Deciding Indistinguishability: A Decision Result for a Set of Cryptographic Game Transformations

Adrien Koutsos

March 13, 2018

Introduction

Motivation

- Security protocols are distributed programs which aim at providing some security properties.
- They are extensively used, and bugs can be very costly.
- Security protocols are often short, but the security properties are complex.

$\Rightarrow$ Need to use formal methods.

Introduction

Goal of this work

We focus on *fully automatic* proofs of *indistinguishability* properties in the *computational* model:

- **Computational model:** the adversary is any *probabilistic polynomial time Turing machine*. This offers strong security guarantees.
- **Indistinguishability properties:** e.g. strong secrecy, anonymity or unlinkability.
- **Fully automatic:** we want a complete decision procedure.

The Private Authentication Protocol

$$\begin{aligned} &A' &:\quad & n_{A'} \xleftarrow{\$} \\ &B &:\quad & n_B \xleftarrow{\$} \end{aligned}$$

$$\begin{aligned} &1 : A' \longrightarrow B &:\quad & \{\langle \mathsf{pk}(A'), n_{A'} \rangle\}_{\mathsf{pk}(B)} \\ &2 : B \longrightarrow A' &:\quad & \begin{cases} \{\langle n_{A'}, n_B \rangle\}_{\mathsf{pk}(A)} & \text{if } \mathsf{pk}(A') = \mathsf{pk}(A) \\ \{\langle n_B, n_B \rangle\}_{\mathsf{pk}(A)} & \text{otherwise} \end{cases} \end{aligned}$$

Model: Messages

Messages

In the computational model, a message is a *distribution over bitstrings*. We only consider distribution built using:

- Random uniform sampling $\mathsf{n_A}, \mathsf{n_B} \dots$ over $\{0,1\}^\eta$.
- Function applications:
  $\mathsf{A}, \mathsf{B}, \langle \_\,,\,\_ \rangle, \pi_i(\_), \{\_\}_{\_}, \mathsf{pk}(\_), \mathsf{sk}(\_), \text{if } \_ \text{ then } \_ \text{ else } \_ \dots.$

Model: Messages

Messages

In the computational model, a message is a *distribution over bitstrings*. We only consider distribution built using:

- Random uniform sampling $\mathsf{n_A}, \mathsf{n_B} \ldots$ over $\{0,1\}^\eta$.
- Function applications:
  $\mathsf{A}, \mathsf{B}, \langle \_\,,\,\_ \rangle, \pi_i(\_), \{\_\}_{\_}, \mathsf{pk}(\_), \mathsf{sk}(\_), \mathsf{if}\ \_\ \mathsf{then}\ \_\ \mathsf{else}\ \_\ \ldots$.

Examples

$\langle \mathsf{n_A}\,,\,\mathsf{A} \rangle$ $\qquad \pi_1(\mathsf{n_B})$ $\qquad \{\langle \mathsf{pk}(\mathsf{A'})\,,\,\mathsf{n_{A'}} \rangle\}_{\mathsf{pk}(\mathsf{B})}$

Model: Messages

The Private Authentication Protocol

$$
\begin{array}{lcl}
1 : \mathsf{A'} \longrightarrow \mathsf{B} & : & \{\langle \mathsf{pk}(\mathsf{A'}) \, , \, \mathsf{n}_{\mathsf{A'}} \rangle\}_{\mathsf{pk}(\mathsf{B})} \\
2 : \mathsf{B} \longrightarrow \mathsf{A'} & : & \begin{cases} \{\langle \mathsf{n}_{\mathsf{A'}} \, , \, \mathsf{n}_{\mathsf{B}} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{if } \mathsf{pk}(\mathsf{A'}) = \mathsf{pk}(\mathsf{A}) \\ \{\langle \mathsf{n}_{\mathsf{B}} \, , \, \mathsf{n}_{\mathsf{B}} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{otherwise} \end{cases}
\end{array}
$$

How do we represent the adversary's inputs?

Model: Messages

The Private Authentication Protocol

$$
\begin{array}{llll}
1 : \mathsf{A'} \longrightarrow \mathsf{B} & : & \{\langle \mathsf{pk}(\mathsf{A'}) , \mathsf{n}_{\mathsf{A'}} \rangle\}_{\mathsf{pk}(\mathsf{B})} & \\
2 : \mathsf{B} \longrightarrow \mathsf{A'} & : & \begin{cases} \{\langle \mathsf{n}_{\mathsf{A'}} , \mathsf{n}_{\mathsf{B}} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{if } \mathsf{pk}(\mathsf{A'}) = \mathsf{pk}(\mathsf{A}) \\ \{\langle \mathsf{n}_{\mathsf{B}} , \mathsf{n}_{\mathsf{B}} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{otherwise} \end{cases}
\end{array}
$$

How do we represent the adversary's inputs?

