DECISION PROCEDURE FOR TRACE EQUIVALENCE

V. Cheval, H. Comon-Lundh, S. Delaune
LSV, ENS Cachan, CNRS, INRIA Saclay

15 November 2011

CONTEXT

- Cryptographic protocols

Most communications take place over a public network

Cryptographic protocols

- small programs designed to secure communication (e.g. secrecy)
- use cryptographic primitives (e.g. encryption, signature)

It important to verify their security

CONTEXT

- Reliable cryptography
- Correct specification
- Implementation satisfying the specification

CONTEXT

- Reliable cryptography
- Correct specification
- Implementation satisfying the specification

- Some security properties

CONTEXT

- Reliable cryptography
- Correct specification
- Implementation satisfying the specification

- Some security properties

Reachability properties

- Secrecy, Authentication, ...

CONTEXT

- Reliable cryptography
- Correct specification
- Implementation satisfying the specification

- Some security properties

Reachability properties
- Secrecy, Authentication, ...

Equivalence properties
- Anonymity, Privacy, Receipt-Freeness, ...

CONTEXT

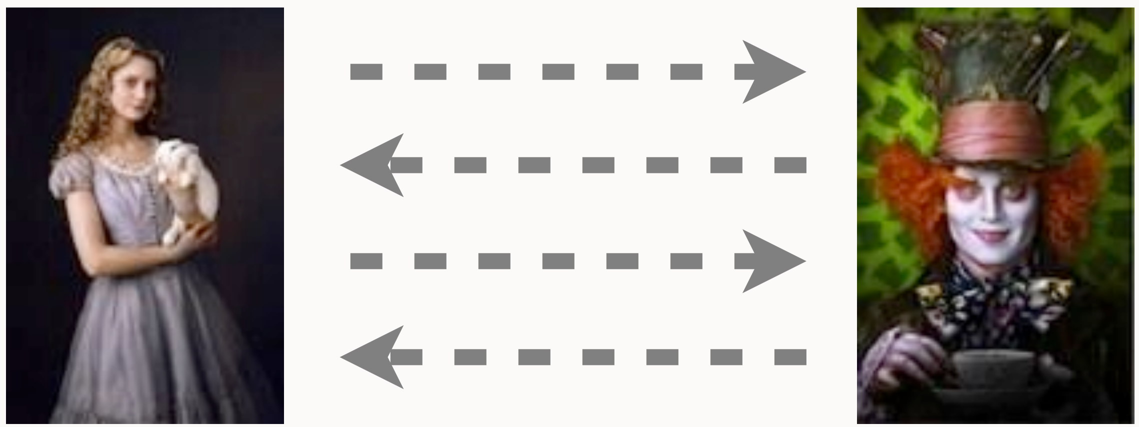
- Formalism

Alice

Bob

CONTEXT

- Formalism



Alice

Bob

CONTEXT

- Formalism

Alice

Intruder

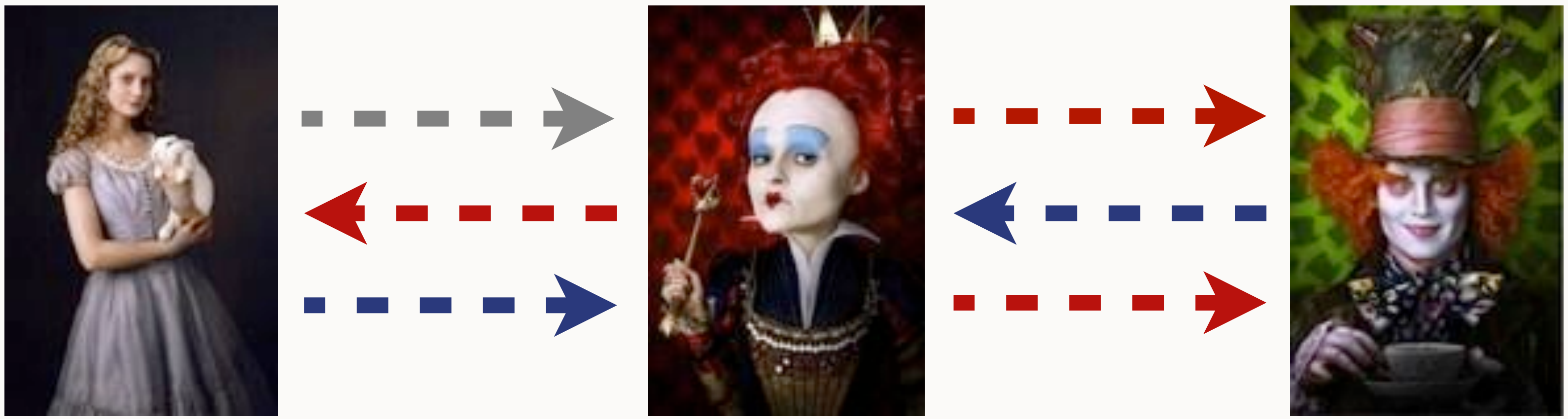
Bob

The intruder can

- intercept all messages
- transmit or modify messages
- test equality between messages
- initiate several sessions

CONTEXT

- Formalism



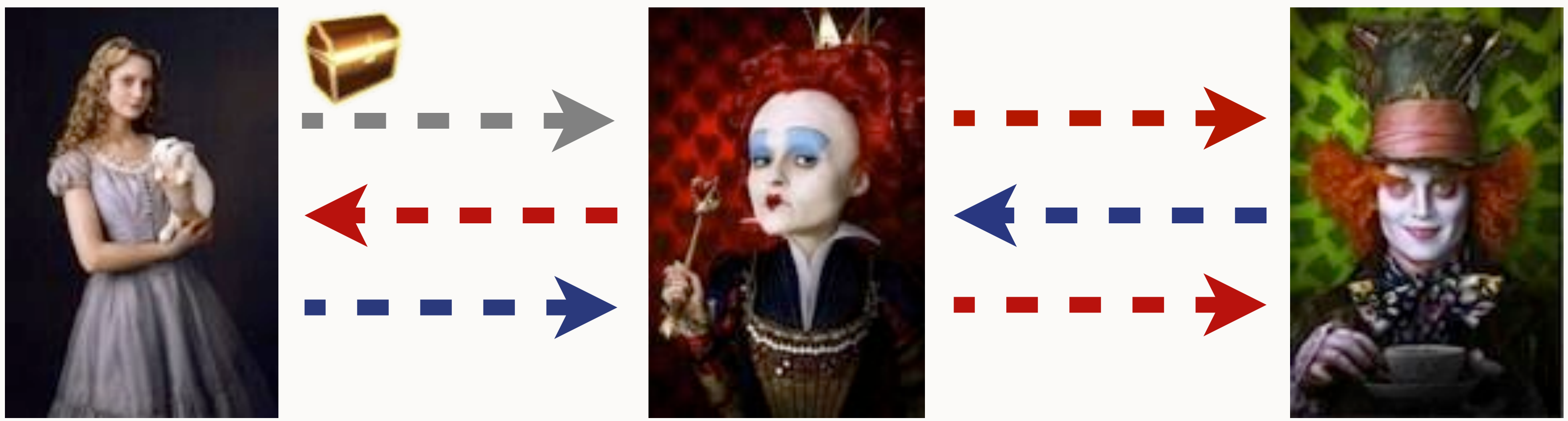
Alice Intruder Bob

The intruder can

- intercept all messages
- transmit or modify messages
- test equality between messages
- initiate several sessions

CONTEXT

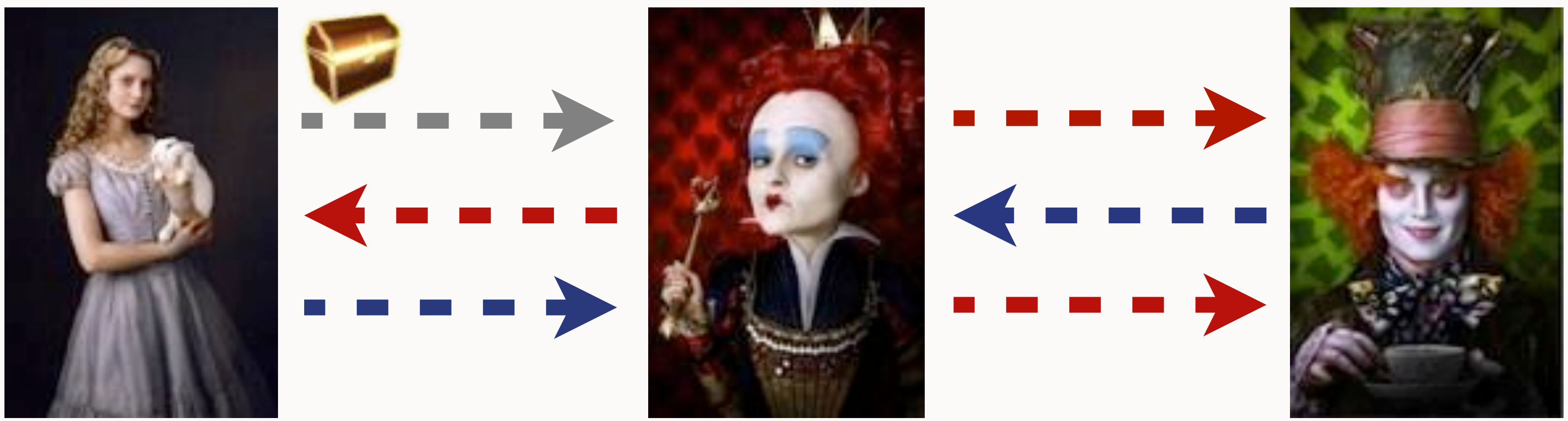
- Reachability properties : secrecy, authentication,...



Alice Intruder Bob

CONTEXT

- Reachability properties : secrecy, authentication,...



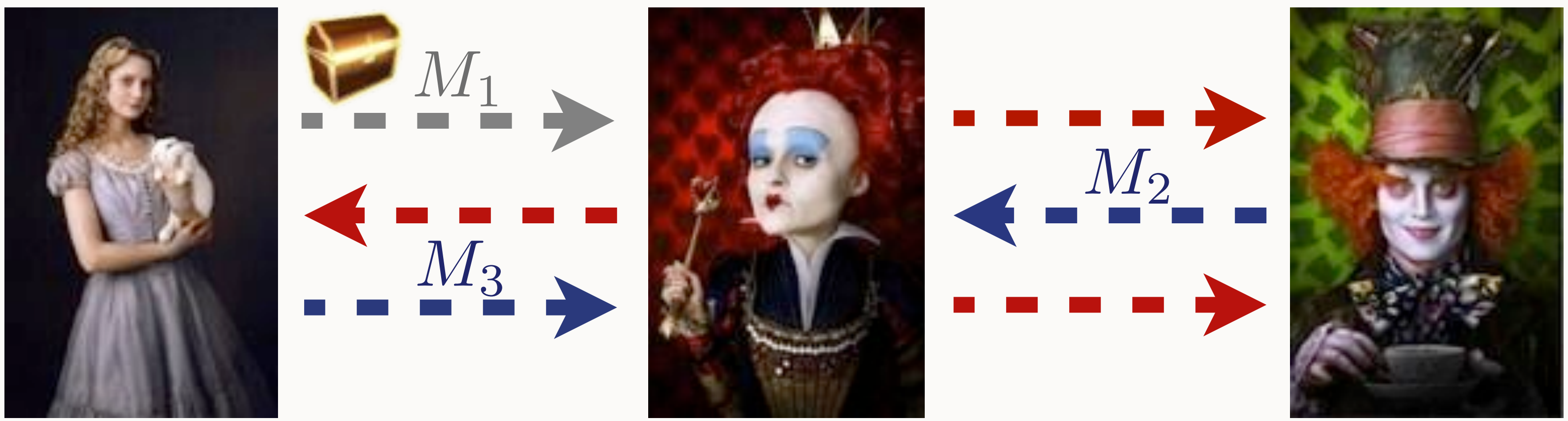
Alice Intruder Bob

Can the intruder deduce Alice's secret ?

CONTEXT

- Reachability properties : secrecy, authentication,...

intruder's knowledge : M_1 M_2 M_3 + basic knowledge

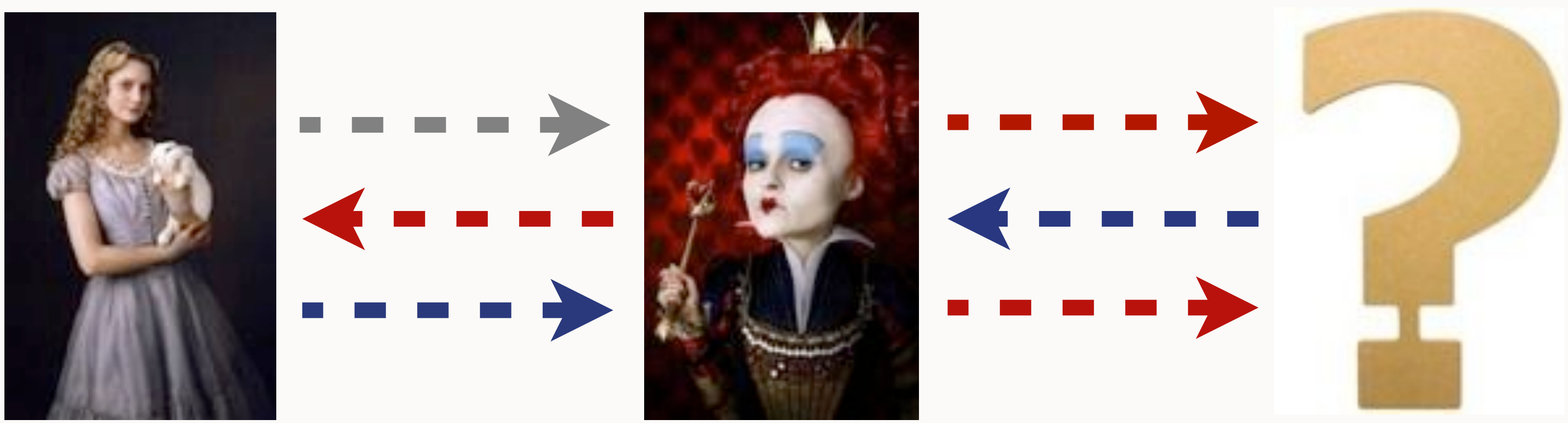


Alice Intruder Bob

Can the intruder deduce Alice's secret ?

CONTEXT

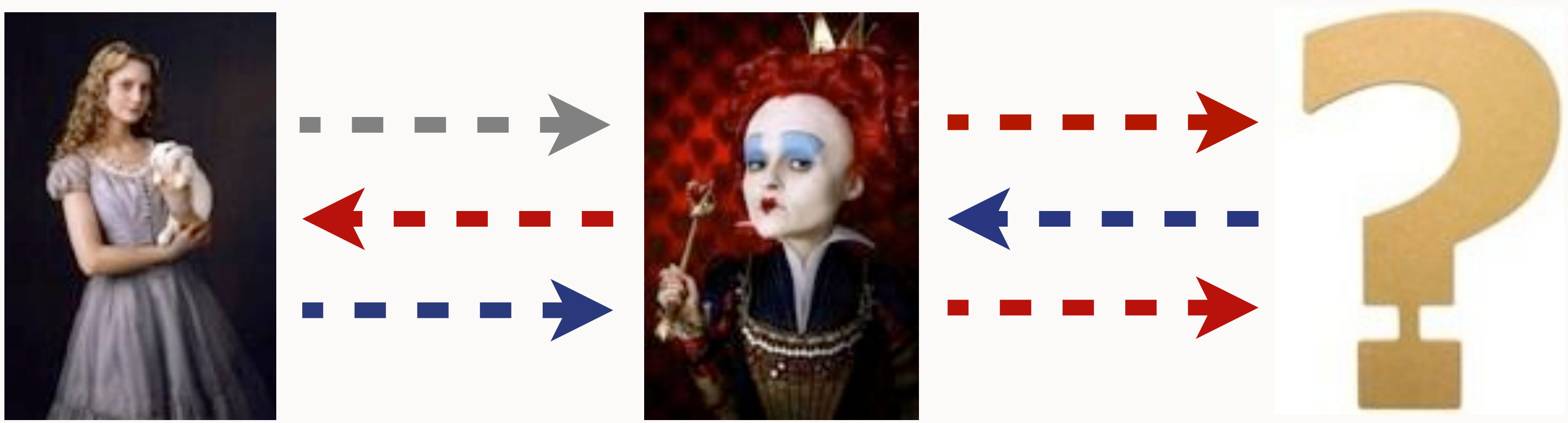
- Equivalence properties : strong secret, anonymity,...



Alice Intruder Unknown

CONTEXT

- Equivalence properties : strong secret, anonymity,...

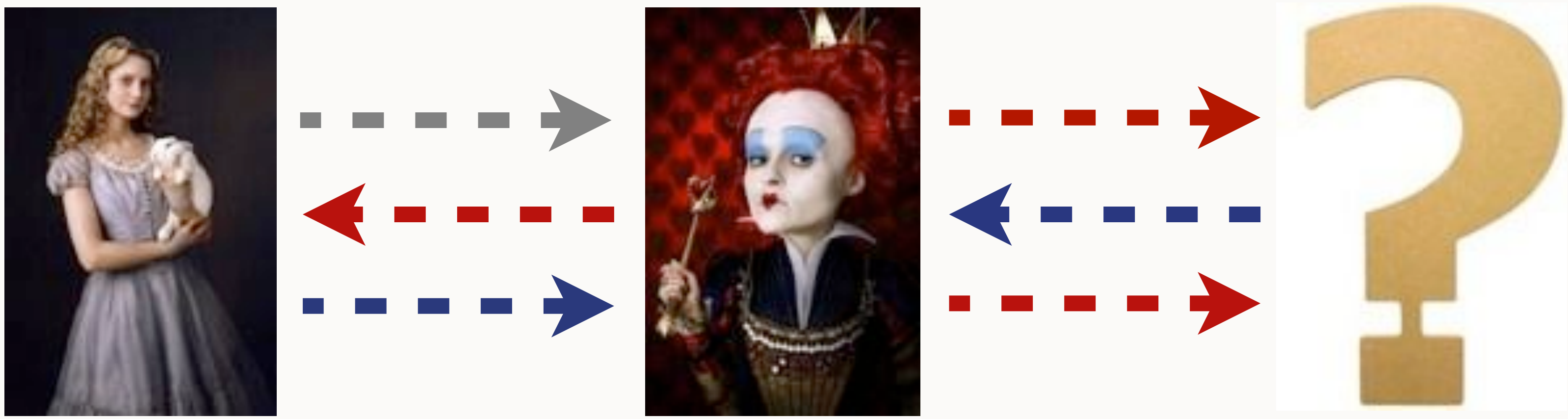


Alice Intruder Unknown

Can the intruder deduce the unknown's identity ?

