PROVING MORE OBSERVATIONAL EQUIVALENCES WITH PROVERIF

Vincent Cheval[(1)] & Bruno Blanchet[(2)]

(1) LSV, ENS Cachan & CNRS & INRIA Saclay Ile-de-France, France

(2) INRIA Paris-Rocquencourt, France

19 March, 2013

PROVING MORE OBSERVATIONAL EQUIVALENCES WITH PROVERIF

Vincent Cheval[(1)] & Bruno Blanchet[(2)]

(1) School of Computer Science, University of Birmingham, UK

(2) INRIA Paris-Rocquencourt, France

19 March, 2013

Context

Most communications take place over a public network

It is important to ensure their security

Symbolic model

Alice

Attacker

Bob

- We assume perfect cryptographic primitives
- Messages are represented by terms

$$\{N\}_{\mathsf{pk}(k)} \qquad \langle N, M \rangle \qquad \mathsf{adec}(\{N\}_{\mathsf{pk}(k)}, k)$$

Symbolic model

Alice

Attacker

Bob

- We assume perfect cryptographic primitives
- Messages are represented by terms

$$\{N\}_{\mathsf{pk}(k)} \qquad \langle N, M \rangle \qquad \mathsf{adec}(\{N\}_{\mathsf{pk}(k)}, k)$$

The attacker can

- intercept all messages
- transmit or modify messages
- test equality between messages

Symbolic model

Alice

Attacker

Bob

- We assume perfect cryptographic primitives
- Messages are represented by terms

$$\{N\}_{\mathsf{pk}(k)} \qquad \langle N, M \rangle \qquad \mathsf{adec}(\{N\}_{\mathsf{pk}(k)}, k)$$

The attacker can

- intercept all messages
- transmit or modify messages
- test equality between messages

Symbolic model

Alice

Attacker

Bob

- We assume perfect cryptographic primitives
- Messages are represented by terms

$$\{N\}_{\mathsf{pk}(k)} \qquad \langle N, M \rangle \qquad \mathsf{adec}(\{N\}_{\mathsf{pk}(k)}, k)$$

The attacker can

- intercept all messages
- transmit or modify messages
- test equality between messages

Symbolic model

Alice

Attacker

Bob

- We assume perfect cryptographic primitives
- Messages are represented by terms

$$\{N\}_{\mathsf{pk}(k)} \qquad \langle N, M \rangle \qquad \mathsf{adec}(\{N\}_{\mathsf{pk}(k)}, k)$$

The attacker can

- intercept all messages
- transmit or modify messages
- test equality between messages

Symbolic model

Alice

Attacker

Bob

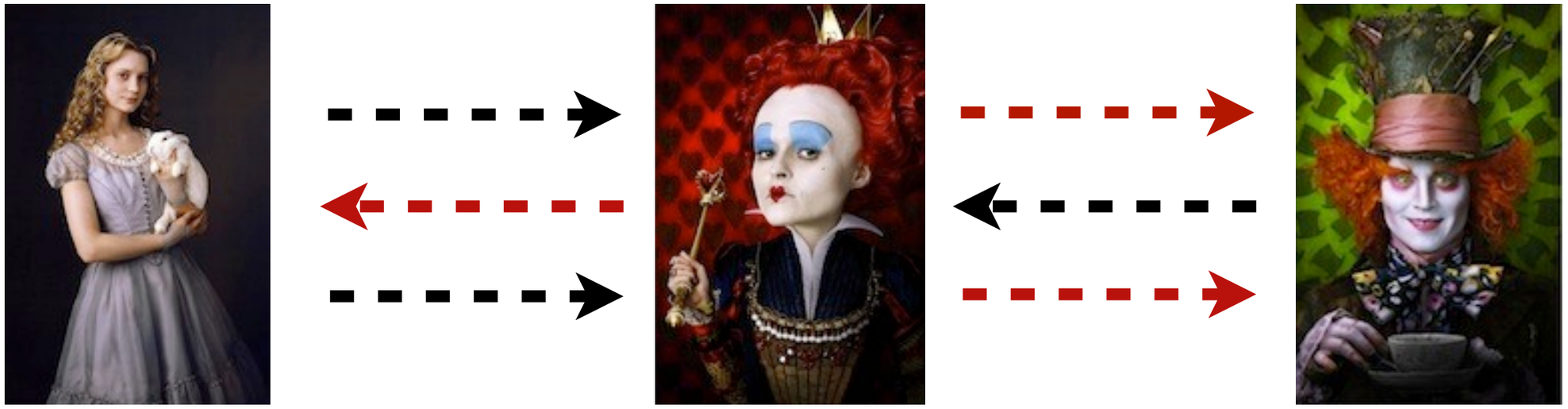
- We assume perfect cryptographic primitives
- Messages are represented by terms

$$\{N\}_{\mathsf{pk}(k)} \qquad \langle N, M \rangle \qquad \mathsf{adec}(\{N\}_{\mathsf{pk}(k)}, k)$$

The attacker can

- intercept all messages
- transmit or modify messages
- test equality between messages