- We use special functions symbols $\mathbf{g}, \mathbf{g}_0, \mathbf{g}_1 \ldots$.

Model: Messages

The Private Authentication Protocol

$$
\begin{aligned}
1 : \mathsf{A'} \longrightarrow \mathsf{B} \quad &: \quad \{\langle \mathsf{pk}(\mathsf{A'}) \,,\, \mathsf{n}_{\mathsf{A'}} \rangle\}_{\mathsf{pk}(\mathsf{B})} \\
2 : \mathsf{B} \longrightarrow \mathsf{A'} \quad &: \quad \begin{cases} \{\langle \mathsf{n}_{\mathsf{A'}} \,,\, \mathsf{n}_{\mathsf{B}} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{if } \mathsf{pk}(\mathsf{A'}) = \mathsf{pk}(\mathsf{A}) \\ \{\langle \mathsf{n}_{\mathsf{B}} \,,\, \mathsf{n}_{\mathsf{B}} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{otherwise} \end{cases}
\end{aligned}
$$

How do we represent the adversary's inputs?

- We use special functions symbols $\mathbf{g}, \mathbf{g}_0, \mathbf{g}_1 \ldots$.
- Intuitively, they can be any *probabilistic polynomial time algorithm*.
- Moreover, branching of the protocol is done using if _ then _ else _.

Model: Messages

The Private Authentication Protocol

$$1 : A' \longrightarrow B \quad : \quad \{\langle pk(A') , n_{A'} \rangle\}_{pk(B)}$$

$$2 : B \longrightarrow A' \quad : \quad \begin{cases} \{\langle n_{A'} , n_B \rangle\}_{pk(A)} & \text{if } pk(A') = pk(A) \\ \{\langle n_B , n_B \rangle\}_{pk(A)} & \text{otherwise} \end{cases}$$

Model: Messages

The Private Authentication Protocol

$$1 : \mathsf{A'} \longrightarrow \mathsf{B} \quad : \quad \{\langle \mathsf{pk}(\mathsf{A'})\,,\, \mathsf{n_{A'}} \rangle\}_{\mathsf{pk}(\mathsf{B})}$$

$$2 : \mathsf{B} \longrightarrow \mathsf{A'} \quad : \quad \begin{cases} \{\langle \mathsf{n_{A'}}\,,\, \mathsf{n_B} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{if } \mathsf{pk}(\mathsf{A'}) = \mathsf{pk}(\mathsf{A}) \\ \{\langle \mathsf{n_B}\,,\, \mathsf{n_B} \rangle\}_{\mathsf{pk}(\mathsf{A})} & \text{otherwise} \end{cases}$$

Term Representing the Messages in PA

$$\begin{aligned} t_1 &= \{\langle \mathsf{pk}(\mathsf{A'})\,,\, \mathsf{n_{A'}} \rangle\}_{\mathsf{pk}(\mathsf{B})} \\ t_2 &= \text{if} \quad \mathsf{EQ}(\pi_1(\mathsf{dec}(\mathbf{g}(t_1), \mathsf{sk}(\mathsf{B}))); \mathsf{pk}(\mathsf{A})) \\ &\quad\ \text{then } \{\langle \pi_2(\mathsf{dec}(\mathbf{g}(t_1), \mathsf{sk}(\mathsf{B})))\,,\, \mathsf{n_B} \rangle\}_{\mathsf{pk}(\mathsf{A})} \\ &\quad\ \text{else} \qquad \{\langle \mathsf{n_B}\,,\, \mathsf{n_B} \rangle\}_{\mathsf{pk}(\mathsf{A})} \end{aligned}$$

Model: Protocol Execution

Protocol Execution

The execution of a protocol P is a sequence of terms using adversarial function symbols:

$$u_0^P, \ldots, u_n^P$$

where u_i^P is the i-th message sent on the network by P.

Model: Protocol Execution

Protocol Execution

The execution of a protocol P is a sequence of terms using adversarial function symbols:

$$u_0^P, \ldots, u_n^P$$

where u_i^P is the i-th message sent on the network by P.

Remark

- Only possible for a bounded number of sessions.
- The sequence of terms can be automatically computed (*folding*).

Model: Security Property

Indistinguishability Properties

Two protocols P and Q are *indistinguishable* if every adversary $\mathcal{A}$ loses the following game:

- We toss a coin b.
- If $b = 0$, then $\mathcal{A}$ interacts with P. Otherwise $\mathcal{A}$ interacts with Q.
  Remark: $\mathcal{A}$ is an active adversary (it is the network).
- After the protocol execution, $\mathcal{A}$ outputs a guess b' for b.