CONTEXT

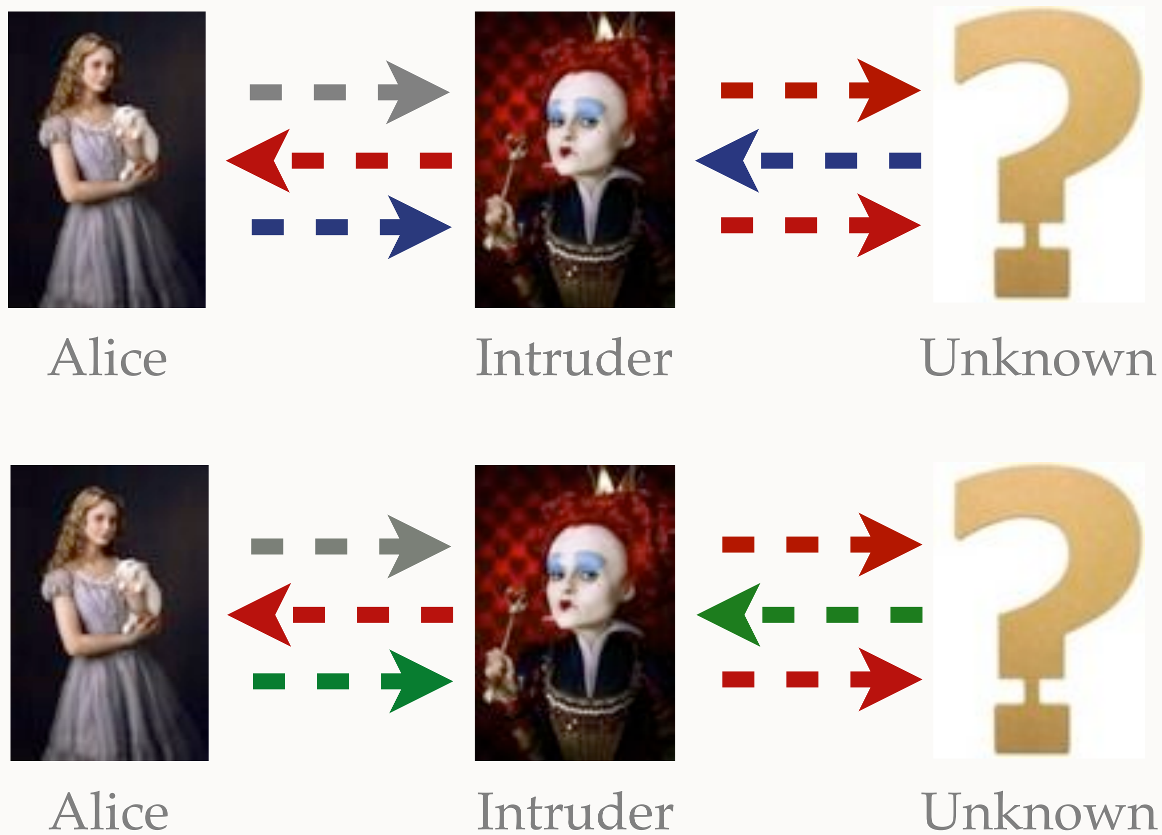
- Equivalence properties : strong secret, anonymity,...



Alice Intruder Unknown

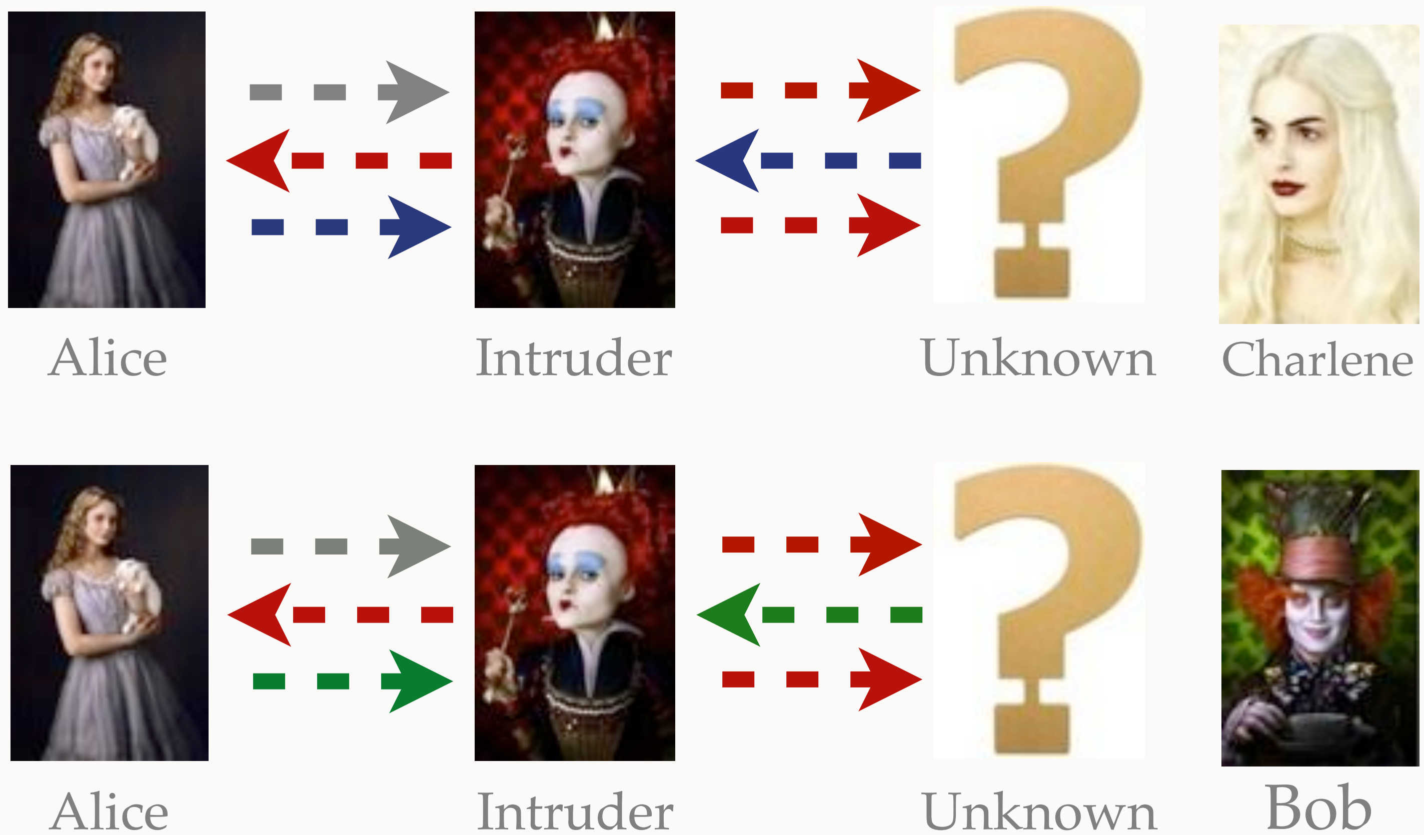
CONTEXT

- Equivalence properties : strong secret, anonymity,...



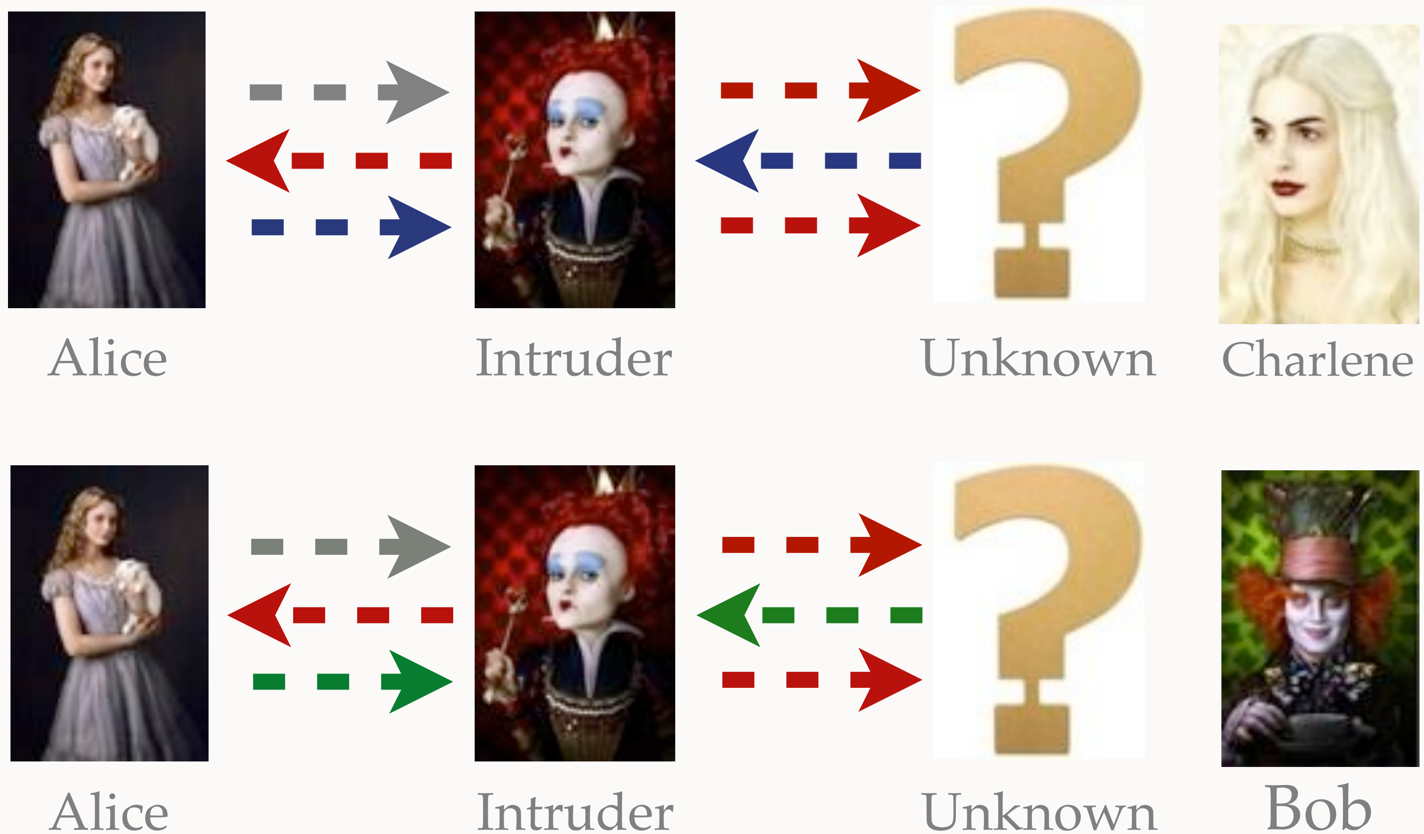
CONTEXT

- Equivalence properties : strong secret, anonymity,...



CONTEXT

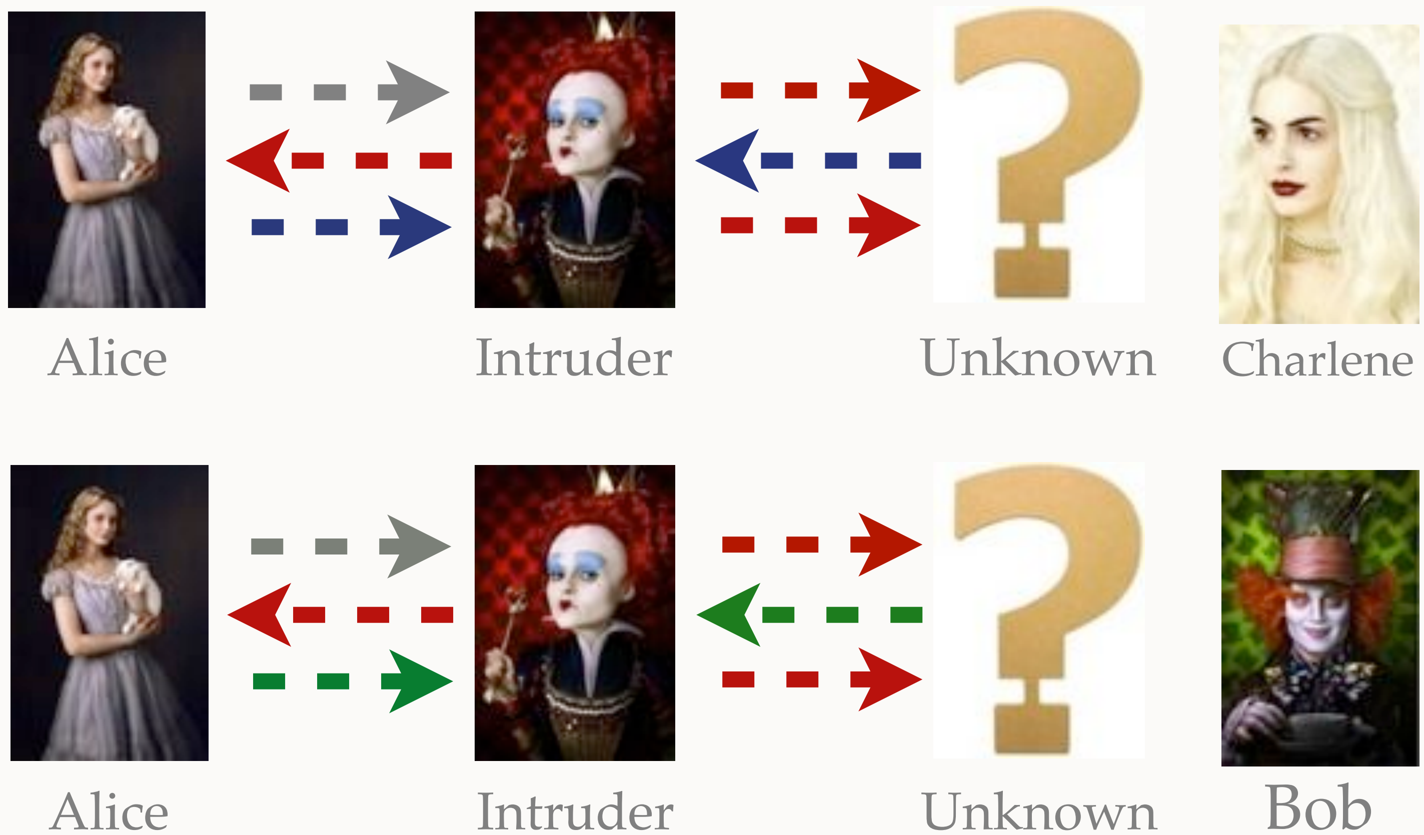
- Equivalence properties : strong secret, anonymity,...



Can the intruder distinguish the two situations ?

CONTEXT

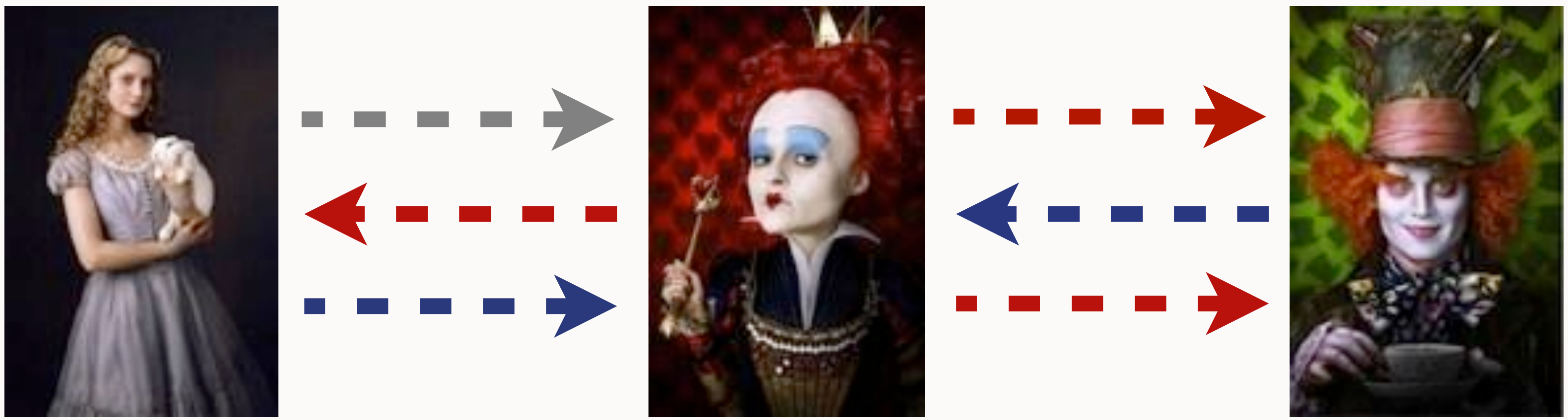
- Equivalence properties : strong secret, anonymity,...



Trace Equivalence

CONTEXT

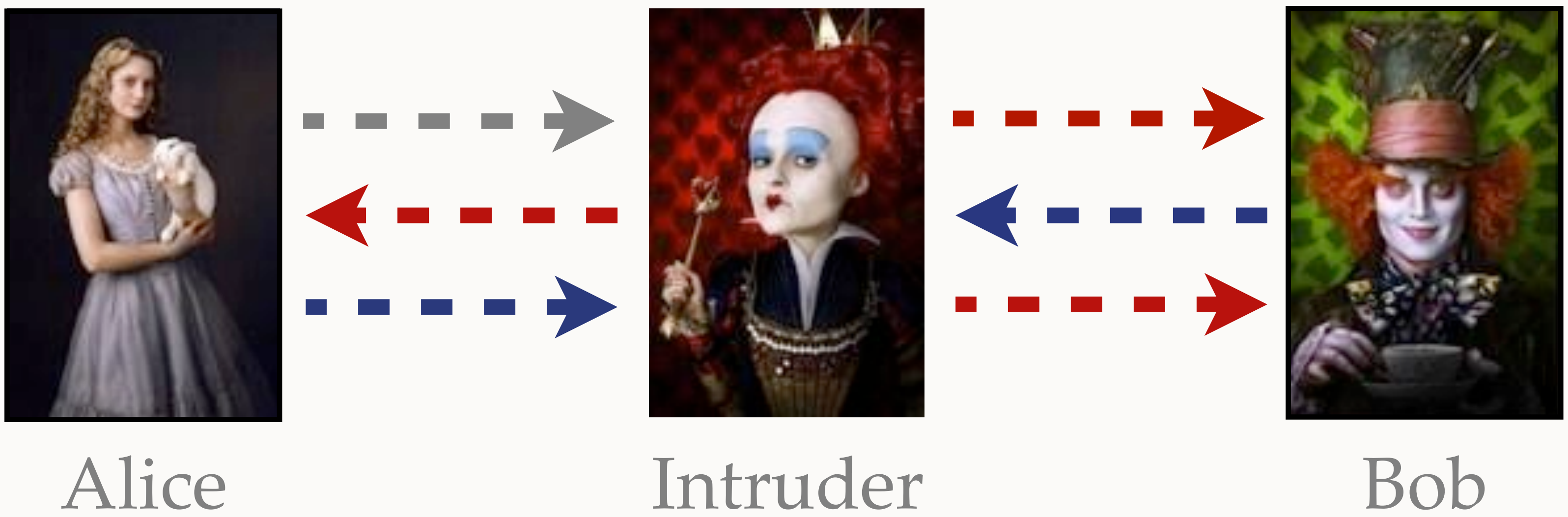
- Knowledge indistinguishability : static equivalence



