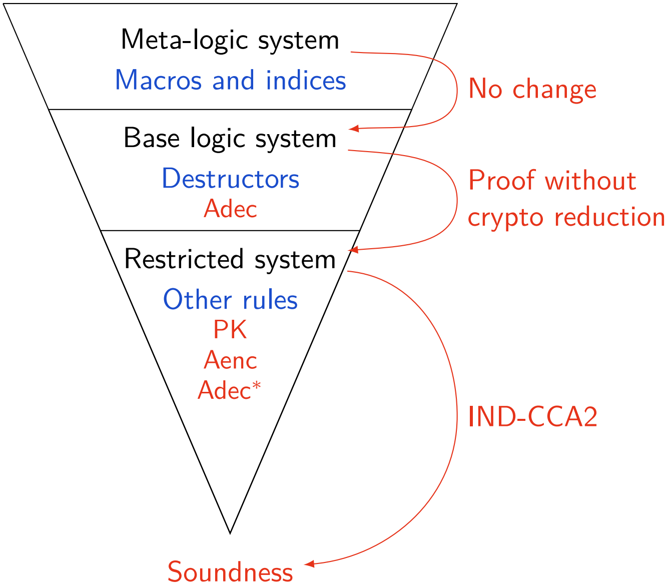
$$\frac{\Gamma; R \vdash t : \mathrm{Low} \qquad \Gamma(k) = \mathrm{AK}[\mathrm{T}]}{\Gamma; R \vdash \mathsf{adec}(t, k[\vec{j}]) : \mathrm{T} + \mathrm{Low}}$$

New rules for IND-CCA2 asymmetric encryption

Public key: PK

Encryption: Aenc

Decryption: Adec



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**