$\mathcal{A}$ wins if it guesses correctly with probability better than $\approx \mathbf{1/2}$.

Model: Security Properties

Proposition

$$P \text{ and } Q \text{ are indistinguishable}$$

$$\Leftrightarrow$$

$$u_0^P, \ldots, u_n^P \text{ and } u_0^Q, \ldots, u_n^Q \text{ are indistinguishable}$$

$$\Leftrightarrow$$

$$u_0^P, \ldots, u_n^P \quad \sim \quad u_0^Q, \ldots, u_n^Q$$

Model: Security Properties

Proposition

$$P \text{ and } Q \text{ are indistinguishable}$$

$$\Leftrightarrow$$

$$u_0^P, \dots, u_n^P \text{ and } u_0^Q, \dots, u_n^Q \text{ are indistinguishable}$$

$$\Leftrightarrow$$

$$u_0^P, \dots, u_n^P \quad \sim \quad u_0^Q, \dots, u_n^Q$$

Example: Privacy for PA

$$t_1^{\mathsf{A}}, t_2^{\mathsf{A}} \quad \sim \quad t_1^{\mathsf{A'}}, t_2^{\mathsf{A'}}$$

Model: Summary

Summary

- **Messages** are represented by *terms*, which are built using names $\mathcal{N}$ and function symbols $\mathcal{F}$.
- **A protocol execution** is represented by a sequence of terms.
- **Indistinguishability properties** are expressed through games:

$$u_0^P, \ldots, u_n^P \quad \sim \quad u_0^Q, \ldots, u_n^Q$$

Basic Games

Basic Games

We know that some indistinguishability games are secure:

- Using α-renaming of random samplings:

$$\mathsf{n}_A, \mathsf{n}_B \sim \mathsf{n}_C, \mathsf{n}_D$$

Basic Games

Basic Games

We know that some indistinguishability games are secure:

- Using α-renaming of random samplings:

$$\mathsf{n_A}, \mathsf{n_B} \sim \mathsf{n_C}, \mathsf{n_D}$$

- Using probabilistic arguments:

$$\text{when } \mathsf{n_A} \not\in \mathsf{st}(t), \quad \begin{cases} t \oplus \mathsf{n_A} \sim \mathsf{n_B} \\ \mathsf{EQ}(t; \mathsf{n_A}) \sim \mathsf{false} \end{cases}$$

Basic Games

Basic Games

We know that some indistinguishability games are secure:

- Using α-renaming of random samplings:

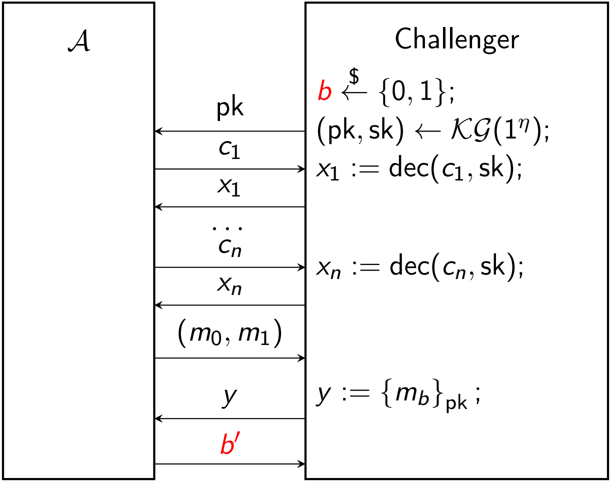
$$\mathsf{n_A}, \mathsf{n_B} \sim \mathsf{n_C}, \mathsf{n_D}$$

- Using probabilistic arguments:

$$\text{when } \mathsf{n_A} \notin \mathsf{st}(t), \quad \begin{cases} t \oplus \mathsf{n_A} \sim \mathsf{n_B} \\ \mathsf{EQ}(t; \mathsf{n_A}) \sim \mathsf{false} \end{cases}$$

- Using *cryptographic assumptions* on the security primitives, e.g. if $\{\_\}_{\_}$, dec(_, _), pk(_), sk(_) is IND-CCA1.

Cryptographic assumptions: IND-CCA1



$\boxed{b = b'?}$

Basic Game: Cryptographic Assumptions

Enc_{CCA1} Games:

$$\vec{v}, \{m_0\}_{\mathrm{pk}} \sim \vec{v}, \{m_1\}_{\mathrm{pk}}$$

Basic Game: Cryptographic Assumptions

Enc_{CCA1} Games:

$$\vec{v}, \{m_0\}_{\mathsf{pk}} \sim \vec{v}, \{m_1\}_{\mathsf{pk}}$$

Assuming:

- sk occurs only in decryption position in $\vec{v}, m_0, m_1$.