Symbolic model

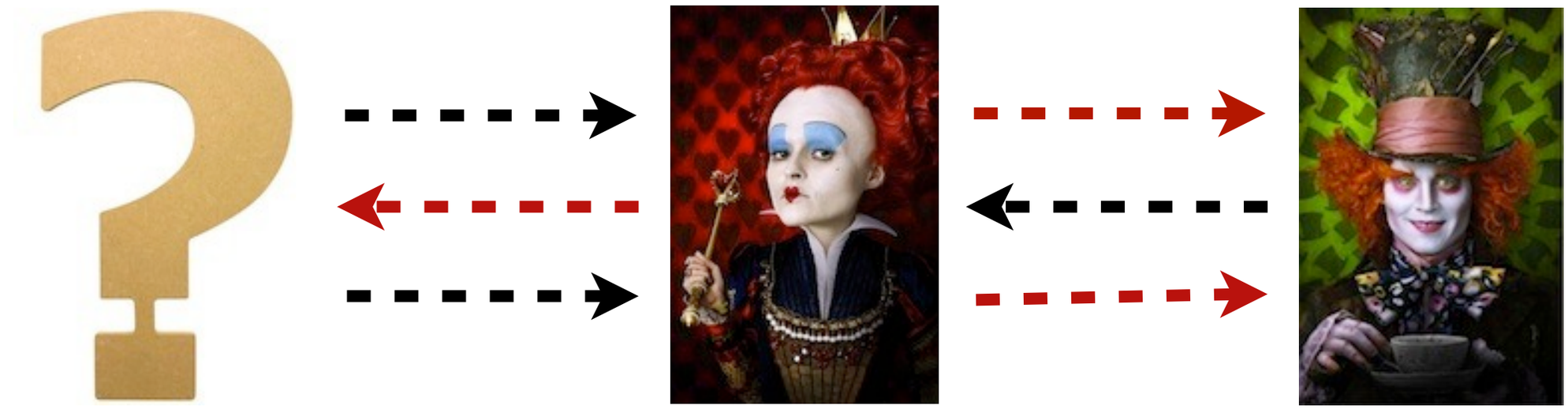


Security properties:

- Reachability properties
- Equivalence properties

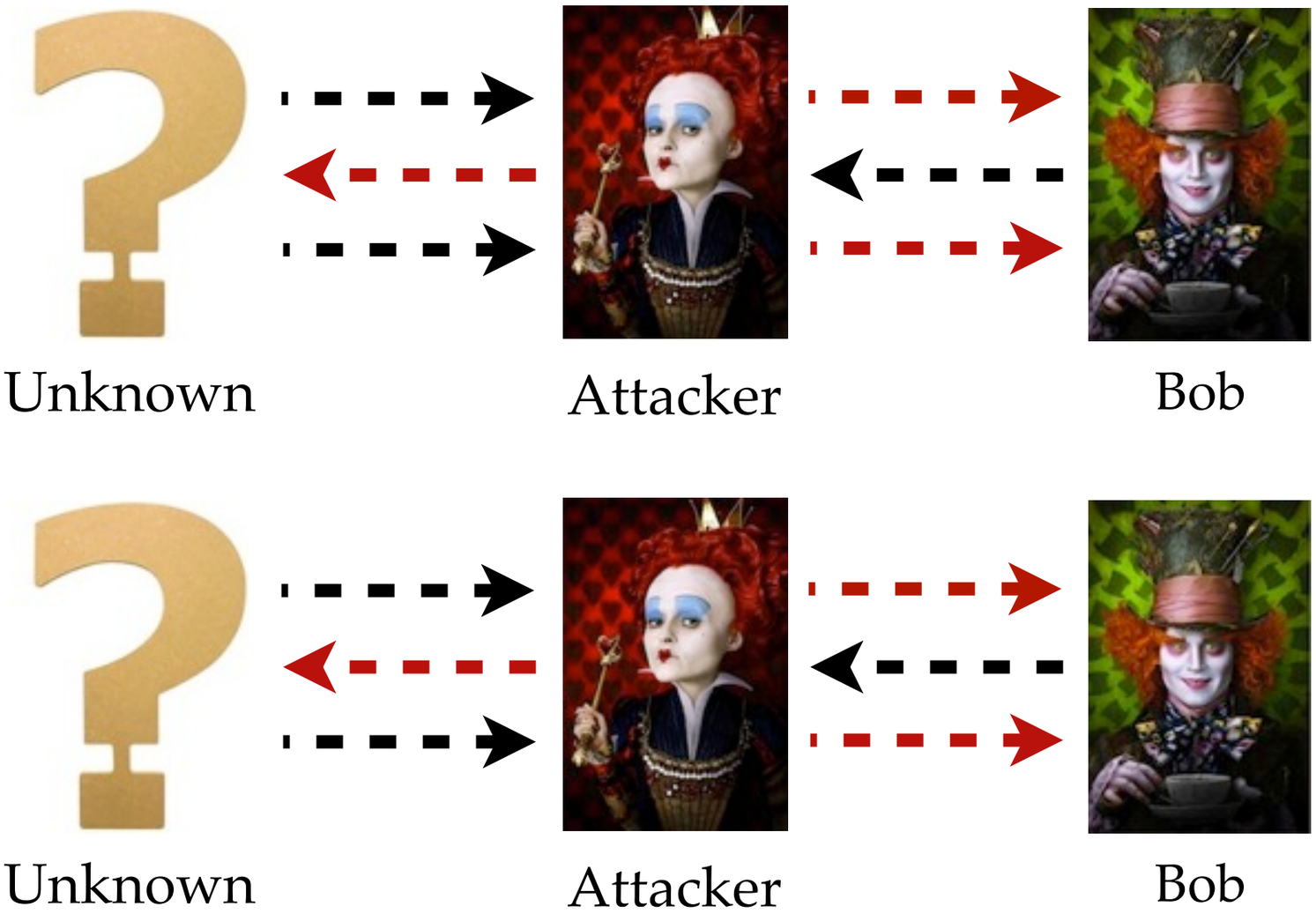
Security properties

Equivalence properties: anonymity



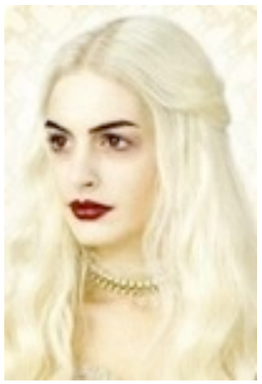
Security properties

Equivalence properties: anonymity



Security properties

Equivalence properties: anonymity



Charlene

Unknown

Attacker

Bob

Alice

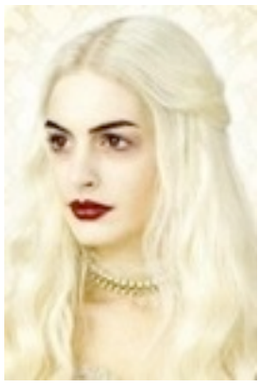
Unknown

Attacker

Bob

Security properties

Equivalence properties: anonymity


Charlene


Unknown


Attacker


Bob


Alice


Unknown

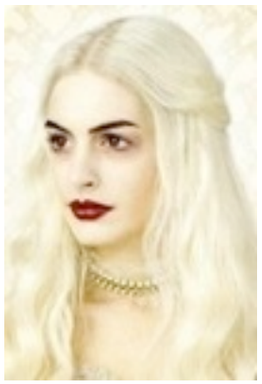

Attacker


Bob

Can the intruder distinguish the two situations ?

Security properties

Equivalence properties: anonymity



Charlene

Unknown

Attacker

Bob

Alice

Unknown

Attacker

Bob

Observational equivalence

Proverif

ProVerif was first an analyzer for reachability properties based on Horn clauses.

- Handle reachability and equivalence properties
- Cryptographic primitives described by equational theory and/or rewriting rules
- Handle processes with replication
- Possible false attack
- Does not always terminate