Alice Intruder Bob

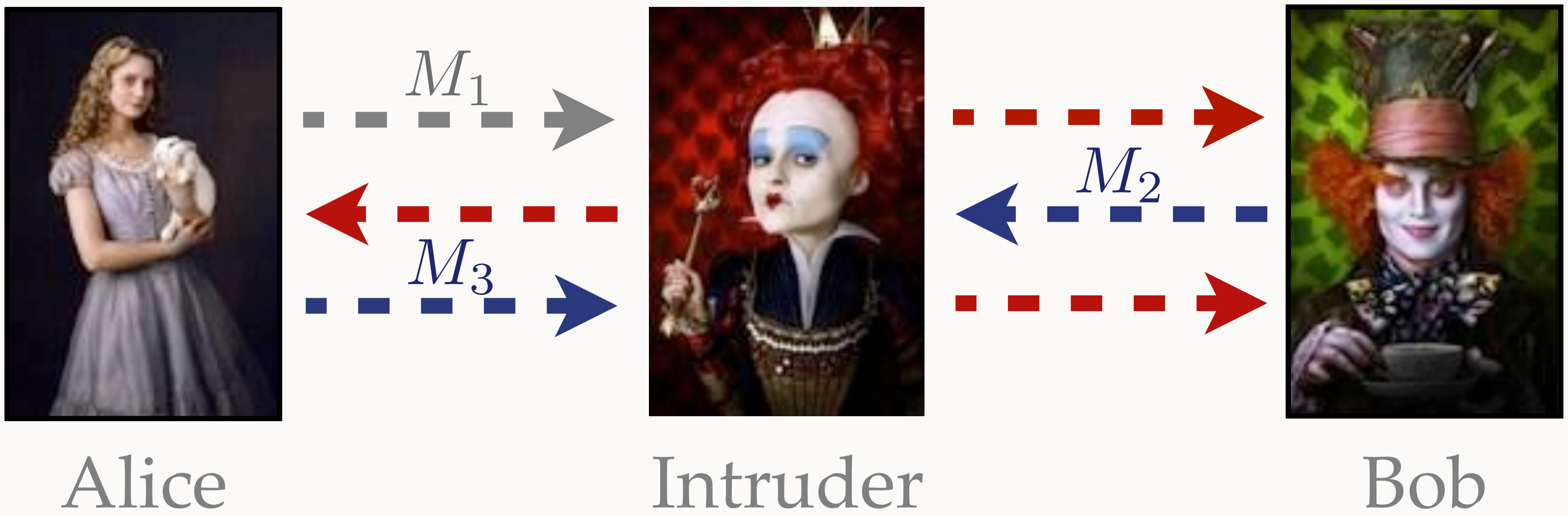
CONTEXT

- Knowledge indistinguishability : static equivalence



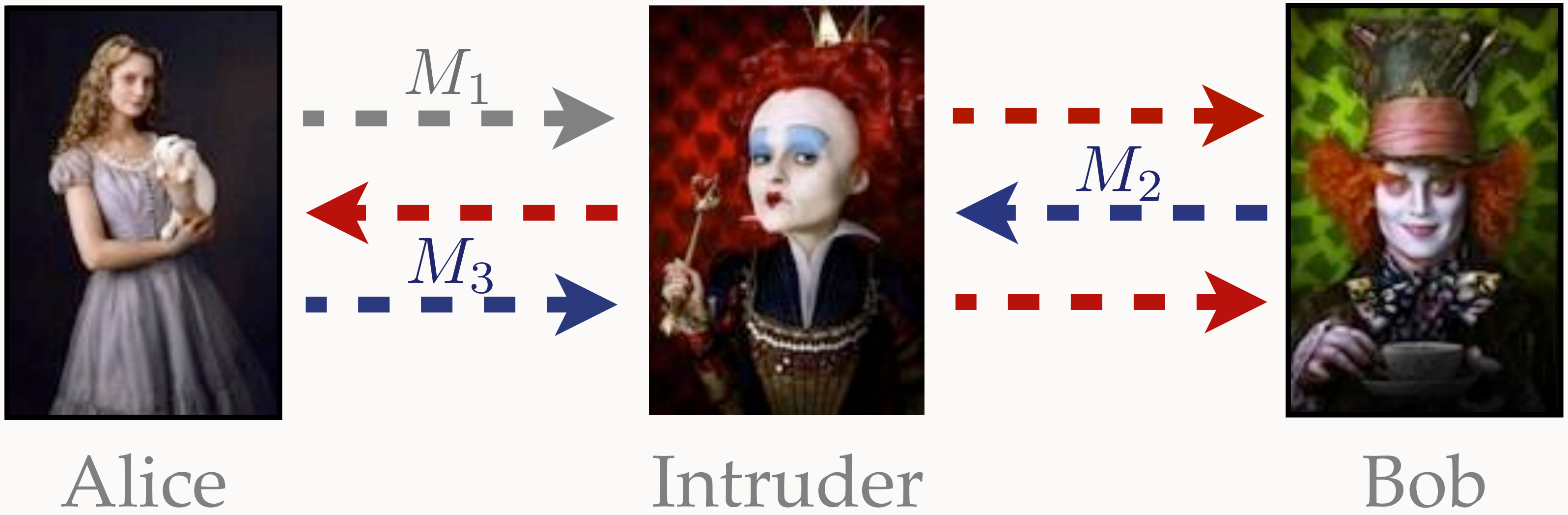
CONTEXT

- Knowledge indistinguishability : static equivalence



CONTEXT

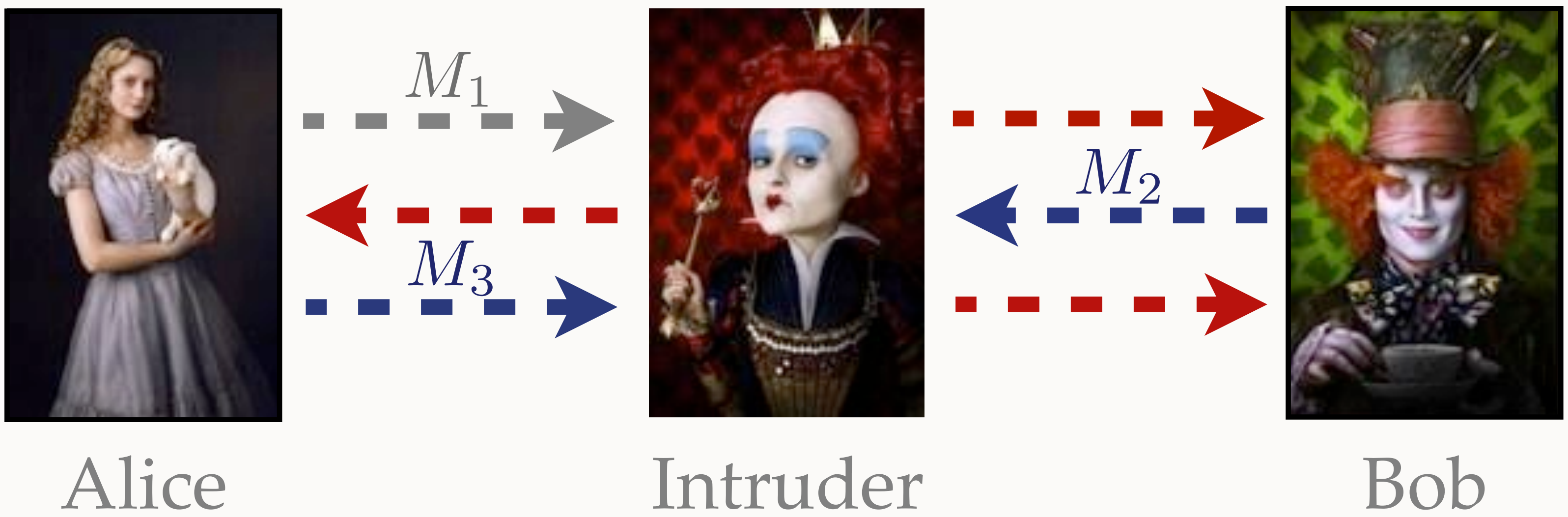
- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

CONTEXT

- Knowledge indistinguishability : static equivalence



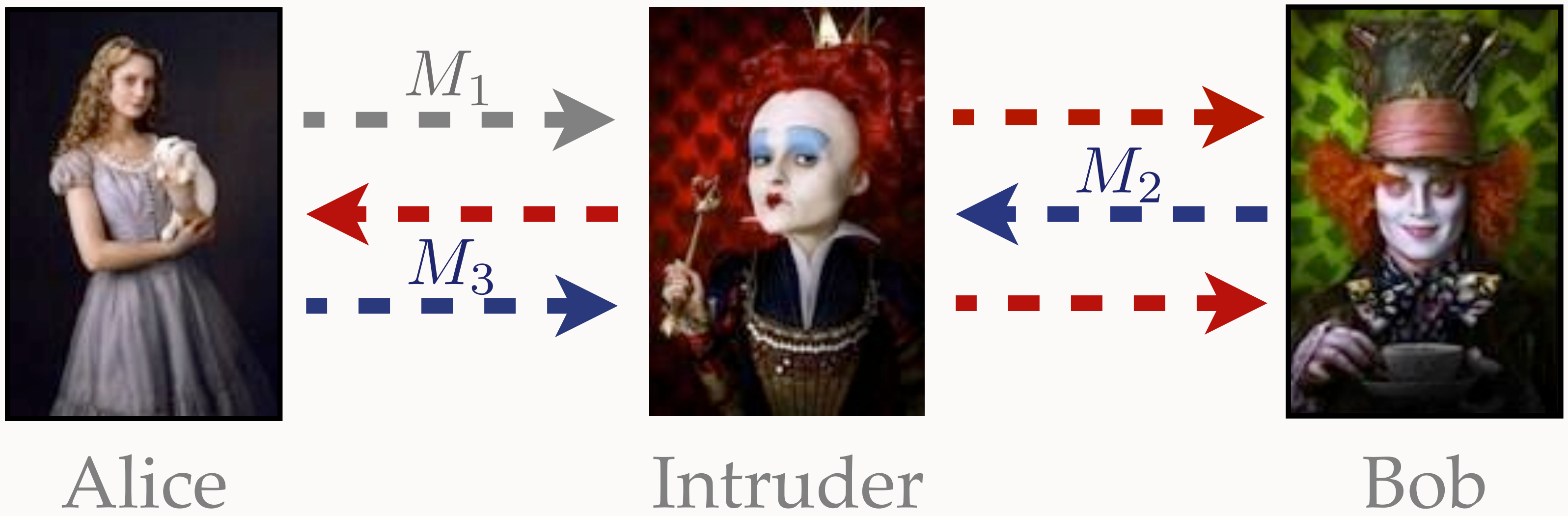
Example with decryption : $dec(\{x\}_y, y) = x$

$\Phi_1 : a, \{b\}_a, b$

$\Phi_2 : c, \{b\}_a, b$

CONTEXT

- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

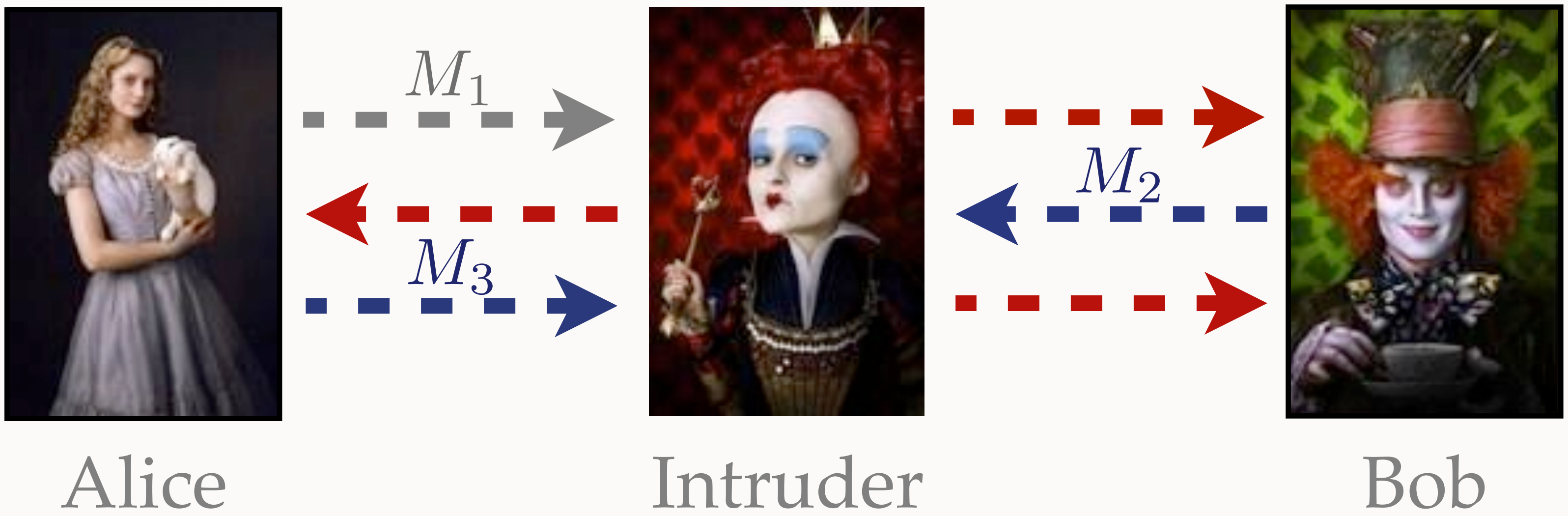
$$\text{dec}(M_2, M_1) = M_3$$

$\Phi_1 : a, \{b\}_a, b$

$\Phi_2 : c, \{b\}_a, b$

CONTEXT

- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

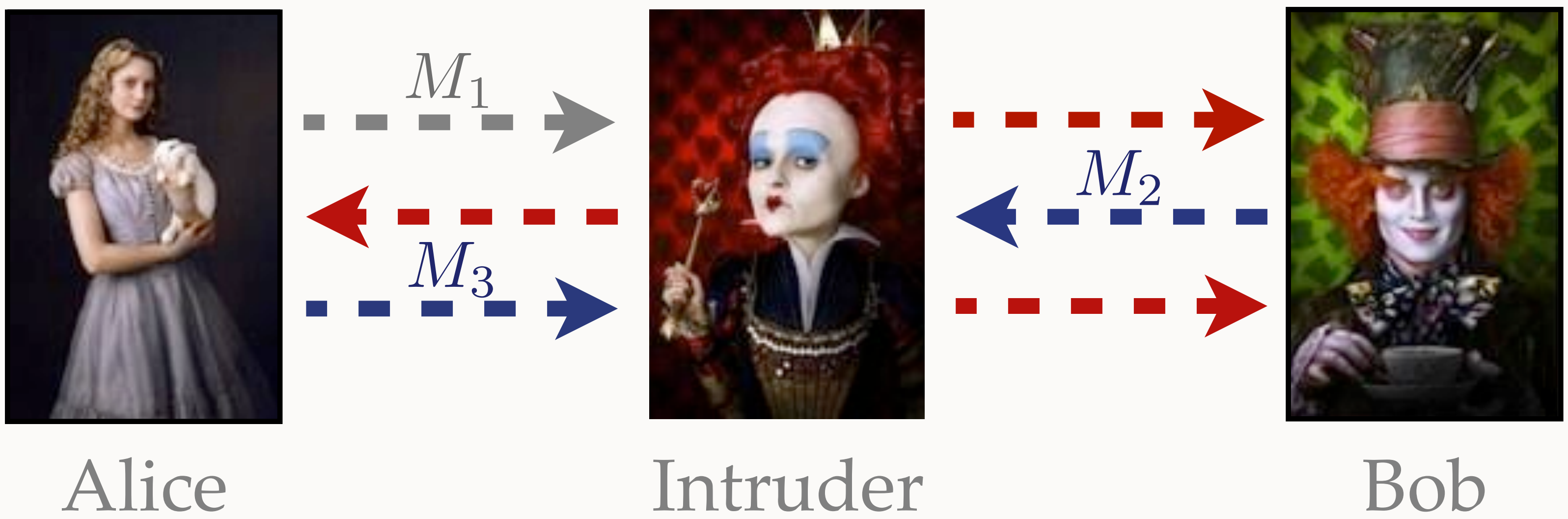
$$\mathrm{dec}(M_2, M_1) = M_3$$

$\Phi_1 : a, \{b\}_a, b$ $\quad$ $\mathrm{dec}(\{b\}_a, a) = b$

$\Phi_2 : c, \{b\}_a, b$ $\quad$ $\mathrm{dec}(\{b\}_a, c) \neq b$

CONTEXT

- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

$$\mathrm{dec}(M_2, M_1) = M_3$$

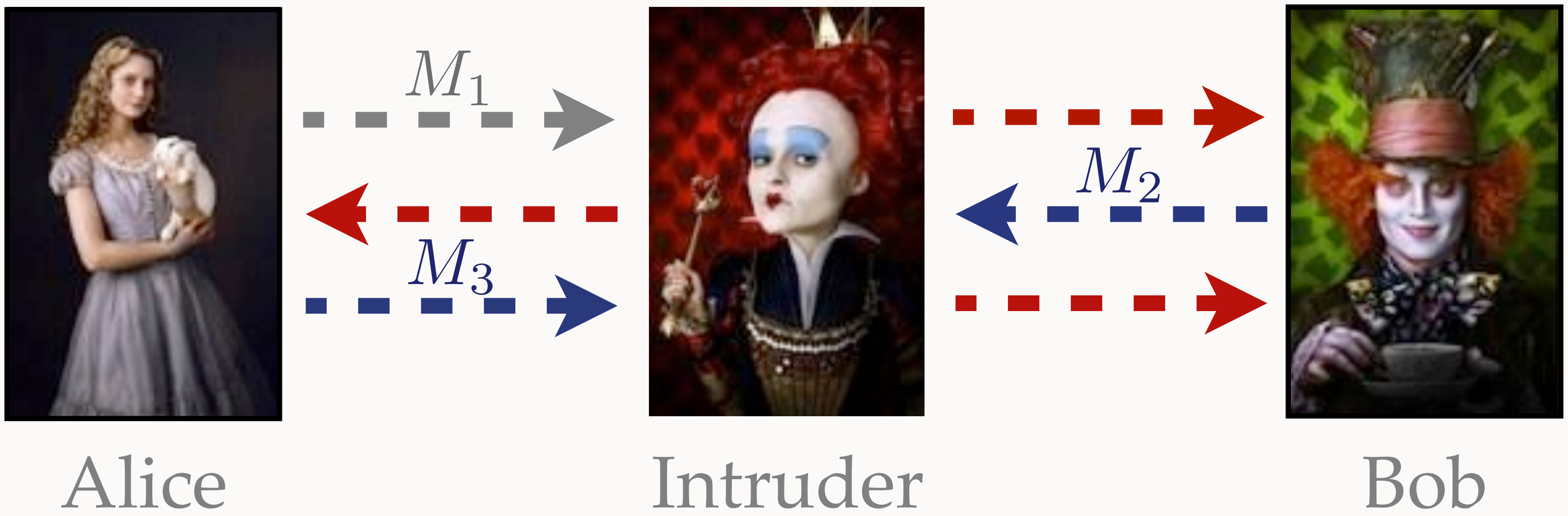
$\Phi_1 : a, \{b\}_a, b$ $\quad$ $\mathrm{dec}(\{b\}_a, a) = b$