Theorem

The Enc_{CCA1} games are secure when the encryption and decryption function are an IND-CCA1 encryption scheme.

Basic Game: Cryptographic Assumptions

Enc_{CCA1} Games:

$$\vec{v}, \{m_0\}_{\mathsf{pk}} \sim \vec{v}, \{m_1\}_{\mathsf{pk}}$$

Assuming:

- sk occurs only in decryption position in $\vec{v}, m_0, m_1$.

Theorem

The Enc_{CCA1} games are secure when the encryption and decryption function are an IND-CCA1 encryption scheme.

Other cryptographic assumptions

IND-CPA, IND-CCA2, CR, PRF, EUF-CMA ...

Game Transformations

Proof Technique

- If $\vec{u} \sim \vec{v}$ is not a basic game, we try to show that it is secure through a succession of *game transformations*:

$$\frac{\vec{s} \sim \vec{t}}{\vec{u} \sim \vec{v}}$$

- This is the way cryptographers or CryptoVerif do proofs.

Game Transformations

Proof Technique

- If $\vec{u} \sim \vec{v}$ is not a basic game, we try to show that it is secure through a succession of *game transformations*:

$$\frac{\vec{s} \sim \vec{t}}{\vec{u} \sim \vec{v}}$$

- This is the way cryptographers or CryptoVerif do proofs.
- **Validity by reduction:** $\vec{u} \sim \vec{v}$ can be replaced by $\vec{s} \sim \vec{t}$ when, given an adversary winning $\vec{u} \sim \vec{v}$, we can build an adversary winning $\vec{s} \sim \vec{t}$.

Game Transformations

Proof Technique

- If $\vec{u} \sim \vec{v}$ is not a basic game, we try to show that it is secure through a succession of *game transformations*:

$$\frac{\vec{s} \sim \vec{t}}{\vec{u} \sim \vec{v}}$$

- This is the way cryptographers or CryptoVerif do proofs.
- **Validity by reduction:** $\vec{u} \sim \vec{v}$ can be replaced by $\vec{s} \sim \vec{t}$ when, given an adversary winning $\vec{u} \sim \vec{v}$, we can build an adversary winning $\vec{s} \sim \vec{t}$.

Example

$$\frac{x \sim y}{y \sim x}\ Sym$$

Structural Game Transformation

Duplicate

$$\frac{x \sim y}{x, x \sim y, y}\ \text{Dup}$$

Structural Game Transformation

Duplicate

$$\frac{\vec{w}_l, x \sim \vec{w}_r, y}{\vec{w}_l, x, x \sim \vec{w}_r, y, y}\ \text{Dup}$$

Structural Game Transformation

Function Application

If you cannot distinguish the arguments, you cannot distinguish the images.

$$\frac{x_1, \ldots, x_n \sim y_1, \ldots, y_n}{f(x_1, \ldots, x_n) \sim f(y_1, \ldots, y_n)}\ \text{FA}$$

Structural Game Transformation

Function Application

If you cannot distinguish the arguments, you cannot distinguish the images.

$$\frac{\vec{w}_l, x_1, \ldots, x_n \sim \vec{w}_r, y_1, \ldots, y_n}{\vec{w}_l, f(x_1, \ldots, x_n) \sim \vec{w}_r, f(y_1, \ldots, y_n)}\ \text{FA}$$

Structural Game Transformation

Case Study

If we use Function Application on (if then else):

$$\frac{b, u, v \sim b', u', v'}{\text{if } b \text{ then } u \text{ else } v \sim \text{if } b' \text{ then } u' \text{ else } v'} \text{ FA}$$

Structural Game Transformation

Case Study

If we use Function Application on (if then else):

$$\frac{b, u, v \sim b', u', v'}{\text{if } b \text{ then } u \text{ else } v \sim \text{if } b' \text{ then } u' \text{ else } v'} \text{ FA}$$

But we can do better:

$$\frac{b, u \sim \quad b', u' \qquad b, v \sim \quad b', v'}{\text{if } b \text{ then } u \text{ else } v \sim \quad \text{if } b' \text{ then } u' \text{ else } v'} \text{ CS}$$

Structural Game Transformation

Case Study

If we use Function Application on (if then else):

$$\frac{b, u, v \sim b', u', v'}{\text{if } b \text{ then } u \text{ else } v \sim \text{if } b' \text{ then } u' \text{ else } v'} \text{ FA}$$

But we can do better:

$$\frac{\vec{w}_l, b, u \sim \vec{w}_r, b', u' \qquad \vec{w}_l, b, v \sim \vec{w}_r, b', v'}{\vec{w}_l, \text{if } b \text{ then } u \text{ else } v \sim \vec{w}_r, \text{if } b' \text{ then } u' \text{ else } v'} \text{ CS}$$

Game Transformation: Term Rewriting System

Remark: $\sim$ is not a congruence!