Examples

Private authentication protocol

Examples

Private authentication protocol

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

Alice

Bob

Examples

Private authentication protocol

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

$\mathsf{pk}(k_A)?$

Alice

Bob

Examples

Private authentication protocol

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

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

$\mathsf{pk}(k_A)?$

Alice

Bob

Examples

Private authentication protocol

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

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

$$\{N\}_{\mathsf{pk}(k_A)}$$

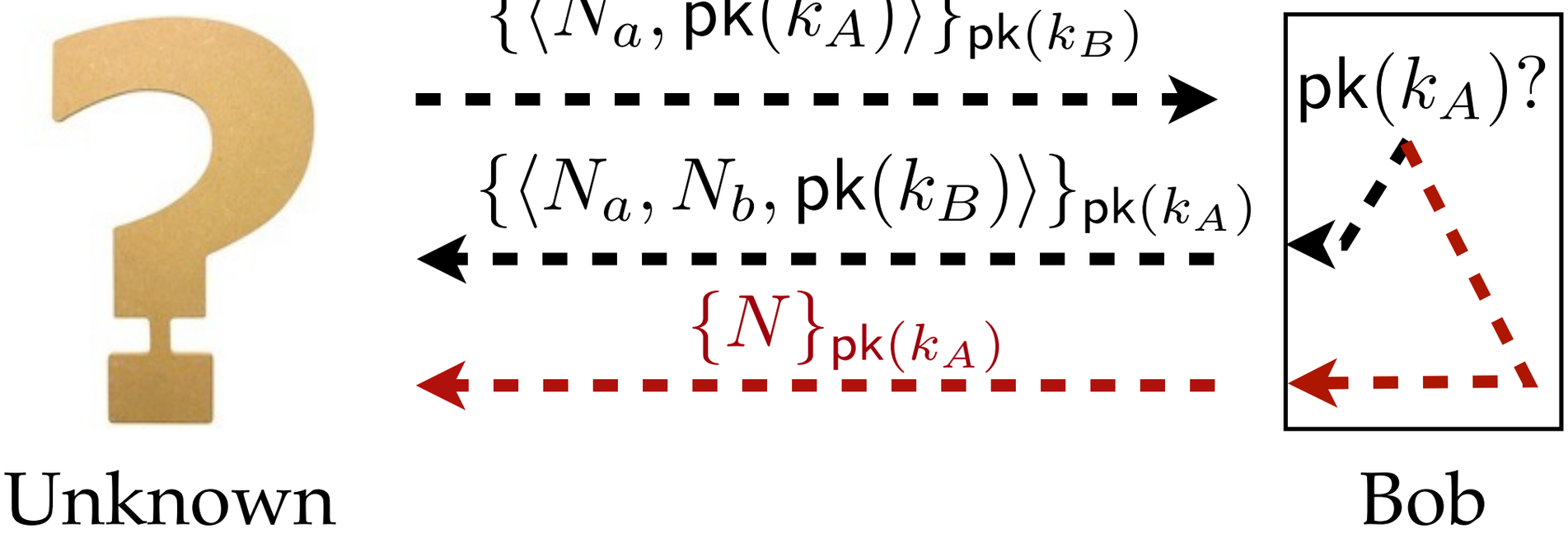
$\mathsf{pk}(k_A)?$

Alice

Bob

Examples

Private authentication protocol

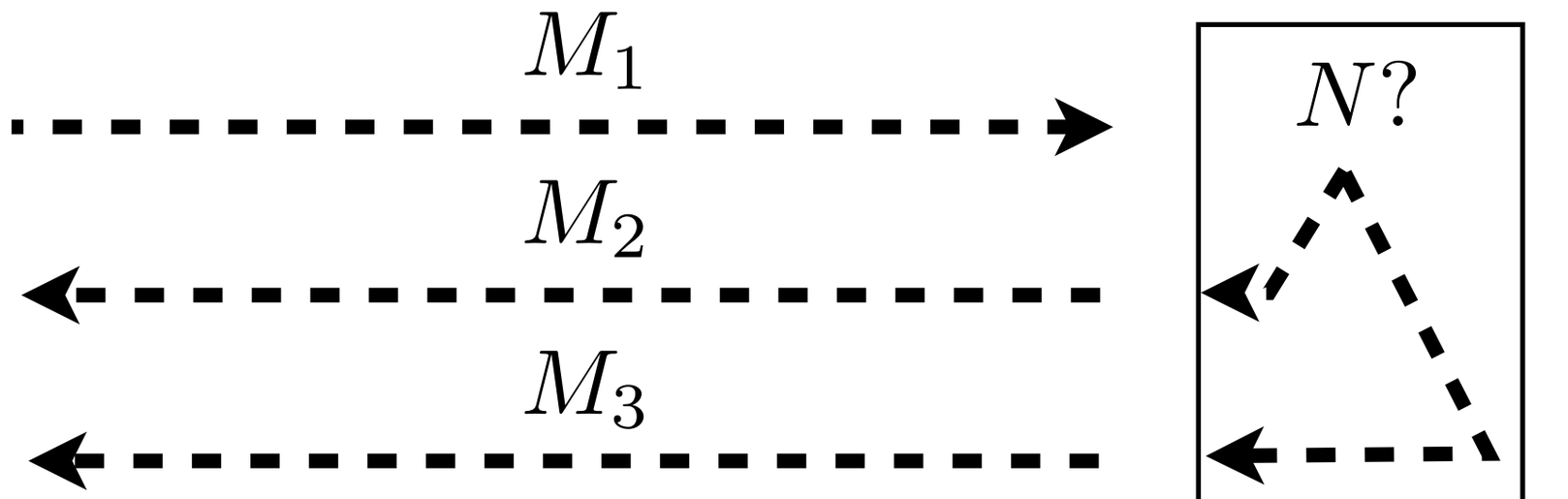


Processes

$$
\begin{aligned}
P, Q := \quad & 0 \\
& \mathsf{in}(c, x); P \\
& \mathsf{out}(c, M); P \\
& P \mid Q \\
& !\, P \\
& \mathsf{new}\ a; P \\
& \mathsf{let}\ x = D\ \mathsf{in}\ P\ \mathsf{else}\ Q
\end{aligned}
$$