$\Phi_2 : c, \{b\}_a, b$ $\quad$ $\mathrm{dec}(\{b\}_a, c) \neq b$

Not equivalent

CONTEXT

- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

$$\text{dec}(M_2, M_1) = M_3$$

$\Phi_1 : a, \{b\}_a, b$ — $\text{dec}(\{b\}_a, a) = b$

$\Phi_2 : c, \{b\}_a, b$ — $\text{dec}(\{b\}_a, c) \neq b$

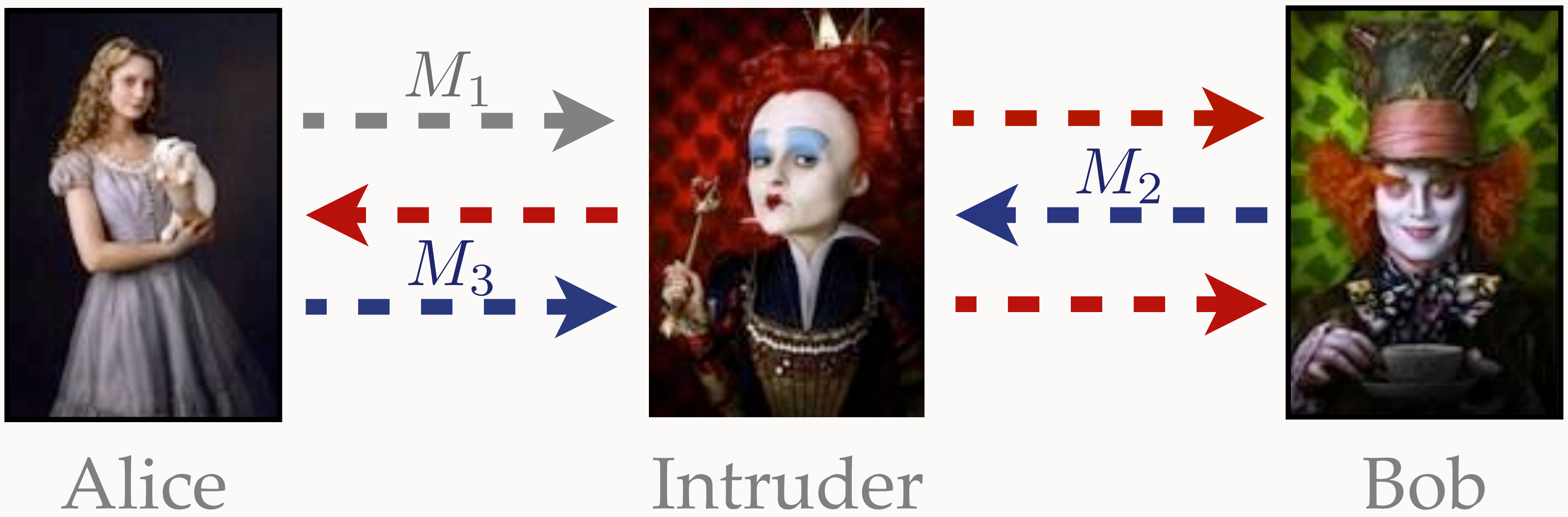
Not equivalent

$\Phi_1 : a, \{b\}_a$

$\Phi_2 : c, \{b\}_a$

CONTEXT

- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

$\text{dec}(M_2, M_1) = M_3$

$\Phi_1 : a, \{b\}_a, b$ $\quad$ $\text{dec}(\{b\}_a, a) = b$

$\Phi_2 : c, \{b\}_a, b$ $\quad$ $\text{dec}(\{b\}_a, c) \neq b$

Not equivalent

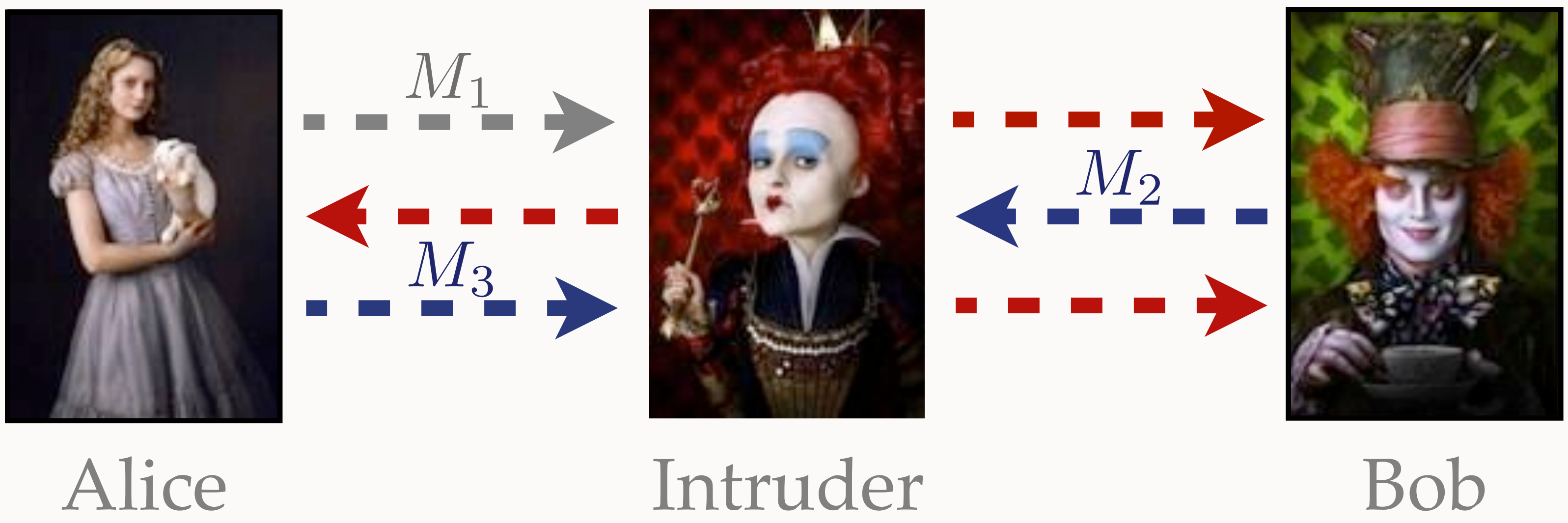
$\Phi_1 : a, \{b\}_a$

$\Phi_2 : c, \{b\}_a$

No test

CONTEXT

- Knowledge indistinguishability : static equivalence



Example with decryption : $dec(\{x\}_y, y) = x$

$\text{dec}(M_2, M_1) = M_3$

$\Phi_1 : a, \{b\}_a, b \quad \text{dec}(\{b\}_a, a) = b$

$\Phi_2 : c, \{b\}_a, b \quad \text{dec}(\{b\}_a, c) \neq b$

Not equivalent

$\Phi_1 : a, \{b\}_a$

$\Phi_2 : c, \{b\}_a$

No test

Equivalent

PREVIOUS WORKS

Most of the previous works focus on stronger equivalence

- A. Tiu and J. E. Dawson. *Automating open bisimulation checking for the spi calculus.*
- M. Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires.* Phd thesis
- B. Blanchet, M. Abadi, and C. Fournet. *Automated verification of selected equivalences for security protocols.*
  - ➡ Tool : B. Blanchet, *ProVerif*

Trace equivalence for simple processes without else branches

- V. Cortier and S. Delaune. *A method for proving observational equivalence.*

MOTIVATION

- Example

Two problematic examples :

- e-passport protocols : M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. *Analysing unlinkability and anonymity using the applied pi calculus.*
- private authentication protocol : M. Abadi and C. Fournet. *Private authentication. Theoretical Computer Science.*

MOTIVATION

- Example

Two problematic examples :

- e-passport protocols : M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. *Analysing unlinkability and anonymity using the applied pi calculus.*
- private authentication protocol : M. Abadi and C. Fournet. *Private authentication. Theoretical Computer Science.*

Alice

$$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$$

Bob

MOTIVATION

- Example

Two problematic examples :

- e-passport protocols : M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. *Analysing unlinkability and anonymity using the applied pi calculus.*
- private authentication protocol : M. Abadi and C. Fournet. *Private authentication. Theoretical Computer Science.*

Alice

$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$

$pk(k_A)?$

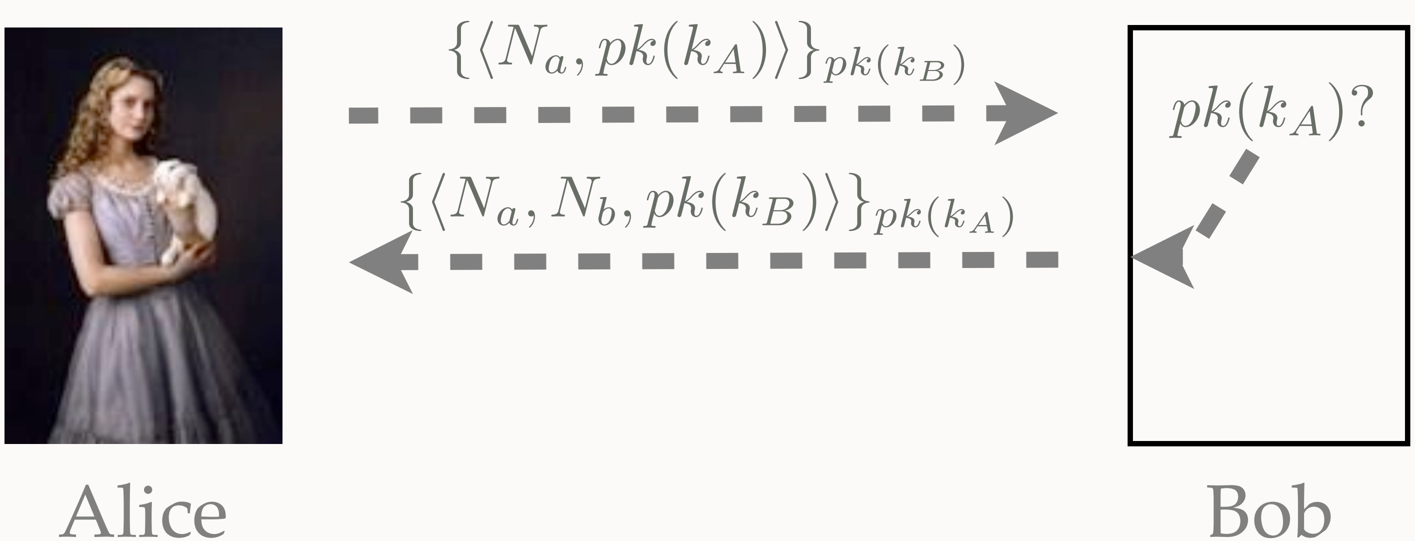
Bob

MOTIVATION

- Example

Two problematic examples :

- e-passport protocols : M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. *Analysing unlinkability and anonymity using the applied pi calculus.*
- private authentication protocol : M. Abadi and C. Fournet. *Private authentication. Theoretical Computer Science.*

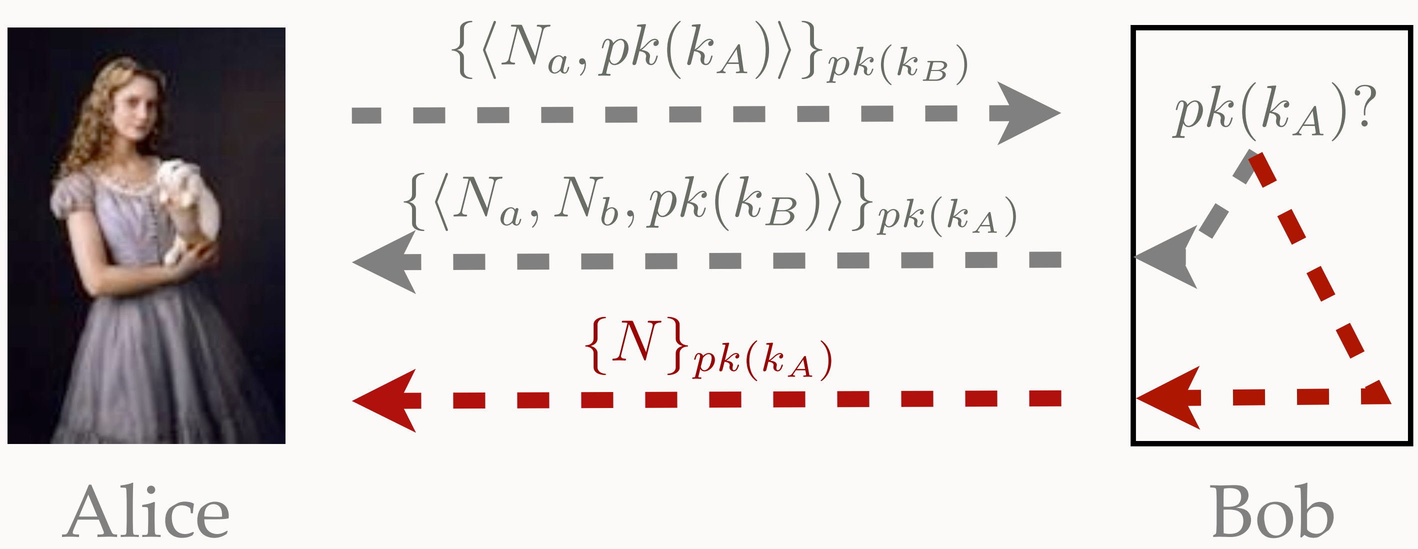


MOTIVATION

- Example

Two problematic examples :

- e-passport protocols : M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. *Analysing unlinkability and anonymity using the applied pi calculus.*
- private authentication protocol : M. Abadi and C. Fournet. *Private authentication. Theoretical Computer Science.*

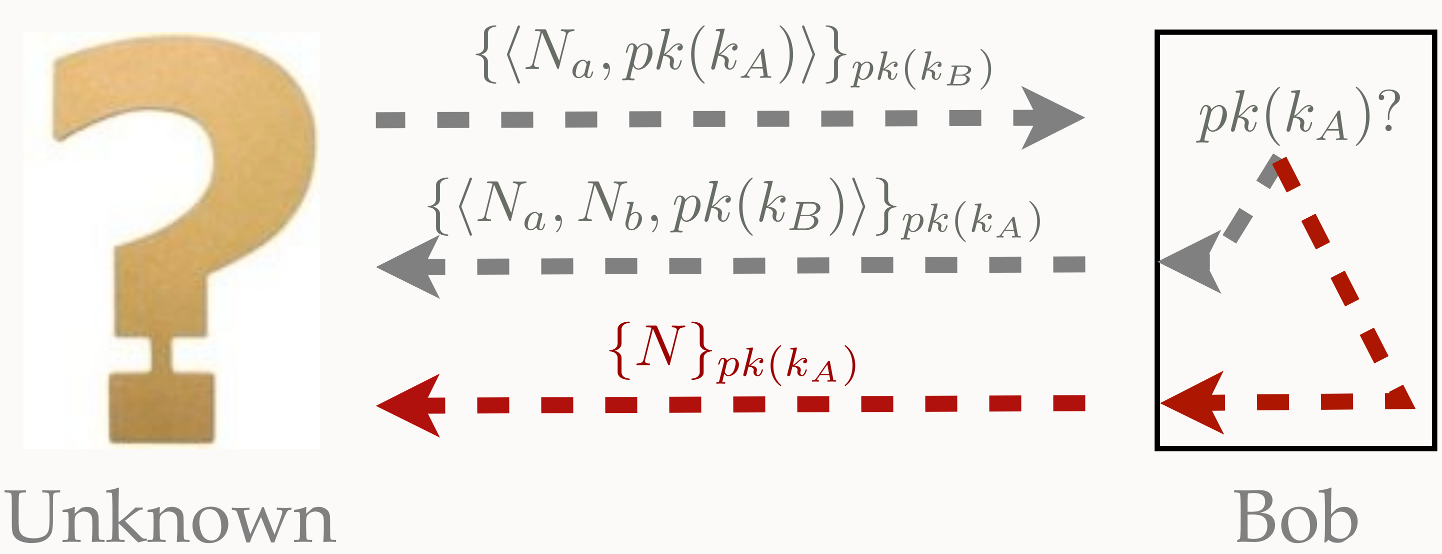


MOTIVATION

- Example

Two problematic examples :

- e-passport protocols : M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. *Analysing unlinkability and anonymity using the applied pi calculus.*
- private authentication protocol : M. Abadi and C. Fournet. *Private authentication. Theoretical Computer Science.*