Counter-Example: $\mathsf{n} \sim \mathsf{n}$ and $\mathsf{n} \sim \mathsf{n}'$, but $\mathsf{n}, \mathsf{n} \not\sim \mathsf{n}, \mathsf{n}'$.

Game Transformation: Term Rewriting System

Remark: $\sim$ is not a congruence!

Counter-Example: $\mathsf{n} \sim \mathsf{n}$ and $\mathsf{n} \sim \mathsf{n}'$, but $\mathsf{n}, \mathsf{n} \not\sim \mathsf{n}, \mathsf{n}'$.

Congruence

If $\mathsf{EQ}(u; v) \sim \mathsf{true}$ then u and v are (almost always) *equal*
$\Rightarrow$ we have a congruence.

$u = v$ syntactic sugar for $\mathsf{EQ}(u; v) \sim \mathsf{true}$

Equational Theory: Protocol Functions

- $\pi_i\left(\langle x_1, x_2 \rangle\right) = x_i$ $\qquad i \in \{1, 2\}$
- $\mathsf{dec}(\{x\}_{\mathsf{pk}(\mathsf{y})}, \mathsf{sk}(\mathsf{y})) = x$

Game Transformation: Term Rewriting System

Equational Theory: Protocol Functions

If Homomorphism:

$f(\vec{u}, \text{if } b \text{ then } x \text{ else } y, \vec{v}) = \text{if } b \text{ then } f(\vec{u}, x, \vec{v}) \text{ else } f(\vec{u}, y, \vec{v})$

$\text{if } (\text{if } b \text{ then } a \text{ else } c) \text{ then } x \text{ else } y =$

$\qquad \text{if } b \text{ then } (\text{if } a \text{ then } x \text{ else } y) \text{ else } (\text{if } c \text{ then } x \text{ else } y)$

Game Transformation: Term Rewriting System

Equational Theory: Protocol Functions

If Homomorphism:

$f(\vec{u}, \text{if } b \text{ then } x \text{ else } y, \vec{v}) = \text{if } b \text{ then } f(\vec{u}, x, \vec{v}) \text{ else } f(\vec{u}, y, \vec{v})$

$\text{if } (\text{if } b \text{ then } a \text{ else } c) \text{ then } x \text{ else } y =$

$\qquad \text{if } b \text{ then } (\text{if } a \text{ then } x \text{ else } y) \text{ else } (\text{if } c \text{ then } x \text{ else } y)$

If Rewriting:

$\text{if } b \text{ then } x \text{ else } x = x$

$\text{if } b \text{ then } (\text{if } b \text{ then } x \text{ else } y) \text{ else } z = \text{if } b \text{ then } x \text{ else } z$

$\text{if } b \text{ then } x \text{ else } (\text{if } b \text{ then } y \text{ else } z) = \text{if } b \text{ then } x \text{ else } z$

Game Transformation: Term Rewriting System

Equational Theory: Protocol Functions

If Homomorphism:

$f(\vec{u}, \text{if } b \text{ then } x \text{ else } y, \vec{v}) = \text{if } b \text{ then } f(\vec{u}, x, \vec{v}) \text{ else } f(\vec{u}, y, \vec{v})$

$\text{if } (\text{if } b \text{ then } a \text{ else } c) \text{ then } x \text{ else } y =$
$\quad\quad \text{if } b \text{ then } (\text{if } a \text{ then } x \text{ else } y) \text{ else } (\text{if } c \text{ then } x \text{ else } y)$

If Rewriting:

$\text{if } b \text{ then } x \text{ else } x = x$

$\text{if } b \text{ then } (\text{if } b \text{ then } x \text{ else } y) \text{ else } z = \text{if } b \text{ then } x \text{ else } z$

$\text{if } b \text{ then } x \text{ else } (\text{if } b \text{ then } y \text{ else } z) = \text{if } b \text{ then } x \text{ else } z$

If Re-Ordering:

$\text{if } b \text{ then } (\text{if } a \text{ then } x \text{ else } y) \text{ else } z =$
$\quad\quad \text{if } a \text{ then } (\text{if } b \text{ then } x \text{ else } z) \text{ else } (\text{if } b \text{ then } y \text{ else } z)$

$\text{if } b \text{ then } x \text{ else } (\text{if } a \text{ then } y \text{ else } z) =$
$\quad\quad \text{if } a \text{ then } (\text{if } b \text{ then } x \text{ else } y) \text{ else } (\text{if } b \text{ then } x \text{ else } z)$

Decidability

Decision Problem: Game Transformations

Input: A game $\vec{u} \sim \vec{v}$.