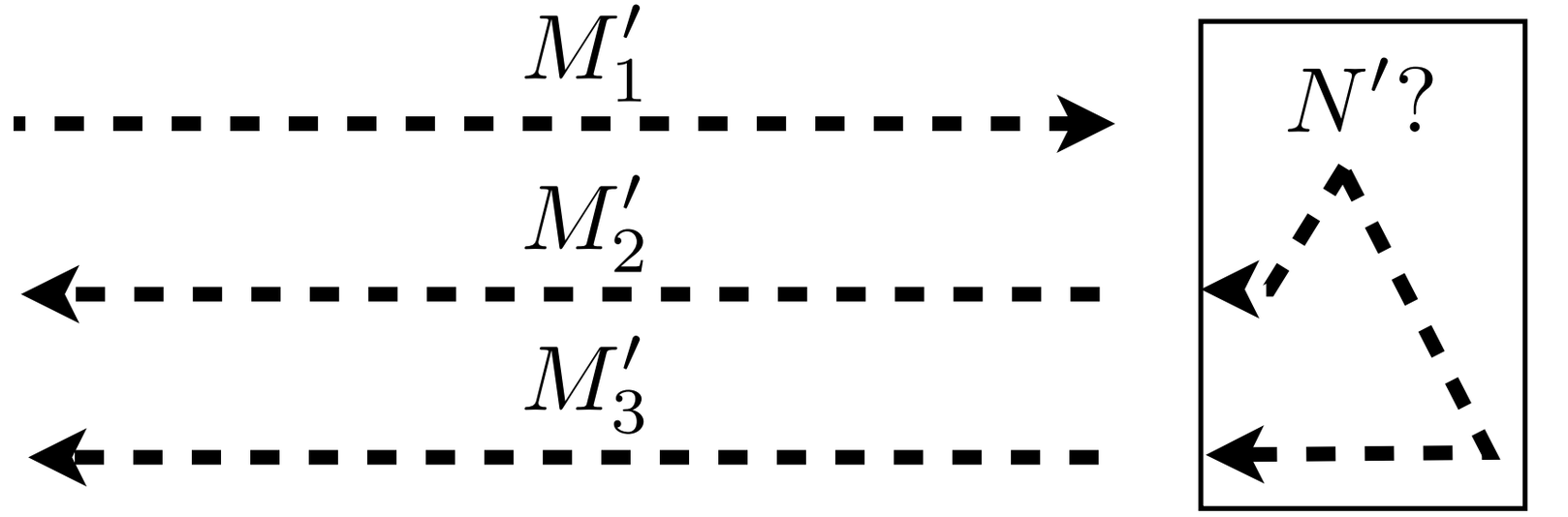
Biprocesses

Alice



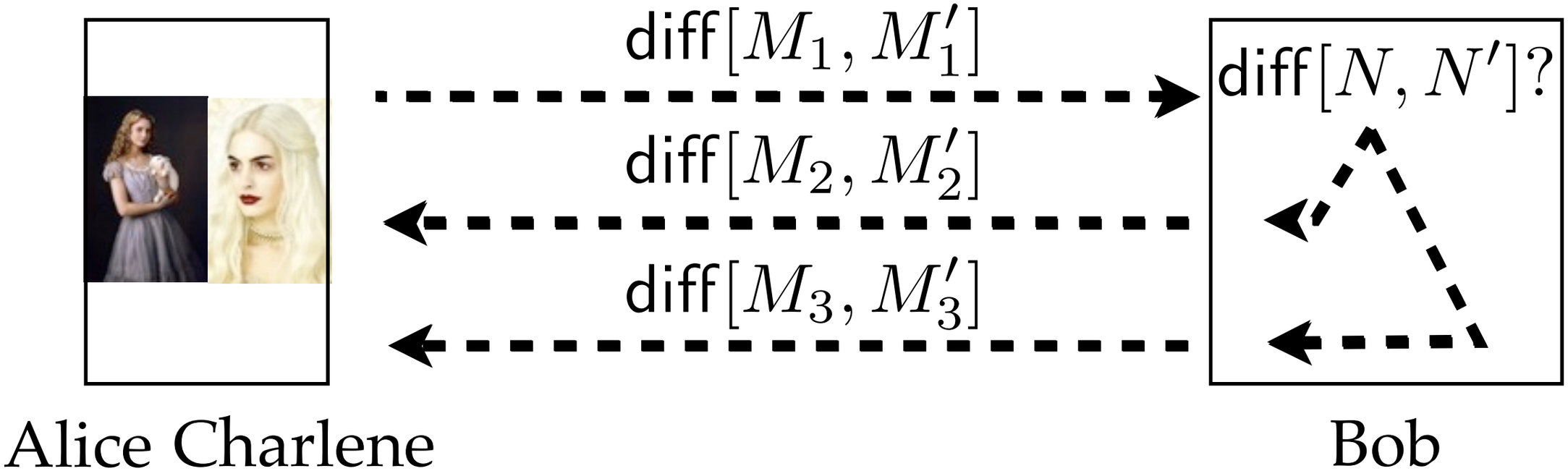
Bob

Charlene

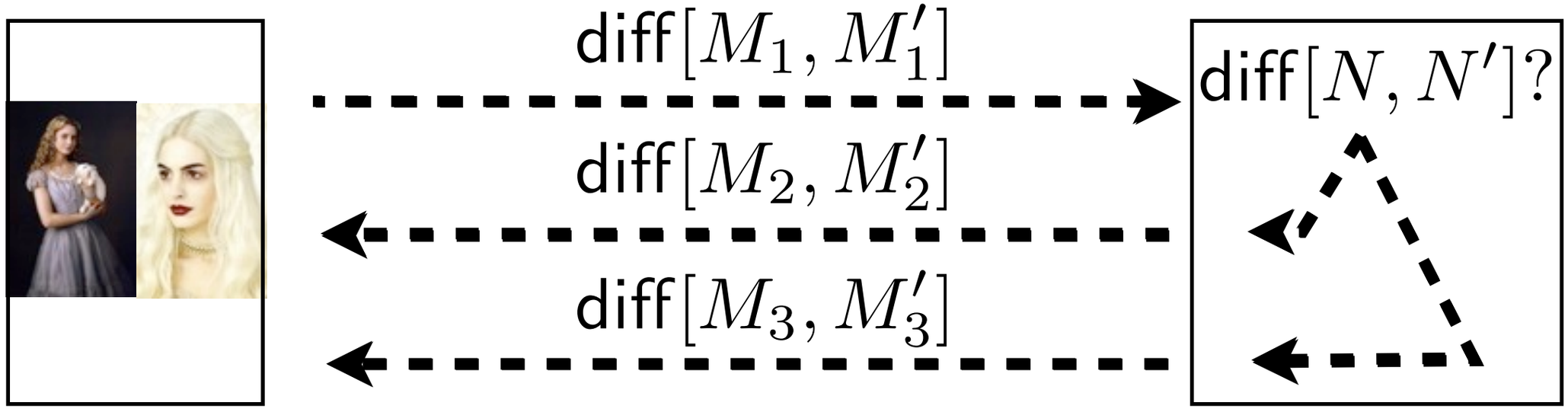


Bob

Biprocesses



Biprocesses



Equivalence too strong: possible false attack

Motivation

The private authentication protocol

Alice

Attacker

Bob

Charlene

Attacker

Bob

Motivation

The private authentication protocol

Alice

Bob

Charlene

Bob

Motivation

The private authentication protocol

Alice

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

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

Bob

Charlene

Bob

Motivation

The private authentication protocol

Alice

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

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