MOTIVATION

- Example

Alice

Intruder

Bob

Charlene

Intruder

Bob

MOTIVATION

- Example

Alice

Bob

Charlene

Bob

MOTIVATION

- Example

$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$

Alice

Bob

Charlene

Bob

MOTIVATION

- Example

$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$

$\{\langle x, y\rangle\}_{pk(k_B)}$

Alice

Bob

Charlene

Bob

MOTIVATION

- Example

$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$ →

$\{\langle x, y\rangle\}_{pk(k_B)}$ →

$pk(k_A) = y$

Alice

Bob

Charlene

Bob

MOTIVATION

- Example

$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$

$\{\langle x, y\rangle\}_{pk(k_B)}$

$pk(k_A) = y$

$\{\langle x, N_b, pk(k_B)\rangle\}_y$

Alice

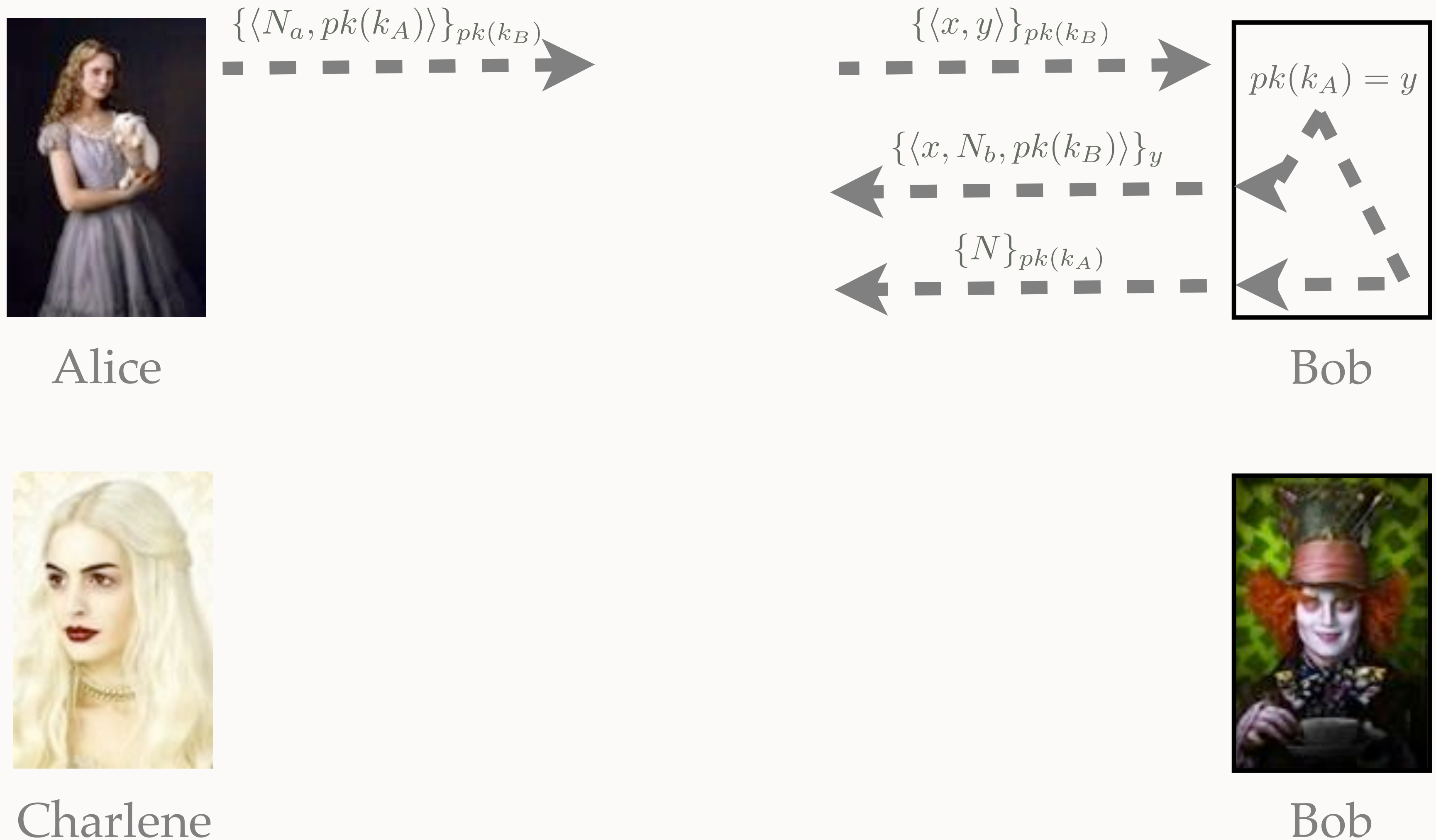
Bob

Charlene

Bob

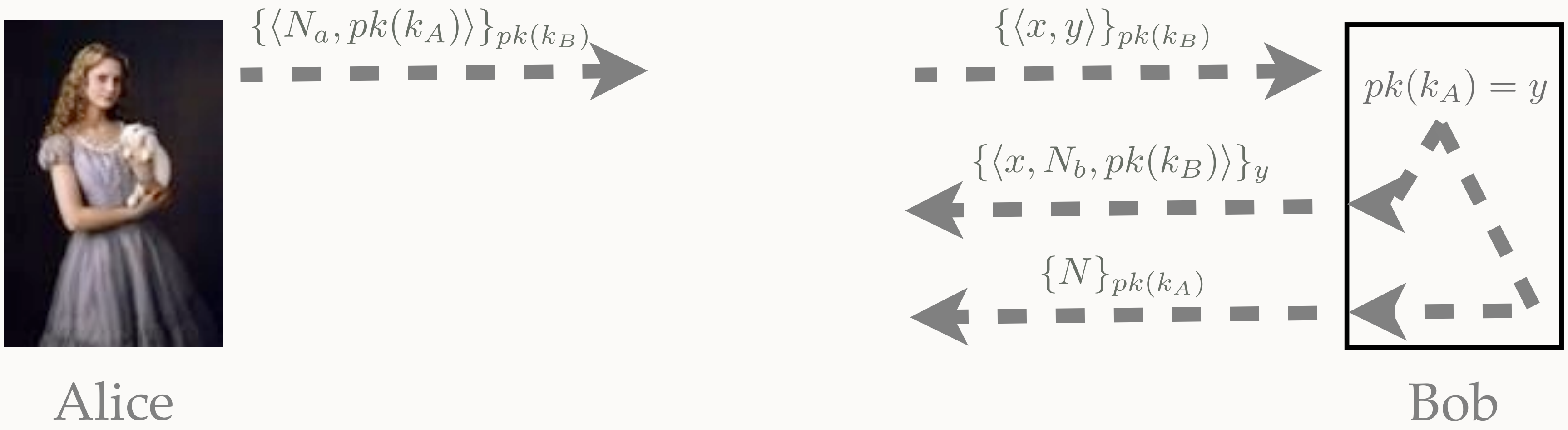
MOTIVATION

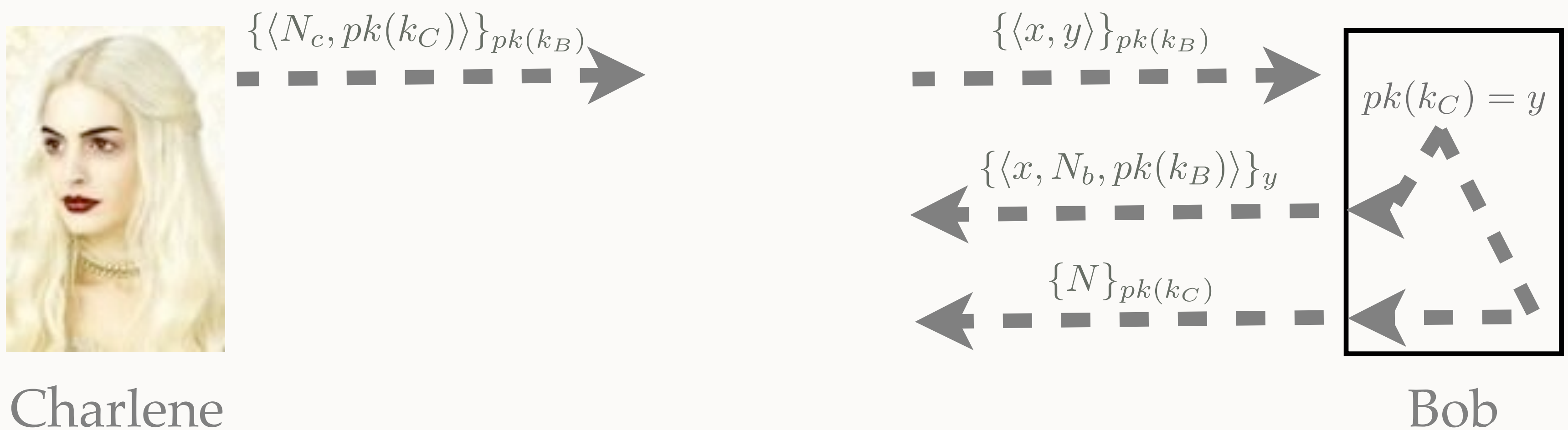
- Example



MOTIVATION

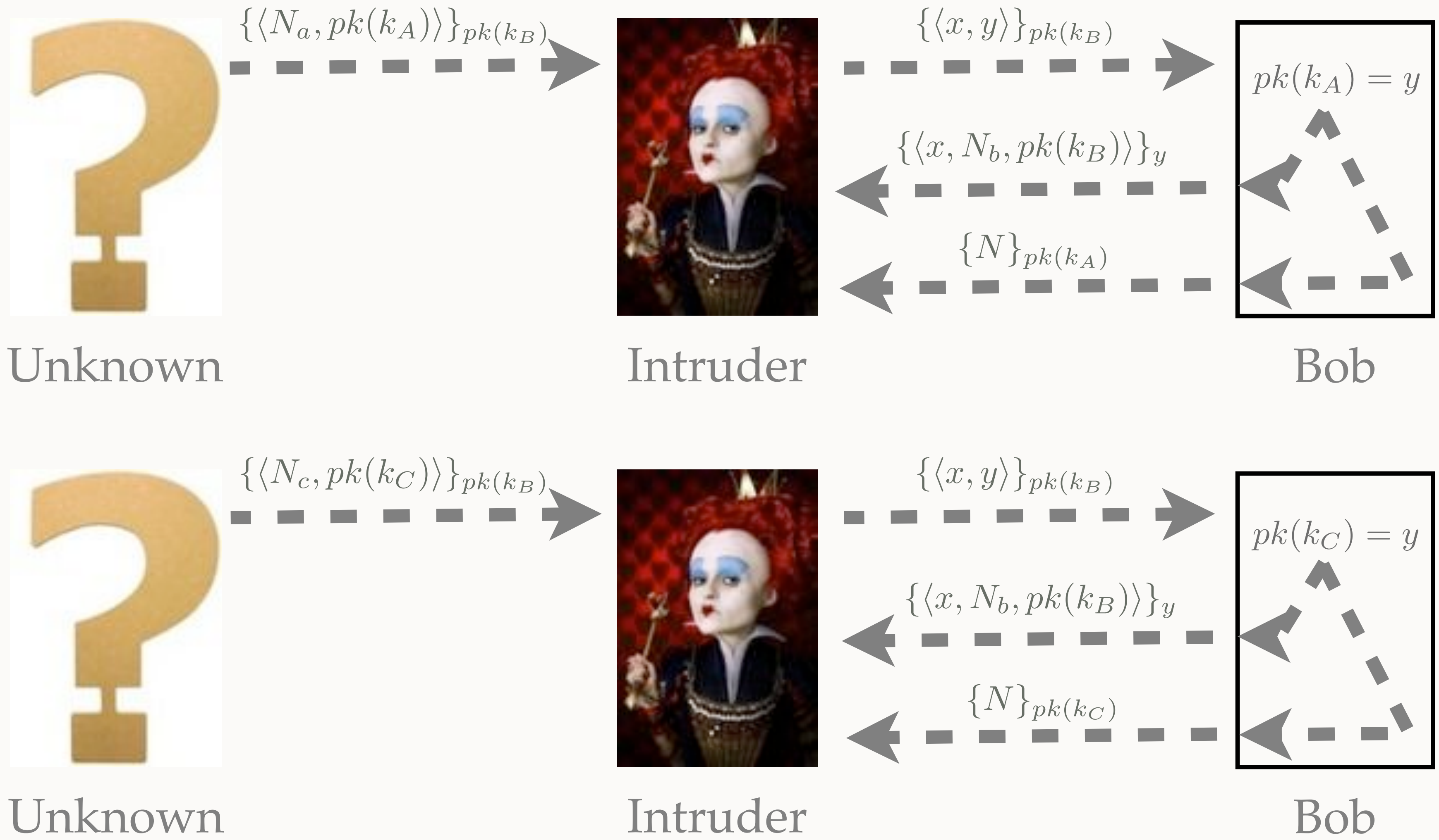
- Example





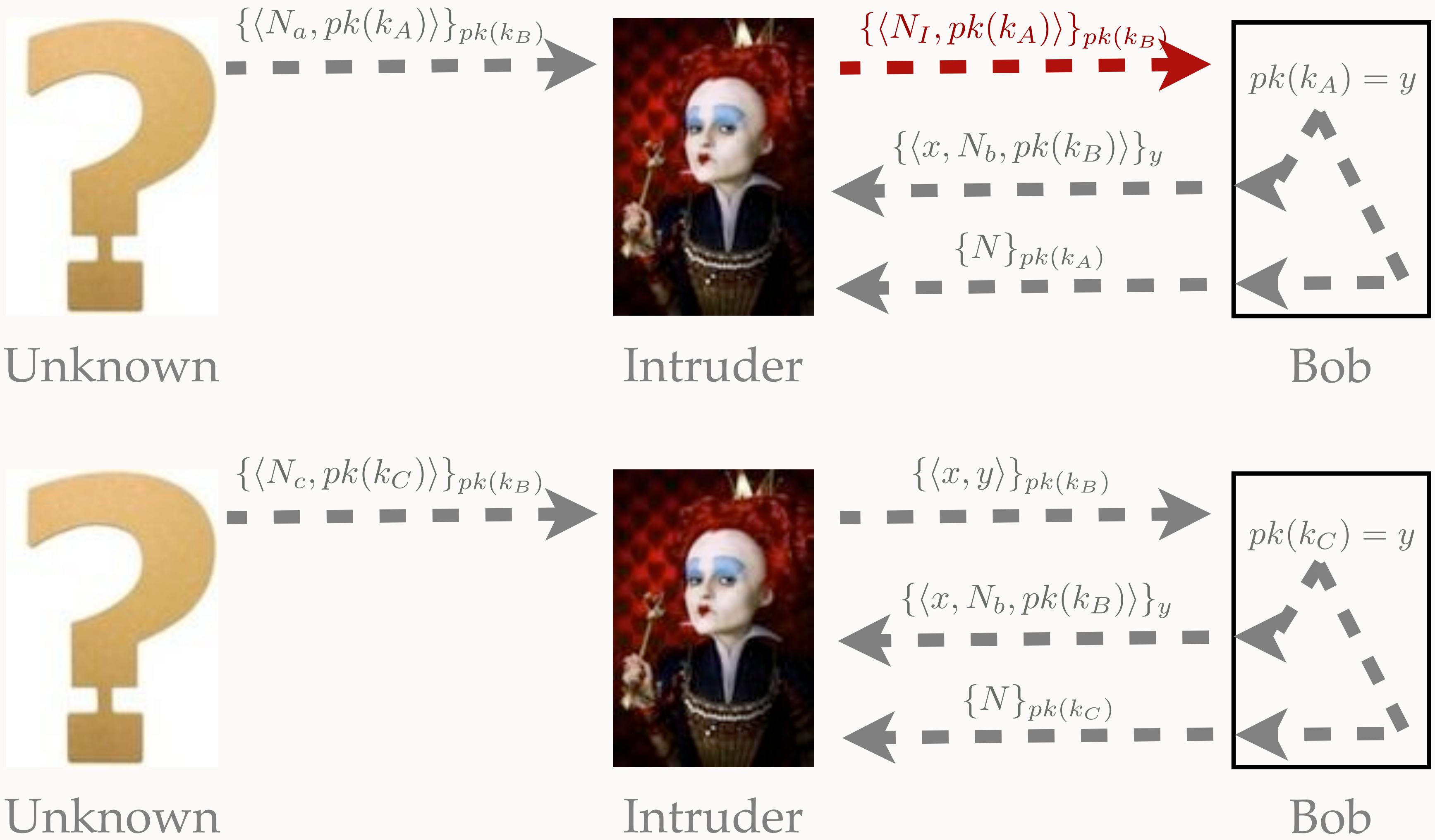
MOTIVATION

- Example



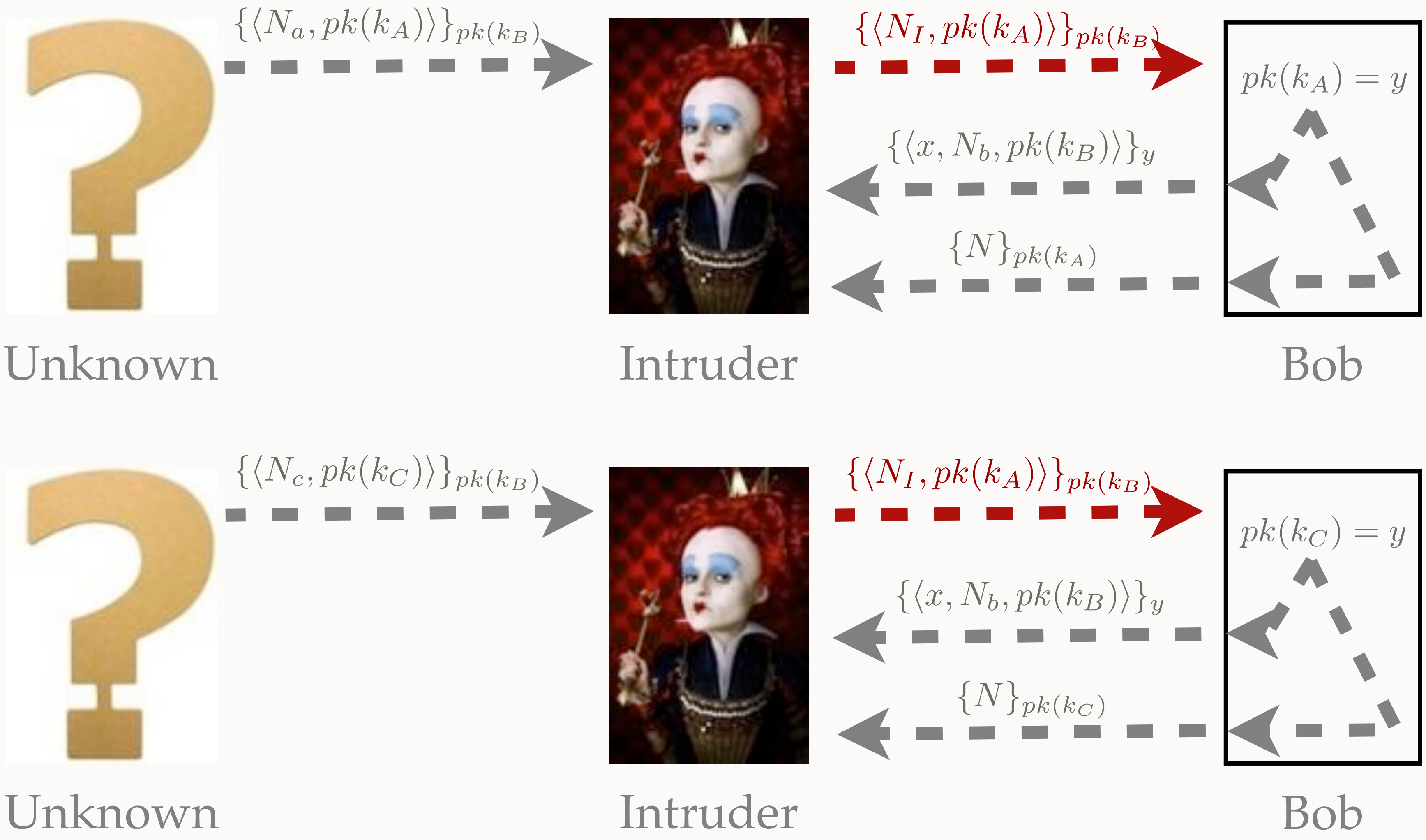
MOTIVATION

- Example



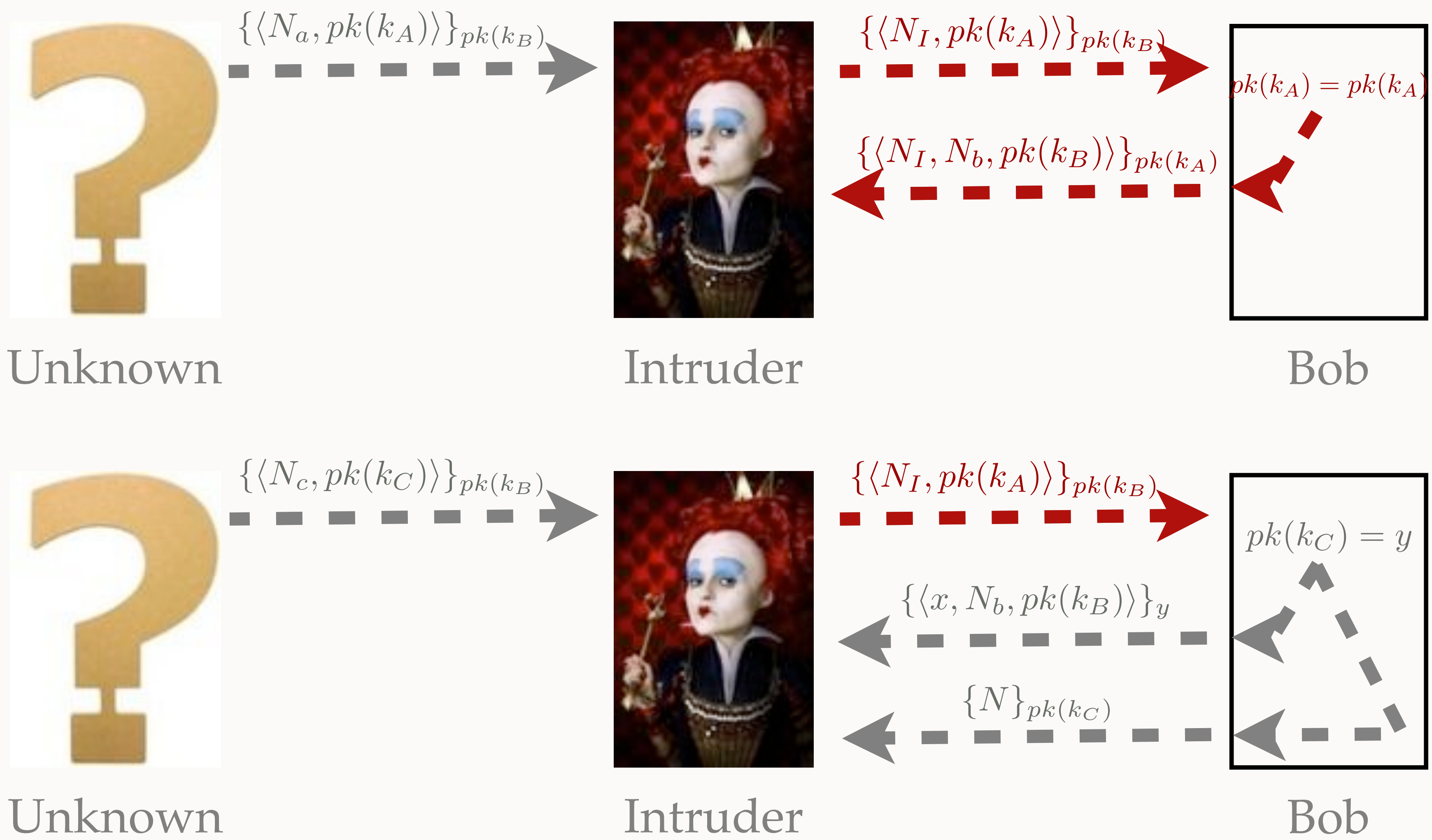
MOTIVATION

- Example



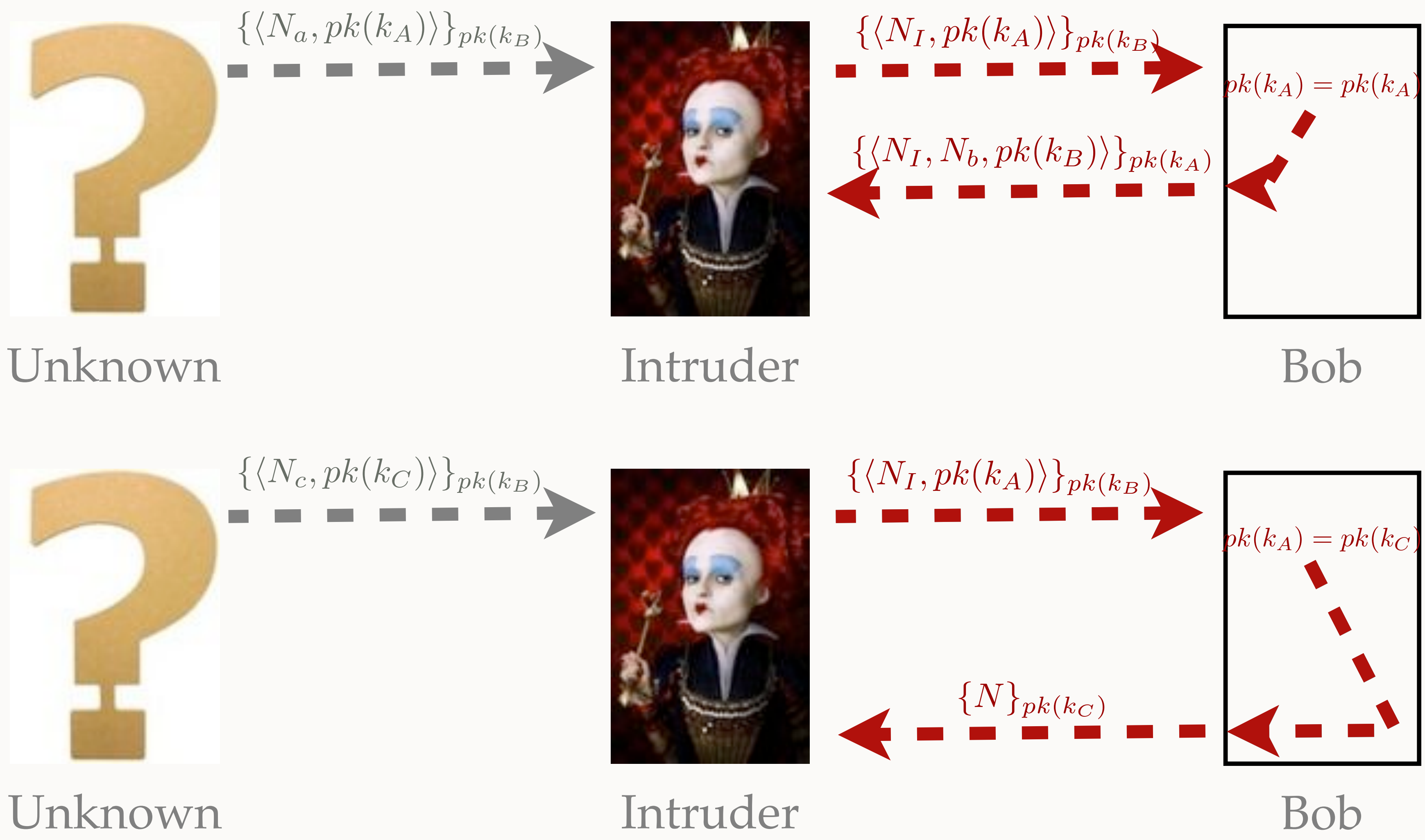
MOTIVATION

- Example



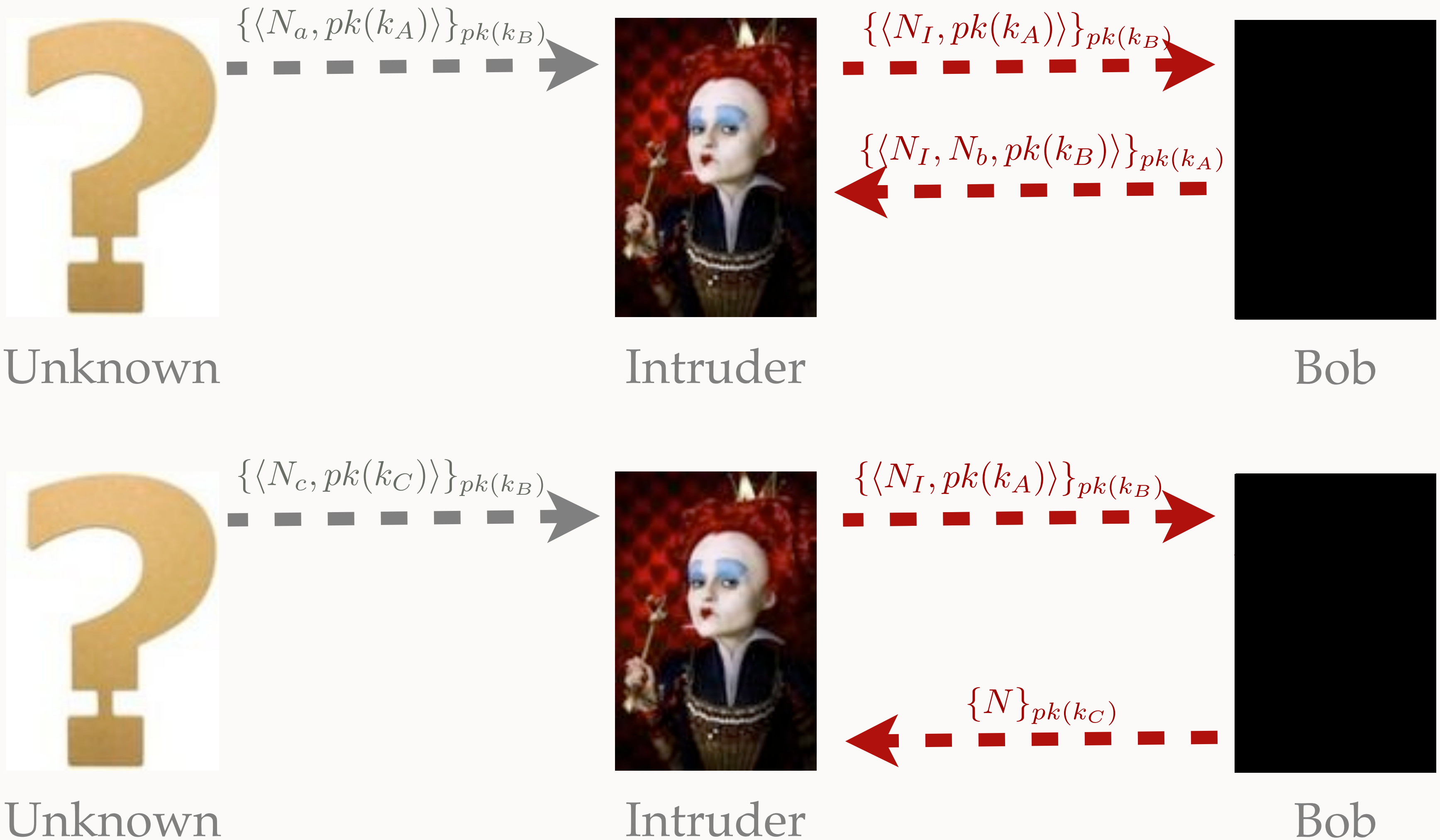
MOTIVATION

- Example



MOTIVATION

- Example



CONTRIBUTION

Decision procedure for verification of trace equivalence

- Infinitely many traces are represented by symbolic constraint system

+ Protocol possibly non-determinist and with non trivial else branches
+ Private channels

- Fixed set of cryptographic primitives : symmetric and asymmetric encryption, pairing and signature
- Bounded number of sessions (no replication in the process algebra)

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces

Alice

Intruder

Bob

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces

Alice

Intruder

Bob

$pk(k_A), pk(k_B), pk(k_C), N_I$

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces

$\{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$

Alice

Intruder

Bob

$$pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}$$

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces

$$pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \overset{?}{\vdash} \{\langle x, y\rangle\}_{pk(k_B)}$$

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces

$$pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \overset{?}{\vdash} \{\langle x, y\rangle\}_{pk(k_B)}$$

$$y \overset{?}{=} pk(k_A)$$

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces

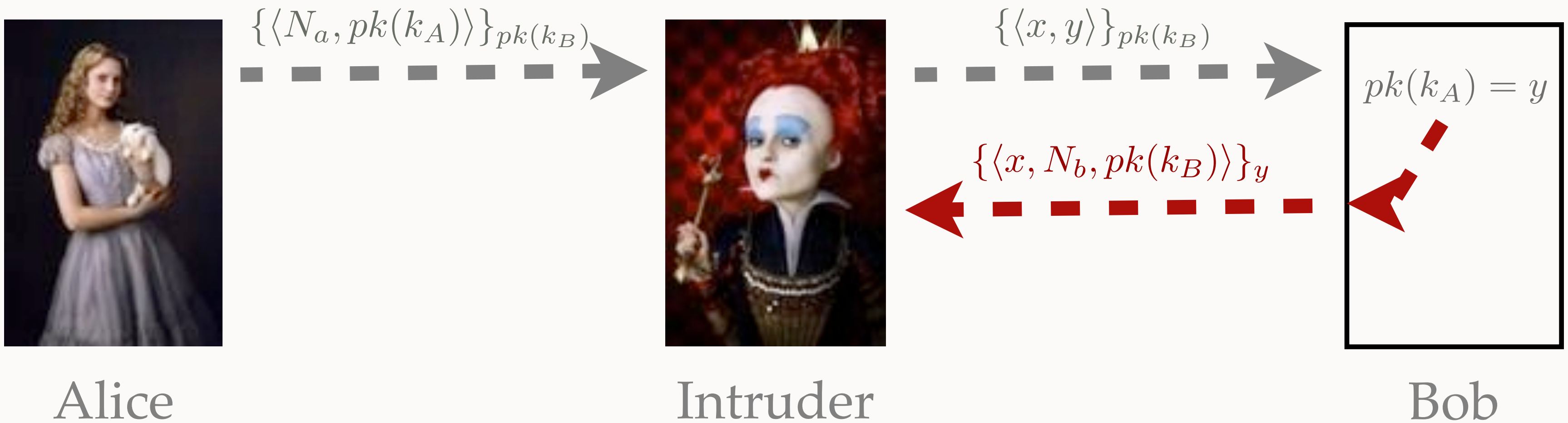
$$pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \overset{?}{\vdash} \{\langle x, y\rangle\}_{pk(k_B)}$$

$$pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$y \overset{?}{=} pk(k_A)$$

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces



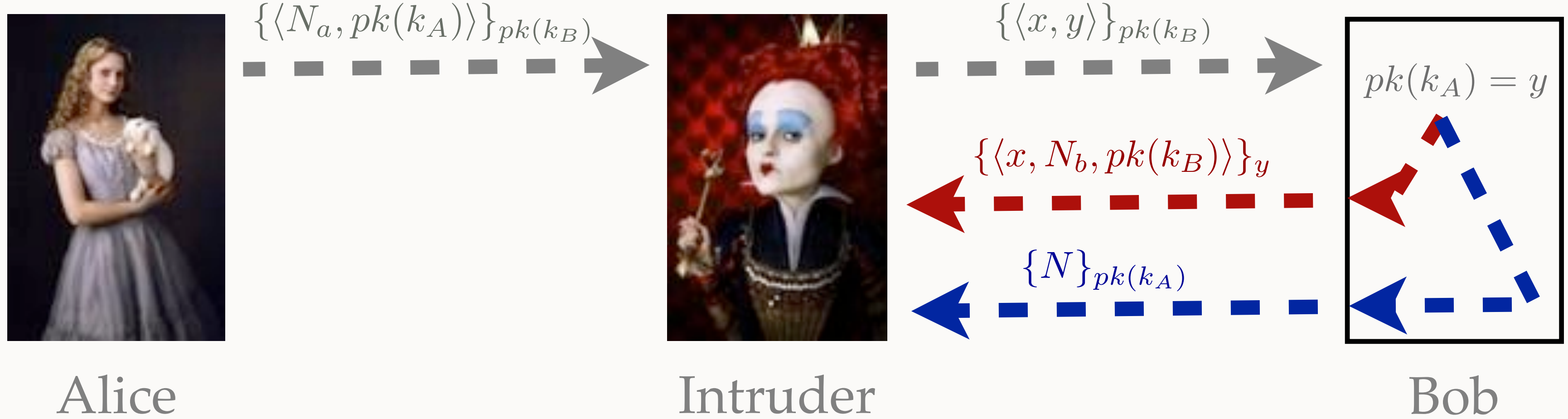
$$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \overset{?}{\vdash} \{\langle x, y\rangle\}_{pk(k_B)}$$

$$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$E: \; y \overset{?}{=} pk(k_A)$$

CONSTRAINT SYSTEM

- One constraint system = one interleaving = several traces