Question: Is there a sequence of game transformations in Ax showing that $\vec{u} \sim \vec{v}$ is secure?

Decidability

Decision Problem: Game Transformations

Input: A game $\vec{u} \sim \vec{v}$.
Question: Is there a sequence of game transformations in Ax showing that $\vec{u} \sim \vec{v}$ is secure?

or equivalently

Decision Problem: Satisfiability

Input: A ground formula $\vec{u} \sim \vec{v}$ in the BC indistinguishability logic.
Question: Is $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ satisfiable?

Game Transformations: Summary

The Non-Basic Game Transformations in Ax

$$\frac{x \sim y}{x, x \sim y, y}\ \text{Dup}$$

$$\frac{x_1, \ldots, x_n \sim y_1, \ldots, y_n}{f(x_1, \ldots, x_n) \sim f(y_1, \ldots, y_n)}\ \text{FA}$$

$$\frac{b, u \sim b', u' \qquad b, v \sim b', v'}{\text{if } b \text{ then } u \text{ else } v \sim \text{if } b' \text{ then } u' \text{ else } v'}\ \text{CS}$$

Game Transformations: Summary

The Non-Basic Game Transformations in Ax

$$\frac{x \sim y}{x, x \sim y, y}\ \text{Dup}$$

$$\frac{x_1, \ldots, x_n \sim y_1, \ldots, y_n}{f(x_1, \ldots, x_n) \sim f(y_1, \ldots, y_n)}\ \text{FA}$$

$$\frac{b, u \sim b', u' \qquad b, v \sim b', v'}{\text{if } b \text{ then } u \text{ else } v \sim \text{if } b' \text{ then } u' \text{ else } v'}\ \text{CS}$$

$$\frac{\vec{u}' \sim \vec{v}'}{\vec{u} \sim \vec{v}}\ R$$

when $\vec{u} =_R \vec{u}'$ and $\vec{v} =_R \vec{v}'$

Term Rewriting System

Theorem

There exists a term rewriting system $\rightarrow_R \subseteq =$ such that:

- $\rightarrow_R$ is convergent.
- $=$ is equal to $({}_R\!\leftarrow \cup \rightarrow_R)^*$.

Strategy

Deconstructing Rules

Rules CS, FA and Dup are decreasing transformations.

Strategy

Deconstructing Rules

Rules CS, FA and Dup are decreasing transformations.

Problems

- The rule R is not decreasing!
- The basic games (CCA1) are given through a recursive schema.

Strategy

Deconstructing Rules

Rules CS, FA and Dup are decreasing transformations.

Problems

- The rule R is not decreasing!
- The basic games (CCA1) are given through a recursive schema.

Naive Idea

R is convergent, so could we restrict proofs to terms in R-normal form?

Difficulties

If Introduction: $x \to \text{if } b \text{ then } x \text{ else } x$

$$\frac{}{\mathsf{n} \sim \text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}$$

Difficulties

If Introduction: $x \rightarrow \text{if } b \text{ then } x \text{ else } x$

$$\dfrac{\dfrac{}{\text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n} \sim \text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n'}}}{\mathsf{n} \sim \text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n'}}\ R$$

Difficulties

If Introduction: $x \to \text{if } b \text{ then } x \text{ else } x$

$$\dfrac{\dfrac{\dfrac{}{\mathsf{n} \sim \mathsf{n}}}{g(), \mathsf{n} \sim g(), \mathsf{n}}\ \mathsf{FA} \qquad \dfrac{\dfrac{}{\mathsf{n} \sim \mathsf{n}'}}{g(), \mathsf{n} \sim g(), \mathsf{n}'}\ \mathsf{FA}}{\dfrac{\text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n} \sim \text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}{\mathsf{n} \sim \text{if } g() \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}\ R}\ \mathsf{CS}$$

Difficulties

If Introduction: : $x \rightarrow$ if b then x else x

$$\frac{}{\vec{u}, \mathsf{n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}$$

Difficulties

If Introduction: : $x \to$ if b then x else x

$$\cfrac{\overline{\phantom{\vec{u}, \text{if } g(\vec{u}) \text{ then n else n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then n else n'}}}}{\cfrac{\vec{u}, \text{if } g(\vec{u}) \text{ then n else n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then n else n'}}{\vec{u}, \text{n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then n else n'}}\ R}$$