$\mathsf{pk}(k_A) = y$

Bob

Charlene

Bob

Motivation

The private authentication protocol

Alice

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

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

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

$\mathsf{pk}(k_A) = y$

Bob

Charlene

Bob

Motivation

The private authentication protocol

Alice

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

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

$\mathsf{pk}(k_A) = y$

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

$\{N\}_{\mathsf{pk}(k_A)}$

Bob

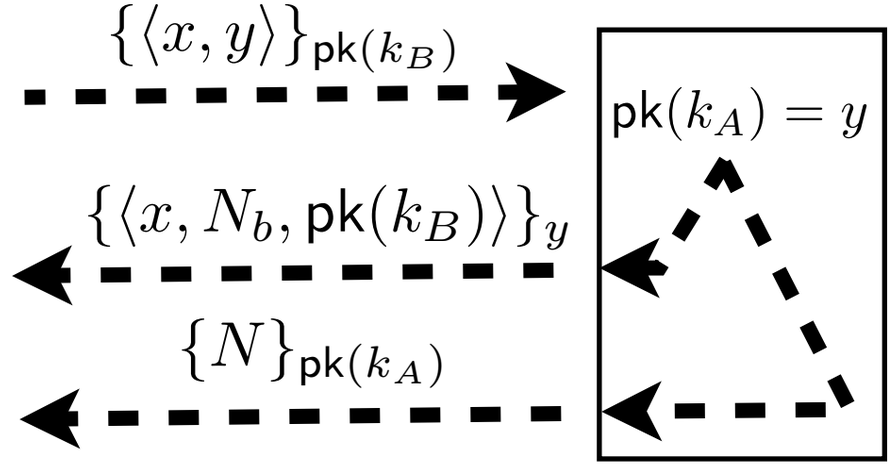
Charlene

Bob

Motivation

The private authentication protocol

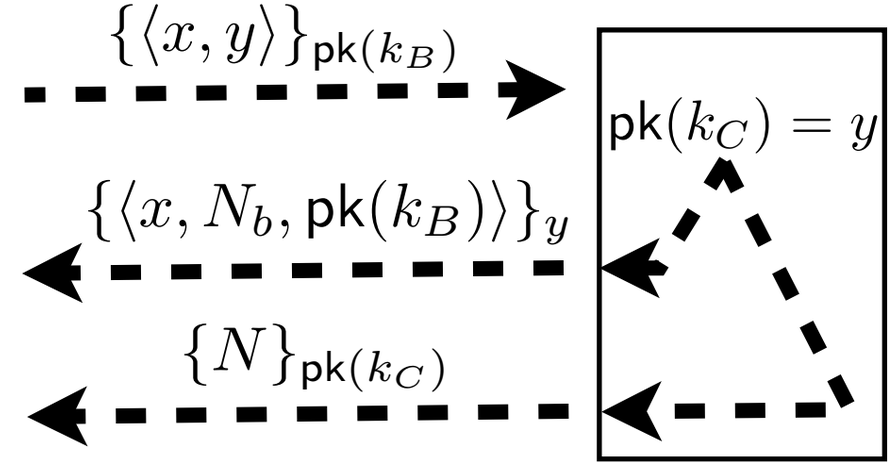
$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$