$$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \overset{?}{\vdash} \{\langle x, y\rangle\}_{pk(k_B)}$$

$$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$E: \; y \overset{?}{=} pk(k_A)$$

$$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \overset{?}{\vdash} \{\langle x, y\rangle\}_{pk(k_B)}$$

$$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{N\}_{pk(k_A)}$$

$$E: \; y \overset{?}{\neq} pk(k_A)$$

CONSTRAINT SYSTEM

- One solution of a constraint system = one trace

$$D:\ pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \vdash \{\langle x, y\rangle\}_{pk(k_B)}$$

$$\Phi:\ pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$E:\ y = pk(k_A)$$

CONSTRAINT SYSTEM

- One solution of a constraint system = one trace

$$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \vdash \{\langle x, y\rangle\}_{pk(k_B)}$$

$$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$E: \; y = pk(k_A)$$

A solution is a pair of substitution (σ, θ) where :

- σ describe the messages
- θ describe how the messages are deduced

CONSTRAINT SYSTEM

- One solution of a constraint system = one trace

$$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \vdash \{\langle x, y\rangle\}_{pk(k_B)}$$

$$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$E: \; y = pk(k_A)$$

A solution is a pair of substitution (σ, θ) where :

- σ describe the messages
- θ describe how the messages are deduced

$$\sigma = \{x \rightarrow N_I;\; y \rightarrow pk(k_A)\}$$

CONSTRAINT SYSTEM

- One solution of a constraint system = one trace

$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \vdash \{\langle x, y\rangle\}_{pk(k_B)}$ $\quad X_1$

$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$

$ax_1 \quad ax_2 \quad ax_3 \quad ax_4 \quad ax_5 \quad ax_6$

$E: \; y = pk(k_A)$

A solution is a pair of substitution (σ, θ) where :

- σ describe the messages
- θ describe how the messages are deduced

$$\sigma = \{x \rightarrow N_I;\; y \rightarrow pk(k_A)\}$$

CONSTRAINT SYSTEM

- One solution of a constraint system = one trace

$$D: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \vdash \{\langle x, y\rangle\}_{pk(k_B)} \qquad X_1$$

$$\Phi: \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

$$ax_1 \quad ax_2 \quad ax_3 \quad ax_4 \quad ax_5 \quad ax_6$$

$$E: \; y = pk(k_A)$$

A solution is a pair of substitution (σ, θ) where :

- σ describe the messages
- θ describe how the messages are deduced

$$\sigma = \{x \rightarrow N_I; \; y \rightarrow pk(k_A)\}$$

$$\theta = \{X_1 \rightarrow \{\langle ax_4, ax_1\rangle\}_{ax_2}\}$$

CONSTRAINT SYSTEM

- One solution of a constraint system = one trace

$$D: \ pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)} \vdash \{\langle x, y\rangle\}_{pk(k_B)} \qquad X_1$$

$$\Phi: \ pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A)\rangle\}_{pk(k_B)}, \{\langle x, N_b, pk(k_B)\rangle\}_y$$

ax_1 ax_2 ax_3 ax_4 ax_5 ax_6

$$E: \ y = pk(k_A)$$

A solution is a pair of substitution (σ, θ) where :

- σ describe the messages
- θ describe how the messages are deduced

$$\sigma = \{x \rightarrow N_I; \ y \rightarrow pk(k_A)\}$$

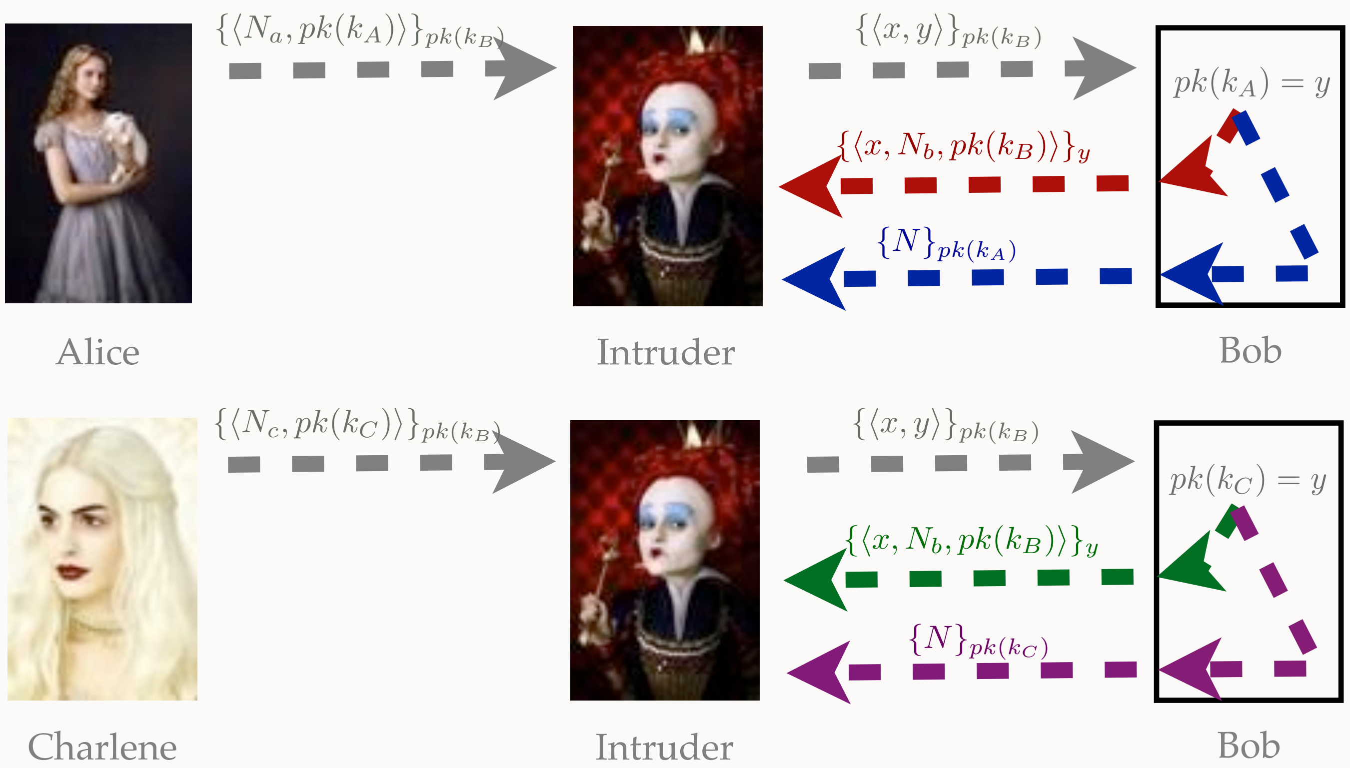
$$\theta = \{X_1 \rightarrow \{\langle ax_4, ax_1\rangle\}_{ax_2}\}$$