Difficulties

If Introduction: : $x \rightarrow \text{if } b \text{ then } x \text{ else } x$

$$\dfrac{\dfrac{\dfrac{}{\vec{u}, \mathsf{n} \sim \vec{u}, \mathsf{n}}}{\vec{u}, g(\vec{u}), \mathsf{n} \sim \vec{u}, g(\vec{u}), \mathsf{n}}\ \text{FA, Dup} \qquad \dfrac{\dfrac{}{\vec{u}, \mathsf{n} \sim \vec{u}, \mathsf{n}'}}{\vec{u}, g(\vec{u}), \mathsf{n} \sim \vec{u}, g(\vec{u}), \mathsf{n}'}\ \text{FA, Dup}}{\dfrac{\vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}{\vec{u}, \mathsf{n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}\ R}\ \text{CS}$$

Difficulties

If Introduction: : $x \rightarrow$ if b then x else x

$$
\dfrac{\dfrac{\dfrac{}{\vec{u}, \mathsf{n} \sim \vec{u}, \mathsf{n}}}{\vec{u}, g(\vec{u}), \mathsf{n} \sim \vec{u}, g(\vec{u}), \mathsf{n}}\ \text{FA, Dup} \qquad \dfrac{\dfrac{}{\vec{u}, \mathsf{n} \sim \vec{u}, \mathsf{n}'}}{\vec{u}, g(\vec{u}), \mathsf{n} \sim \vec{u}, g(\vec{u}), \mathsf{n}'}\ \text{FA, Dup}}{\dfrac{\vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}{\vec{u}, \mathsf{n} \sim \vec{u}, \text{if } g(\vec{u}) \text{ then } \mathsf{n} \text{ else } \mathsf{n}'}\ R}\ \text{CS}
$$

Bounded Introduction

Still, the introduced conditional $g(\vec{u})$ is bounded by the other side.

Decision Procedure

Proof Cut: Introduction of a Conditional on Both Sides

$$\dfrac{\dfrac{a, s \sim b, t \qquad\qquad a, s \sim b, t}{\text{if } a \text{ then } s \text{ else } s \sim \text{if } b \text{ then } t \text{ else } t}\ \text{CS}}{s \sim t}\ R$$

Decision Procedure

Proof Cut: Introduction of a Conditional on Both Sides

$$\dfrac{\dfrac{a, s \sim b, t \qquad\qquad a, s \sim b, t}{\text{if } a \text{ then } s \text{ else } s \sim \text{if } b \text{ then } t \text{ else } t}\ \text{CS}}{s \sim t}\ R$$

Lemma

From a proof of $a, s \sim b, t$ we can extract a smaller proof of $s \sim t$.

Decision Procedure

Proof Cut: Introduction of a Conditional on Both Sides

$$\cfrac{\cfrac{a, s \sim b, t \qquad\qquad a, s \sim b, t}{\text{if } a \text{ then } s \text{ else } s \sim \text{if } b \text{ then } t \text{ else } t}\ \text{CS}}{s \sim t}\ R$$

Lemma

From a proof of $a, s \sim b, t$ we can extract a smaller proof of $s \sim t$.

$\Rightarrow$ **Proof Cut Elimination**

Decision Procedure

Proof Cut

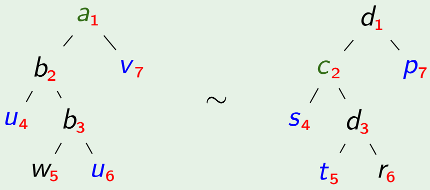
$$
\dfrac{
\dfrac{a_1, b_2, b_3, u_4, w_5, u_6, v_7 \sim d_1, c_2, d_3, s_4, t_5, r_6, p_7}{
\begin{tikzcd}[column sep=small, row sep=small]
 & & a_1 \arrow[dl, dash] \arrow[dr, dash] & \\
 & b_2 \arrow[dl, dash] \arrow[dr, dash] & & v_7 \\
u_4 & & b_3 \arrow[dl, dash] \arrow[dr, dash] & \\
 & w_5 & & u_6
\end{tikzcd}
\sim
\begin{tikzcd}[column sep=small, row sep=small]
 & & d_1 \arrow[dl, dash] \arrow[dr, dash] & \\
 & c_2 \arrow[dl, dash] \arrow[dr, dash] & & p_7 \\
s_4 & & d_3 \arrow[dl, dash] \arrow[dr, dash] & \\
 & t_5 & & r_6
\end{tikzcd}
}\ \mathrm{FA}^{(3)}
}{
\text{if } a \text{ then } u \text{ else } v \sim \text{if } c \text{ then } s \text{ else } t
}\ R
$$

where $p \equiv \text{if } c \text{ then } s \text{ else } t$

Decision Procedure

Proof Cut

$$\dfrac{\dfrac{a_1, b_2, b_3, u_4, w_5, u_6, v_7 \sim d_1, c_2, d_3, s_4, t_5, r_6, p_7}{}\ \mathrm{FA}^{(3)}}{}$$



$$\dfrac{}{\text{if } a \text{ then } u \text{ else } v \sim \text{if } c \text{ then } s \text{ else } t}\ R$$

where $p \equiv$ if c then s else t

Key Lemma

If $b, b \sim b', b''$ can be shown using only FA, Dup and CCA1 then $b' \equiv b''$.