Alice

Bob

$\{\langle N_c, \mathsf{pk}(k_C)\rangle\}_{\mathsf{pk}(k_B)}$

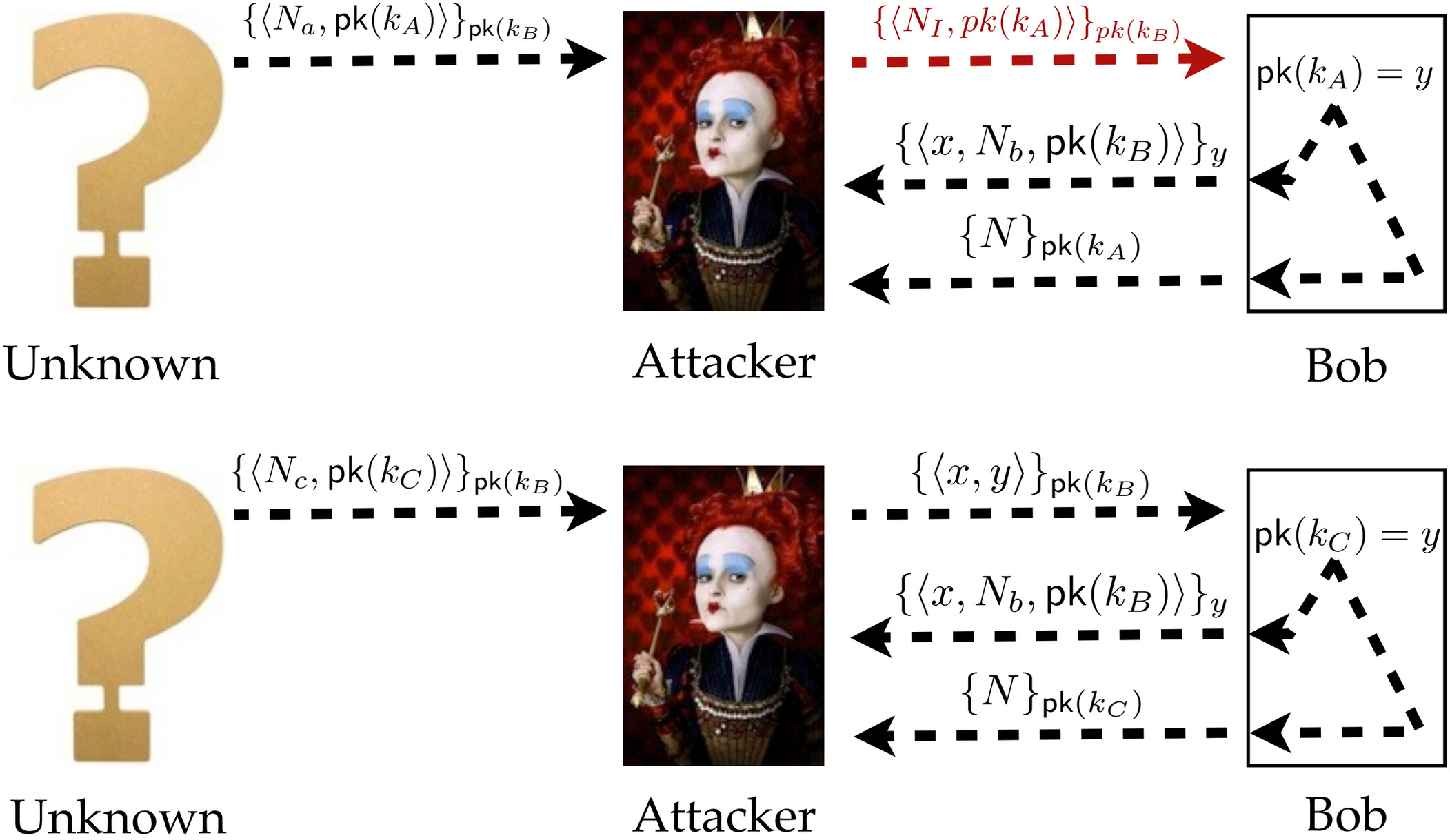


Charlene

Bob

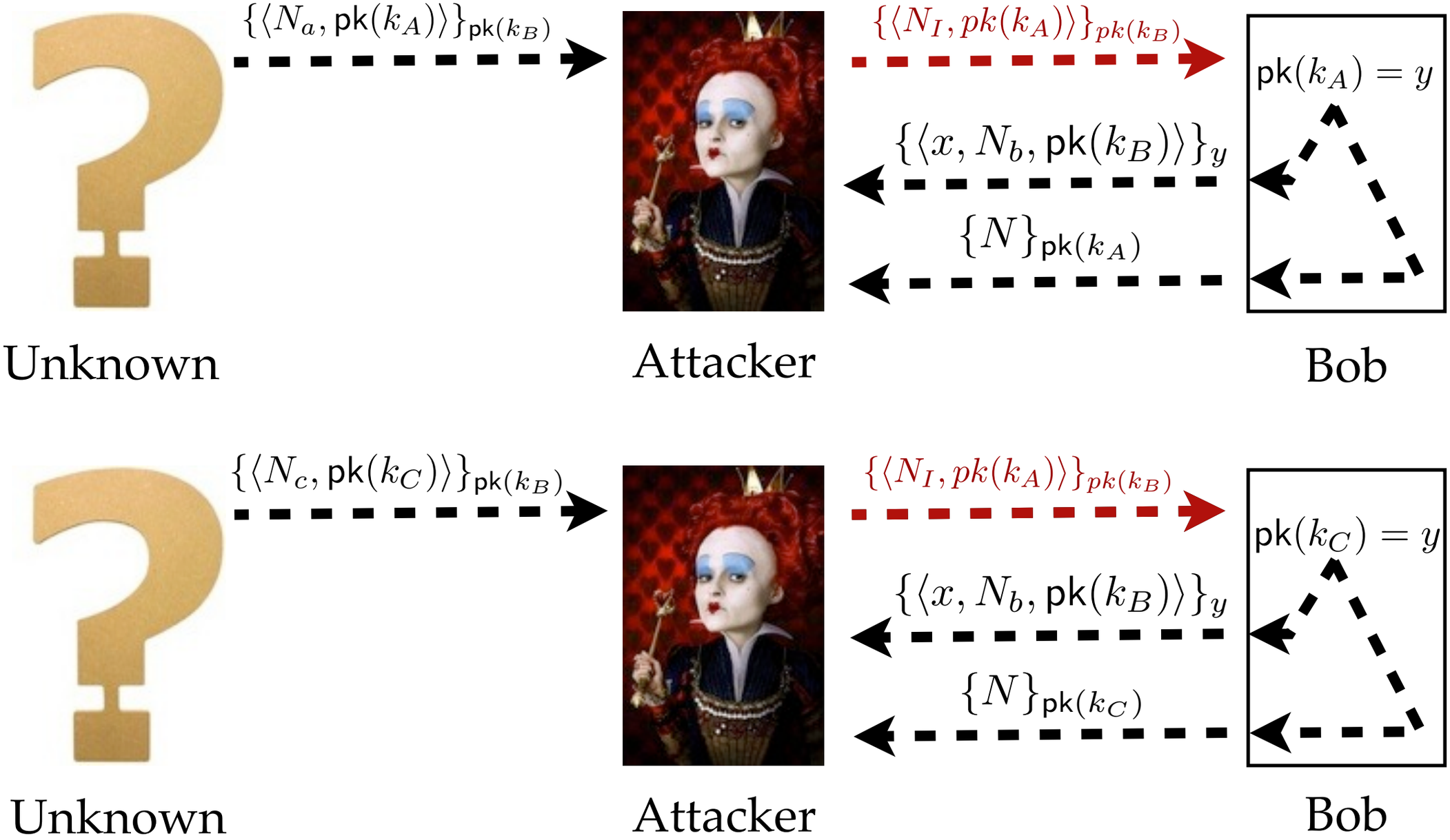