$$\sigma = \{x \rightarrow N_a; \ y \rightarrow pk(k_A)\}$$

$$\theta = \{X_1 \rightarrow ax_5\}$$

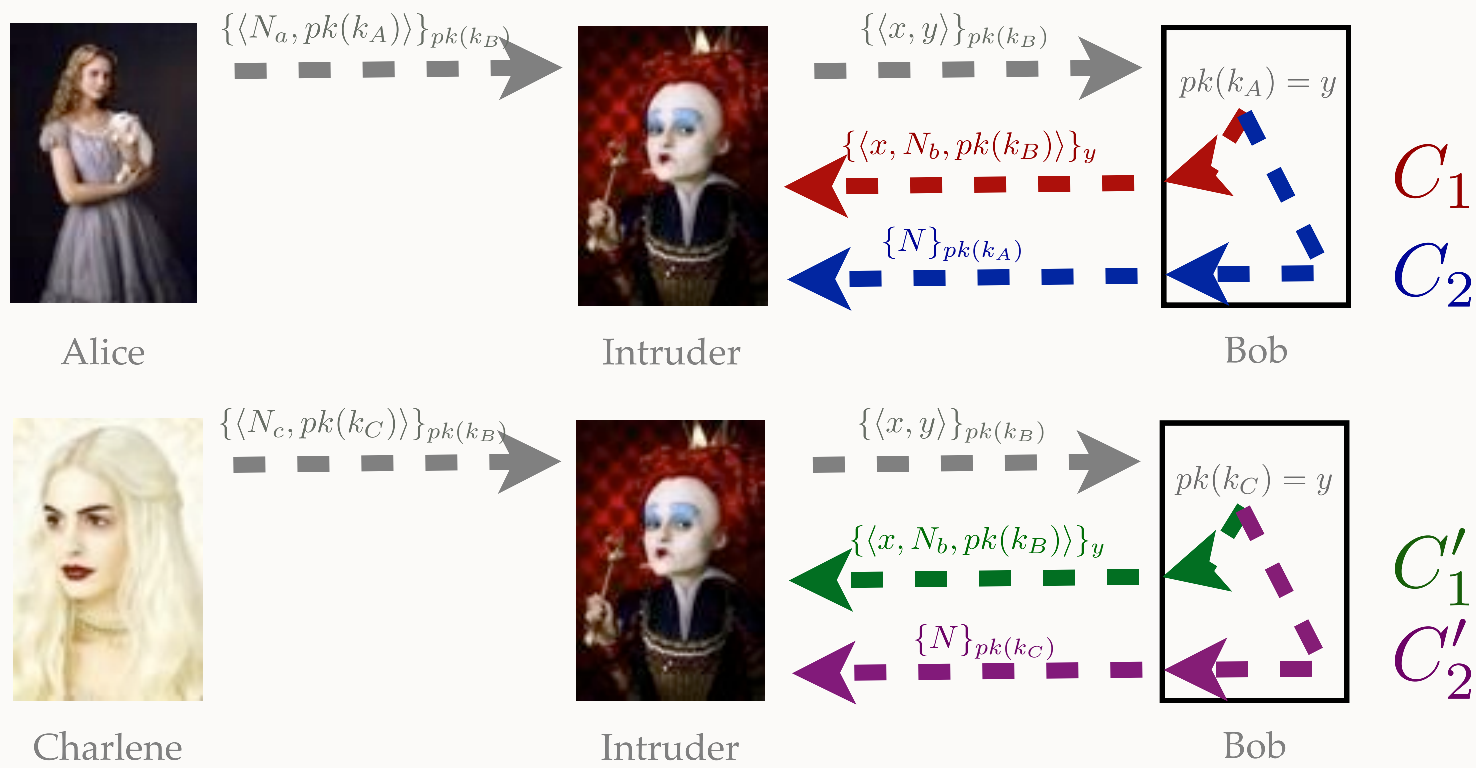
CONSTRAINT SYSTEM

- Set of constraint systems



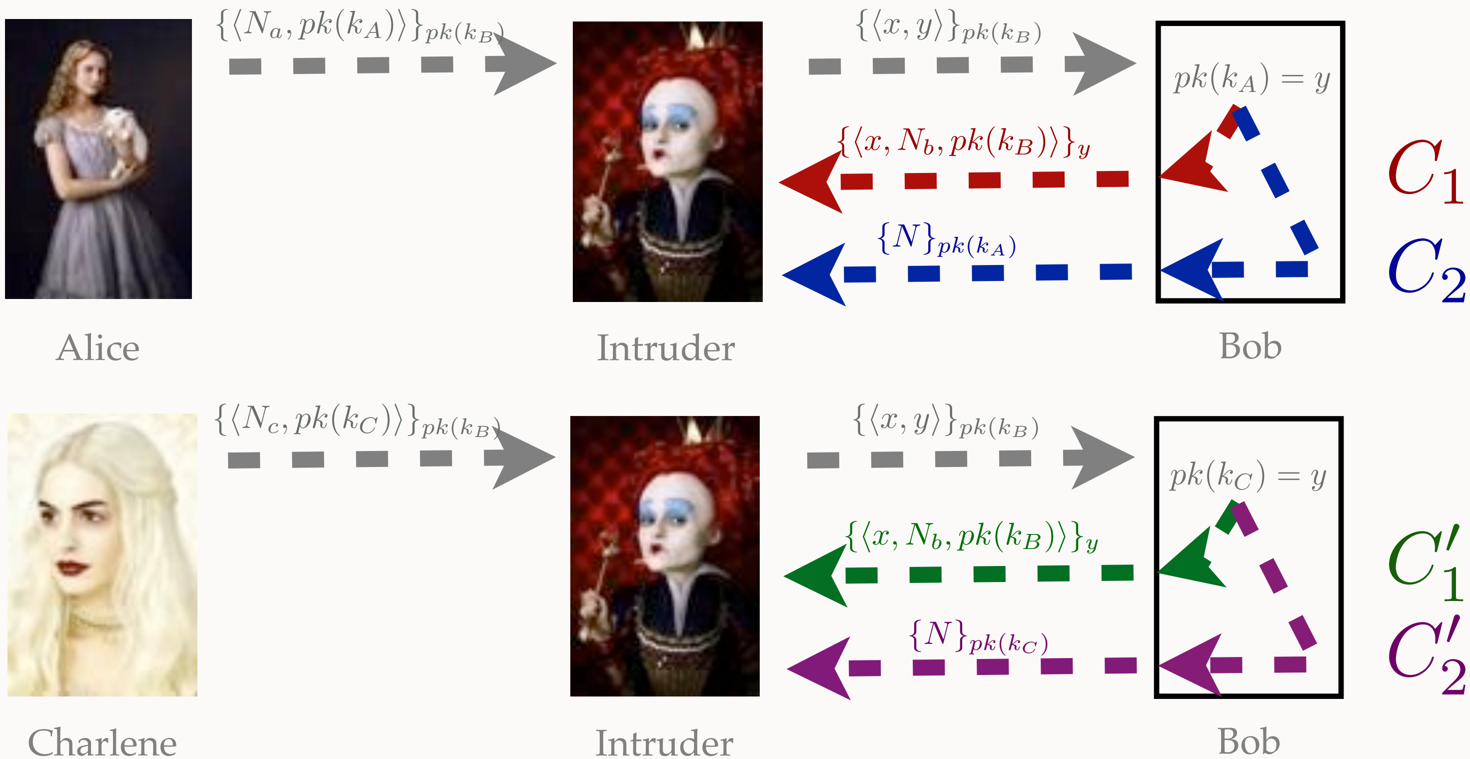
CONSTRAINT SYSTEM

- Set of constraint systems



CONSTRAINT SYSTEM

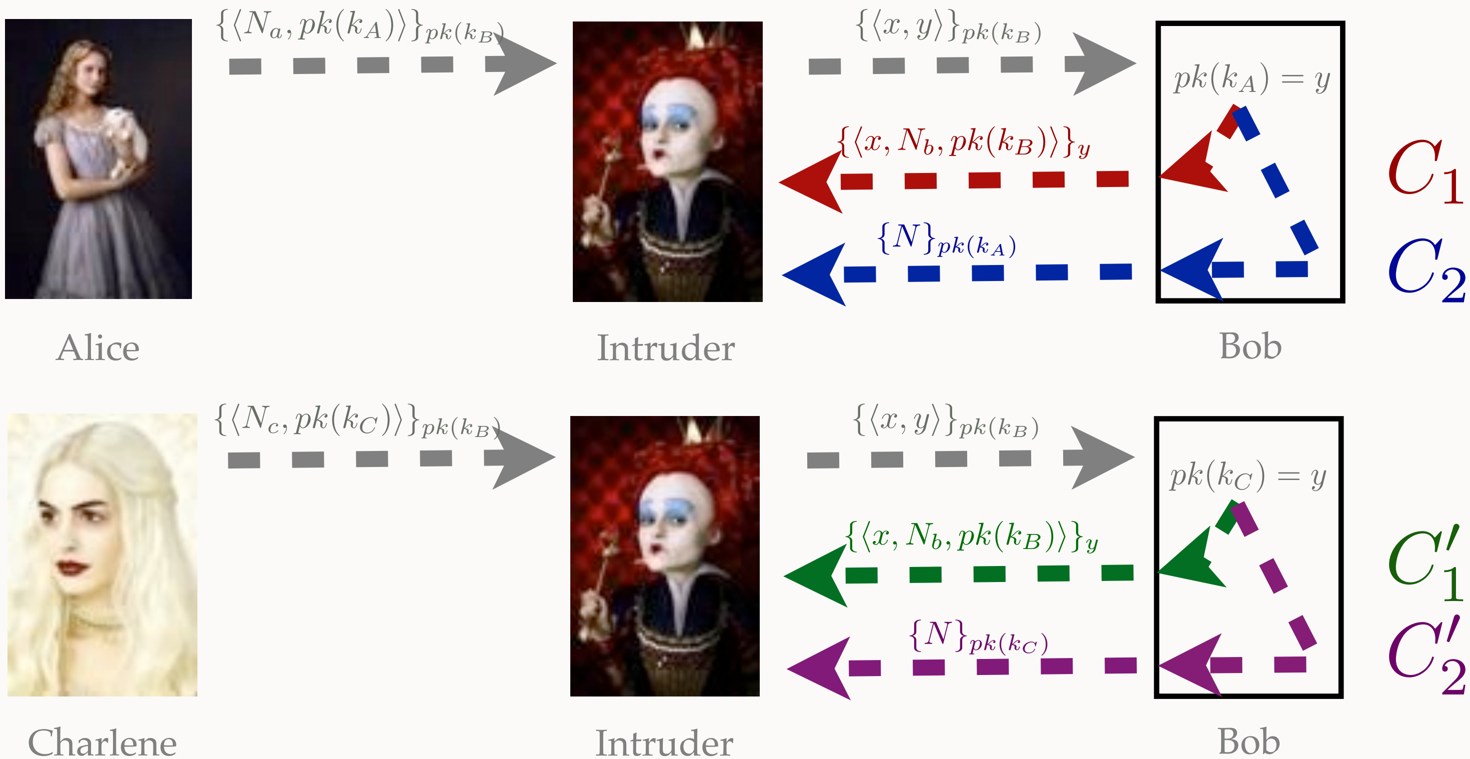
- Set of constraint systems



$$\{C_1;\ C_2\} \approx \{C'_1;\ C'_2\}$$

CONSTRAINT SYSTEM

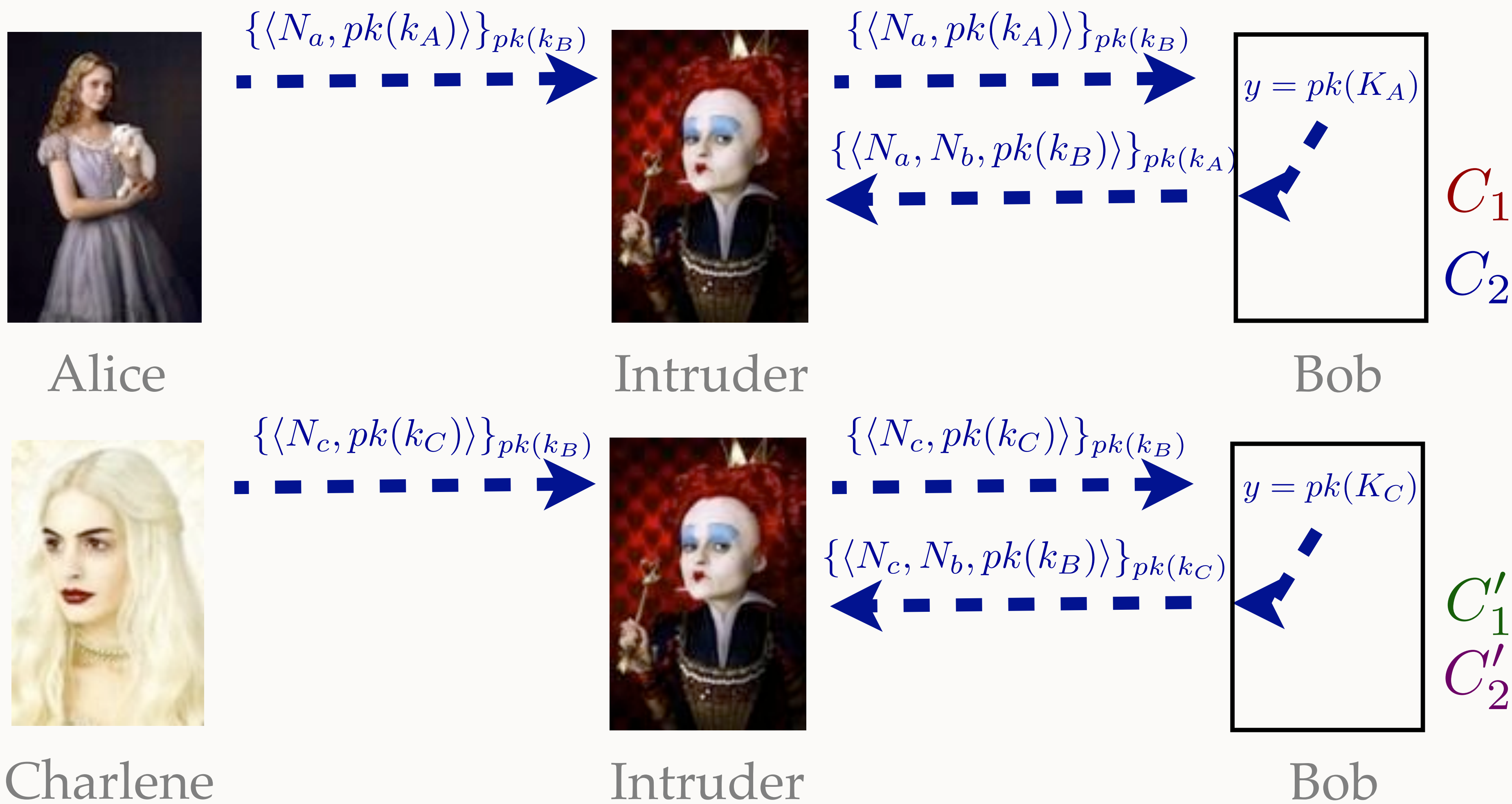
- Set of constraint systems



Symbolic equivalence between sets of constraint systems

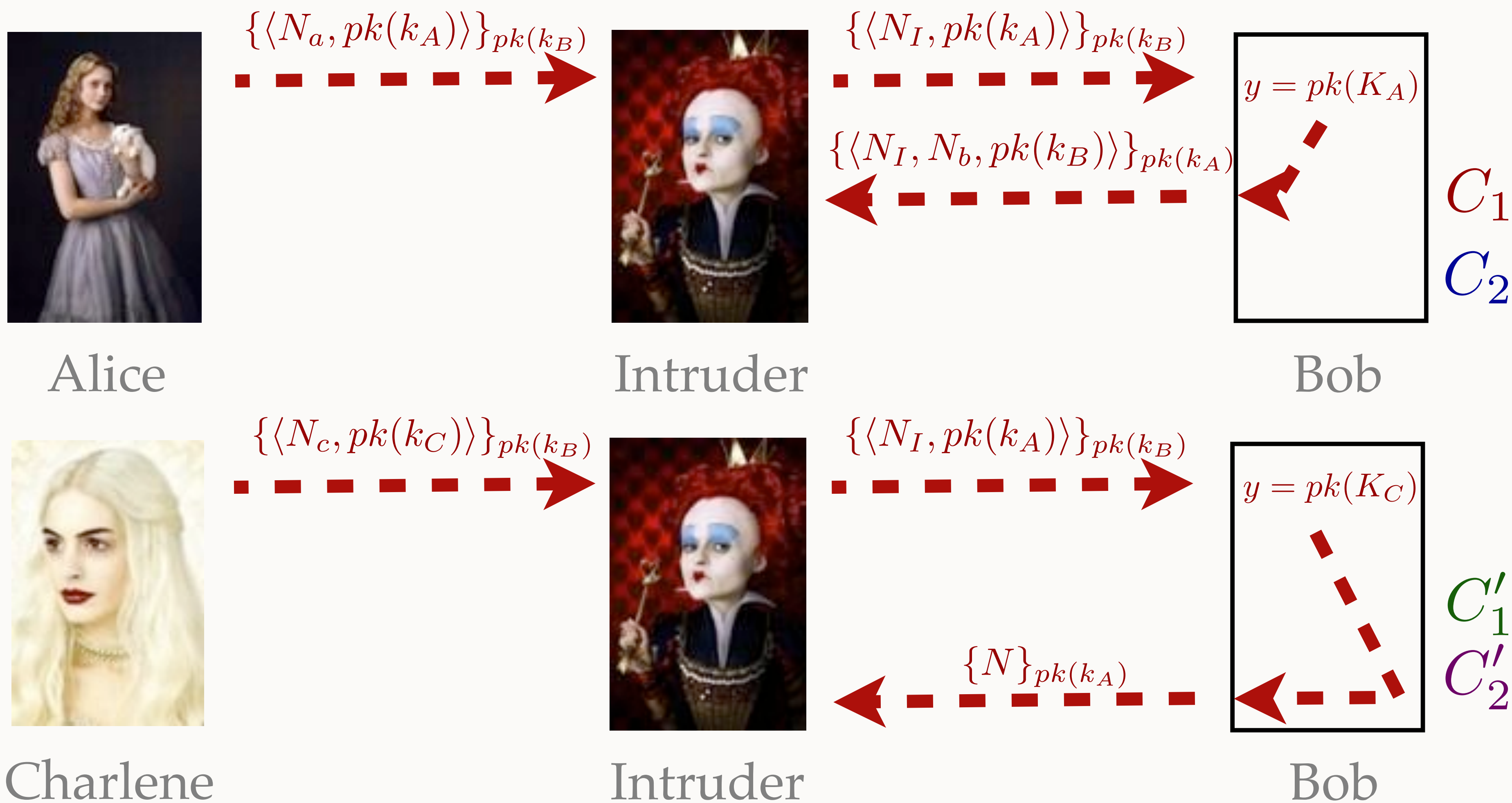
CONSTRAINT SYSTEM

- Why sets of constraint systems are necessary ?



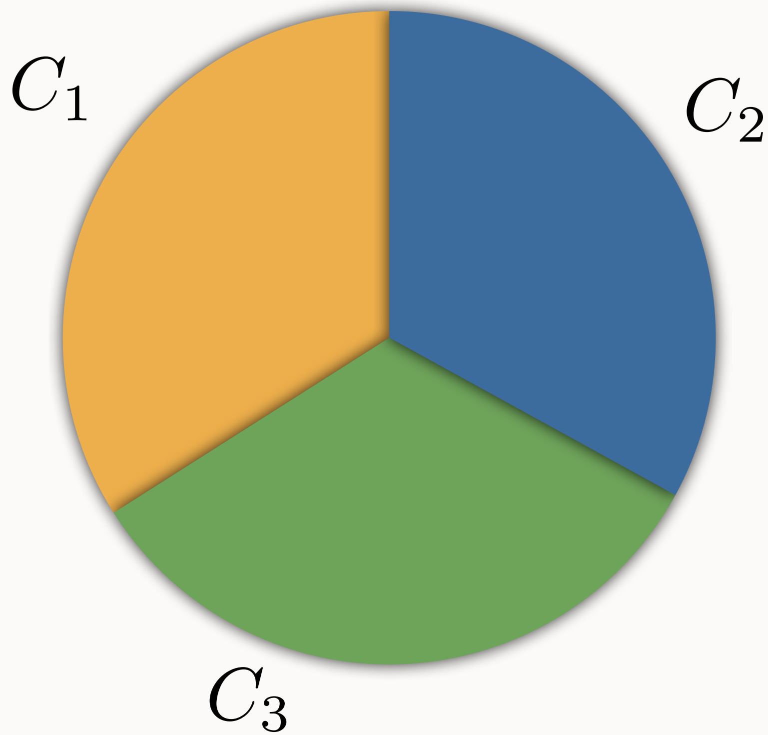
CONSTRAINT SYSTEM

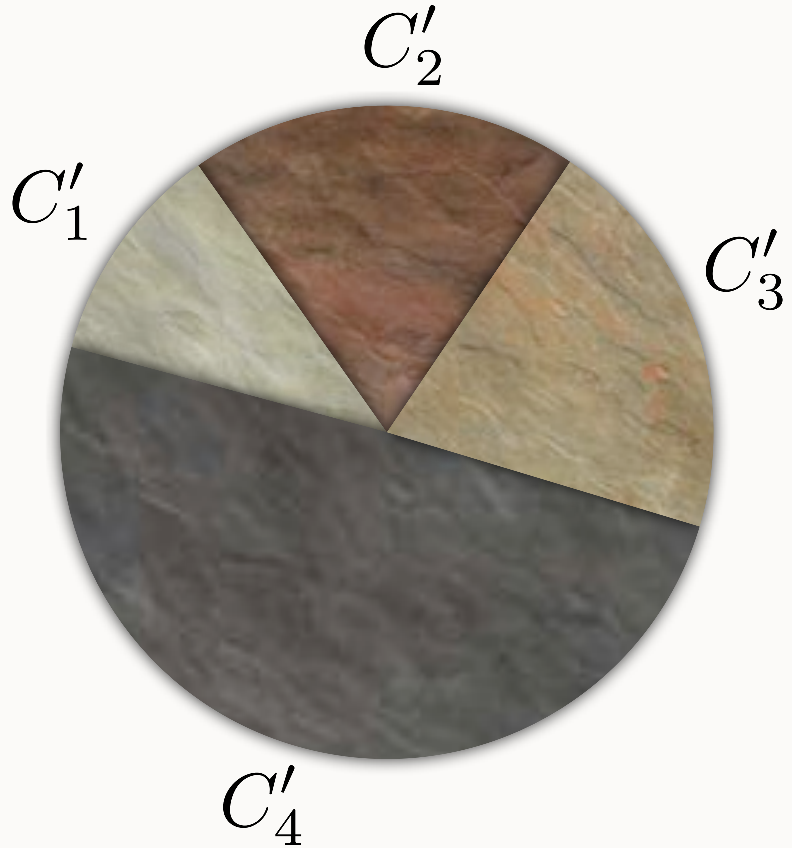
- Why sets of constraint systems are necessary ?



CONSTRAINT SYSTEM

- Why sets of constraint systems are necessary ?