Decision Procedure

Proof Cut

$$
\dfrac{\dfrac{a_1, b_2, b_3, u_4, w_5, u_6, v_7 \sim d_1, c_2, d_3, s_4, t_5, r_6, p_7}{\begin{array}{ccccc} & & a_1 & & \\ & / & & \backslash & \\ b_2 & & & & v_7 \\ / \quad \backslash & & & & \\ u_4 \quad b_3 & & & & \\ \quad / \quad \backslash & & & & \\ \quad w_5 \quad u_6 & & & & \end{array} \sim \begin{array}{ccccc} & & d_1 & & \\ & / & & \backslash & \\ c_2 & & & & p_7 \\ / \quad \backslash & & & & \\ s_4 \quad d_3 & & & & \\ \quad / \quad \backslash & & & & \\ \quad t_5 \quad r_6 & & & & \end{array}}\ \mathrm{FA}^{(3)}}{\text{if } a \text{ then } u \text{ else } v \sim \text{if } c \text{ then } s \text{ else } t}\ R
$$

where $p \equiv$ if c then s else t

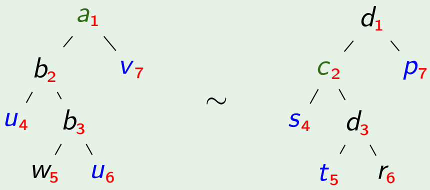
Proof Cut Elimination

- $b_2, b_3 \sim c_2, d_3 \quad \Rightarrow \quad c \equiv d.$

Decision Procedure

Proof Cut

$$\dfrac{a_1, b_2, b_3, u_4, w_5, u_6, v_7 \sim d_1, c_2, d_3, s_4, t_5, r_6, p_7}{}\ \mathrm{FA}^{(3)}$$



$$\dfrac{}{\text{if } a \text{ then } u \text{ else } v \sim \text{if } c \text{ then } s \text{ else } t}\ R$$

where $p \equiv \text{if } c \text{ then } s \text{ else } t$

Proof Cut Elimination

- $b_2, b_3 \sim c_2, d_3 \quad \Rightarrow \quad c \equiv d.$
- $a_1, b_2 \sim d_1, c_2 \quad \Rightarrow \quad a \equiv b.$

Strategy: Theorem

Theorem

The following problem is decidable:
Input: A game $\vec{u} \sim \vec{v}$.
Question: Is there a sequence of game transformations in Ax showing that $\vec{u} \sim \vec{v}$ is secure?

Strategy: Theorem

Theorem

The following problem is decidable:
Input: A game $\vec{u} \sim \vec{v}$.
Question: Is there a sequence of game transformations in Ax showing that $\vec{u} \sim \vec{v}$ is secure?

Remark: Basic Games

The above result holds when using CCA2 as basic games.

Strategy: Theorem

Theorem

The following problem is decidable:
Input: A game $\vec{u} \sim \vec{v}$.
Question: Is there a sequence of game transformations in Ax showing that $\vec{u} \sim \vec{v}$ is secure?

Remark: Basic Games

The above result holds when using CCA2 as basic games.

Sketch

- Commute rule applications to order them as follows:

$$(2\mathsf{Box} + R_\square) \cdot \mathsf{CS}_\square \cdot \mathsf{FA}_{\mathsf{if}} \cdot \mathsf{FA}_{\mathsf{f}} \cdot \mathsf{Dup} \cdot \mathsf{U}$$

- We do proof cut eliminations to get a small proof.

Conclusion

Our Works

- Designed and proved correct a set of game transformations.
- Showed a decision result for this set of game transformations.

Conclusion

Our Works

- Designed and proved correct a set of game transformations.
- Showed a decision result for this set of game transformations.

Advantages and Drawbacks

- Full automation.
- Completeness: absence of proof implies the existence of an attack.
- Bounded number of sessions.
- Cannot easily add cryptographic assumptions: current result only of CCA2.

Conclusion

Our Works

- Designed and proved correct a set of game transformations.
- Showed a decision result for this set of game transformations.

Advantages and Drawbacks

- Full automation.
- Completeness: absence of proof implies the existence of an attack.
- Bounded number of sessions.
- Cannot easily add cryptographic assumptions: current result only of CCA2.

Future Works

- Support for a large class of primitives and associated assumptions.
- Interactive/automatic prover using the strategy.

Thanks for your attention