$S = \{C_1; C_2; C_3\}$ $\quad$ $S' = \{C'_1; C'_2; C'_3; C'_4\}$

CONSTRAINT SYSTEM

- Symbolic equivalence between sets of constraint systems

To check whether P and P' are trace equivalent, we have to check that :

$S \approx S'$, for all symbolic interleaving

Symbolic equivalence $S \approx S$

- For all $C \in S$, for all $(\theta, \sigma) \in \mathrm{Sol}(C)$, there exists $C' \in S'$ and σ' such that $(\theta, \sigma') \in \mathrm{Sol}(C')$ and $\Phi\sigma \sim \Phi'\sigma'$
- and conversely

CONSTRAINT SYSTEM

- Previous works on constraint system

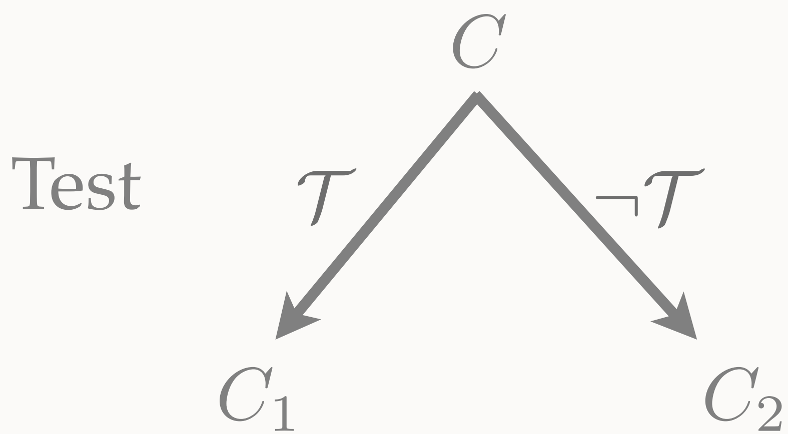
1. M. Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires.* Phd thesis
2. Y. Chevalier and M. Rusinowitch. *Decidability of equivalence of symbolic derivations.*
3. V. Cortier and S. Delaune. *A method for proving observational equivalence.*
4. A. Tiu and J. E. Dawson. *Automating open bisimulation checking for the spi calculus.*
5. V. Cheval, H. Comon-Lundh, S. Delaune. *Automating security analyss: symbolic equivalence of constraint systems*

Focus on :

- symbolic equivalence between two constraint systems (All)
- positive constraint system (no disequations) (All)
- subterm convergent equational theory (1,2 & 3)
- more restricted equational theory (4 & 5)

THE ALGORITHM

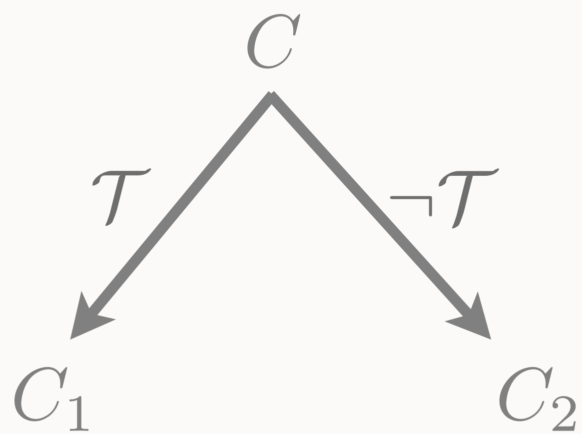
- Set of rules



THE ALGORITHM

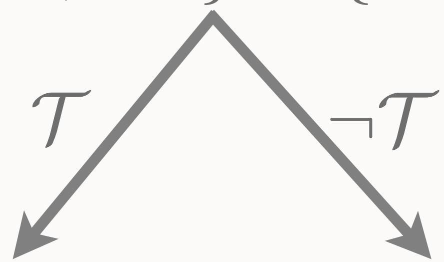
- Set of rules

Test



- How to apply the rules :

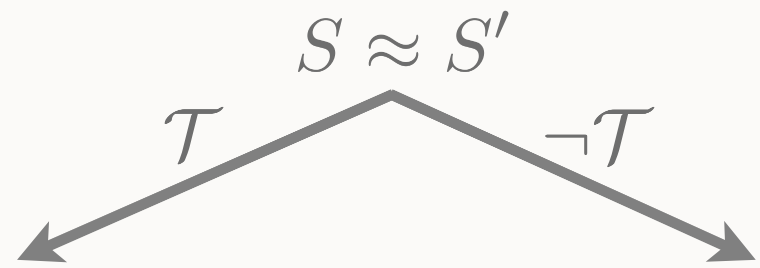
$$\{C^1;\ C^2;\ \ldots\} \approx \{C^n;\ \ldots\}$$



$$\{C_1^1;\ C_1^2;\ \ldots\} \approx \{C_1^n;\ \ldots\} \qquad \{C_2^1;\ C_2^2;\ \ldots\} \approx \{C_2^n;\ \ldots\}$$

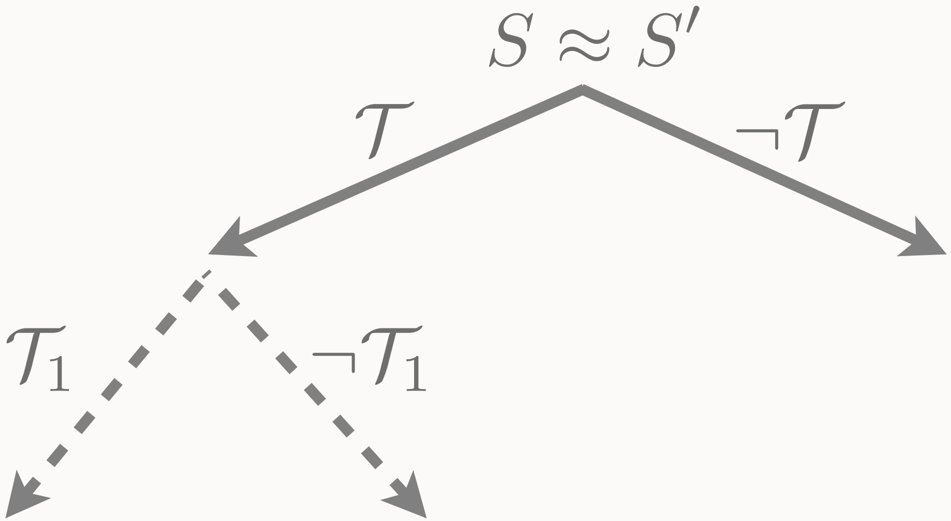
THE ALGORITHM

- A complete execution



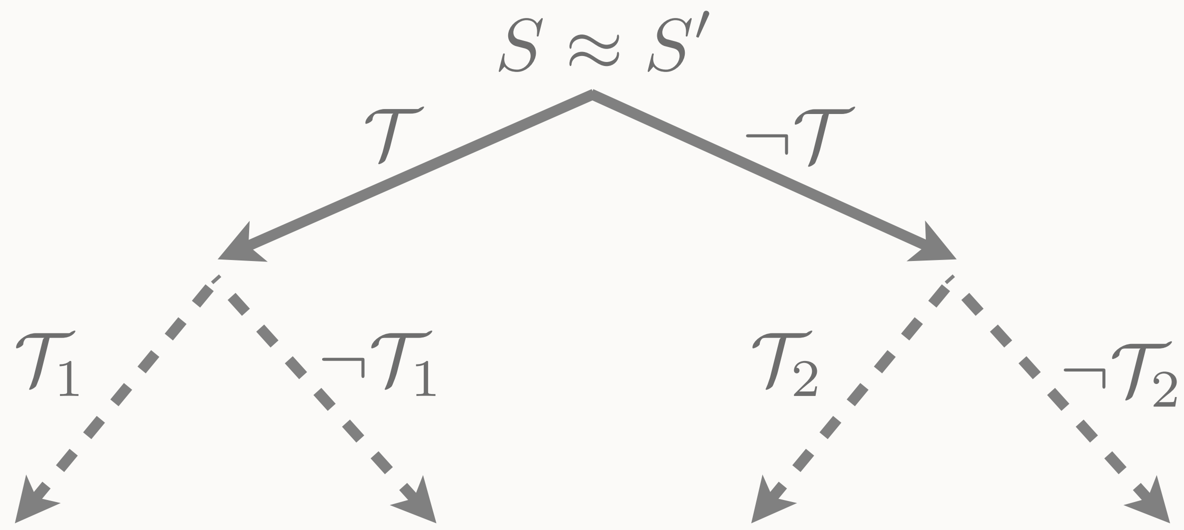
THE ALGORITHM

- A complete execution



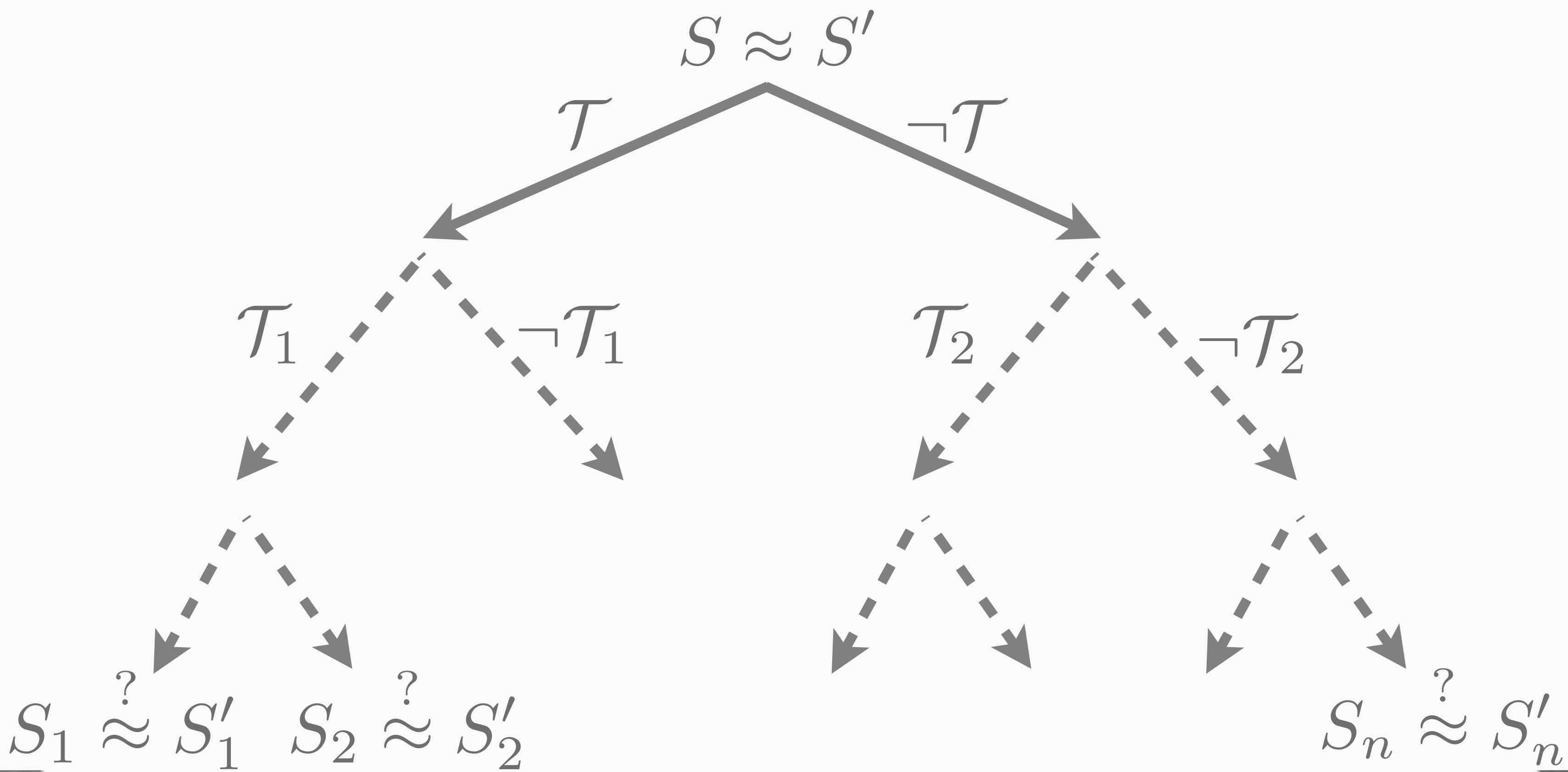
THE ALGORITHM

- A complete execution



THE ALGORITHM

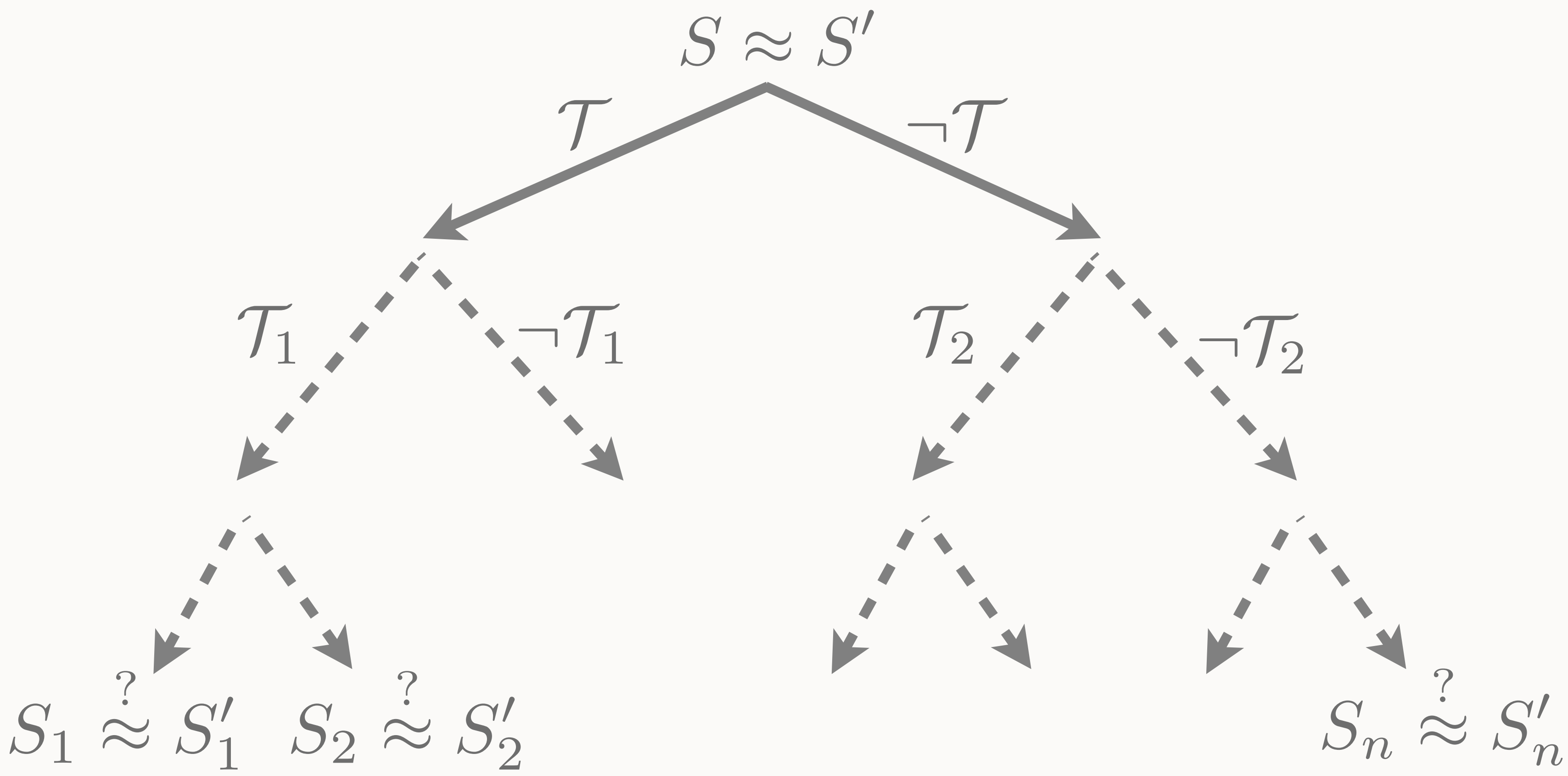
- A complete execution



The application of the rules creates a binary tree where each node is a pair of sets of constraint systems

THE ALGORITHM

- A complete execution



The symbolic equivalence is syntactically decided on each leaf

THE ALGORITHM

- Example of rule : Cons

Test $\mathcal{T} = \exists X_1, X_2 \text{ s.t. } X = \text{enc}(X_1, X_2)$

$$\begin{cases} \dots \\ T \vdash_X \text{enc}(u_1, u_2) \\ \dots \end{cases}$$

$\mathcal{T}$ $\quad$ $\neg\mathcal{T}$

$$\begin{cases} \dots \\ T \vdash_{X_1} u_1 \\ T \vdash_{X_2} u_2 \\ X = \text{enc}(X_1, X_2) \\ \dots \end{cases} \qquad \begin{cases} \dots \\ T \vdash_X \text{enc}(u_1, u_2) \\ \text{Top}(X) \neq \text{enc} \\ \dots \end{cases}$$

THE ALGORITHM

- The solved form of a constraint system
  - Existence of solutions (Reachability)

$$m_1, \ldots, m_n \vdash x$$
$$m_1, \ldots, m_n, \ldots, m_{n'} \vdash y$$

  - Matching solutions (including disequations)

$$a, b \vdash x \quad a, b, c \vdash y \quad x \neq y$$

$$a, b \vdash x \quad a, b, c \vdash y \quad x \neq f(y)$$

  - Static equivalence

$$a, \{b\}_c \vdash x \quad a, \{b\}_c, c \vdash y$$

$$a, b \vdash x \quad a, b, c \vdash y$$

RESULT

Let (S_0, S'_0) be an initial pair of set of constraint systems, we have :

(S, S')

(S, S')

RESULT

Let (S_0, S_0') be an initial pair of set of constraint systems, we have :

> If all leaves (S, S') on the tree satisfy the testing condition then $S_0 \approx S_0'$.

(S, S')

RESULT

Let (S_0, S'_0) be an initial pair of set of constraint systems, we have :

If all leaves (S, S') on the tree satisfy the testing condition then $S_0 \approx S'_0$.

If $S_0 \approx S'_0$ then all leaves (S, S') on the tree satisfy the testing condition.

RESULT

Let (S_0, S'_0) be an initial pair of set of constraint systems, we have :

If all leaves (S, S') on the tree satisfy the testing condition then $S_0 \approx S'_0$.

If $S_0 \approx S'_0$ then all leaves (S, S') on the tree satisfy the testing condition.

The strategy terminates

FUTURE WORK

- Contribution

  Decision procedure for trace equivalence

  - Infinitely many traces are represented by symbolic constraint system

  + Protocol possibly non-determinist and with non trivial else branches

  + Private channels

  - Fixed set of cryptographic primitives : symmetric and asymmetric encryption, pairing and signature

  - Bounded number of sessions (no replication in the process algebra)

- Future work

  - Experiment shows that the implementation is not efficient enough
  - More cryptographic primitives
  - Link with ProVerif

TERMINATION

- The disequations problem

$$D: \begin{array}{l} a, b \vdash x_1 \\ a, b \vdash x_2 \\ a, b \vdash y \end{array}$$

$$E: [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

$$x_2 = a$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle$$

$$x_2 = a$$

$$E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

$$\downarrow \; x_2 = a$$

$$E : [x_1 \neq y] \wedge y \neq \langle x_1, a, b \rangle$$

$$\downarrow \; y = \langle y_1, y_2, y_3 \rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

$$x_2 = a$$

$$E : [x_1 \neq y] \wedge y \neq \langle x_1, a, b \rangle$$

$$y = \langle y_1, y_2, y_3 \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle$$

$$x_2 = a$$

$$E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle$$

$$y = \langle y_1, y_2, y_3 \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

$$x_2 = a$$

$$E : [x_1 \neq y] \wedge y \neq \langle x_1, a, b \rangle$$

$$y = \langle y_1, y_2, y_3 \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge [y_1 \neq x_1 \vee y_2 \neq a \vee y_3 \neq b]\rangle$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

$$x_2 = a$$

$$E : [x_1 \neq y] \wedge y \neq \langle x_1, a, b \rangle$$

$$y = \langle y_1, y_2, y_3 \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge [y_1 \neq x_1 \vee y_2 \neq a \vee y_3 \neq b]\rangle$$

$$y_3 = b$$

TERMINATION

- The disequations problem

$$E : [x_1 \neq y \vee x_2 \neq a] \wedge y \neq \langle x_1, x_2, b \rangle$$

$$\downarrow \; x_2 = a$$

$$E : [x_1 \neq y] \wedge y \neq \langle x_1, a, b \rangle$$

$$\downarrow \; y = \langle y_1, y_2, y_3 \rangle$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle$$

$$\downarrow$$

$$E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \wedge [y_1 \neq x_1 \vee y_2 \neq a \vee y_3 \neq b]\rangle$$

$$\downarrow \; y_3 = b$$

$$E : [x_1 \neq \langle y_1, y_2, b \rangle] \wedge [y_1 \neq x_1 \vee y_2 \neq a]\rangle$$