Motivation

The private authentication protocol



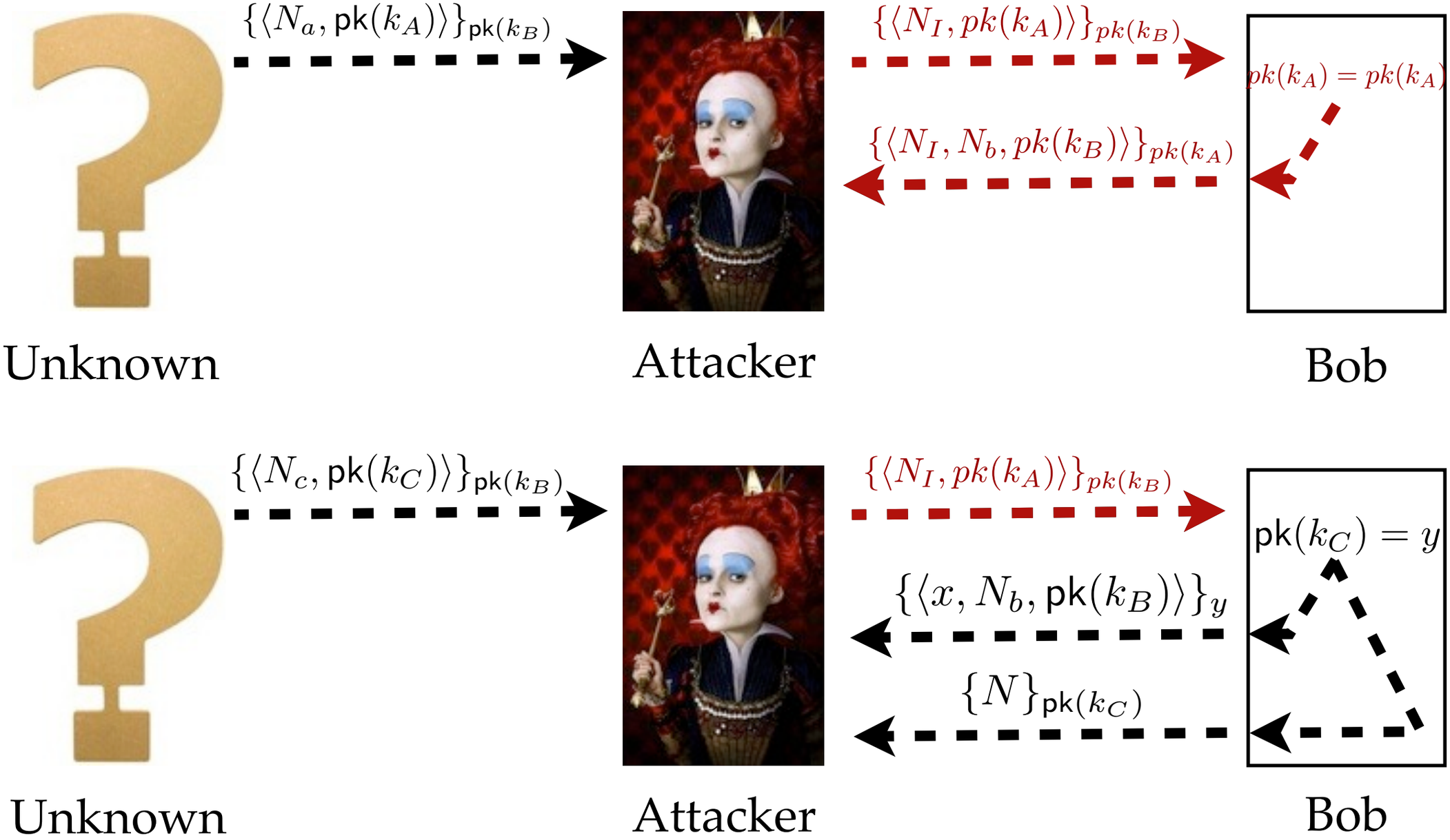
Motivation

The private authentication protocol



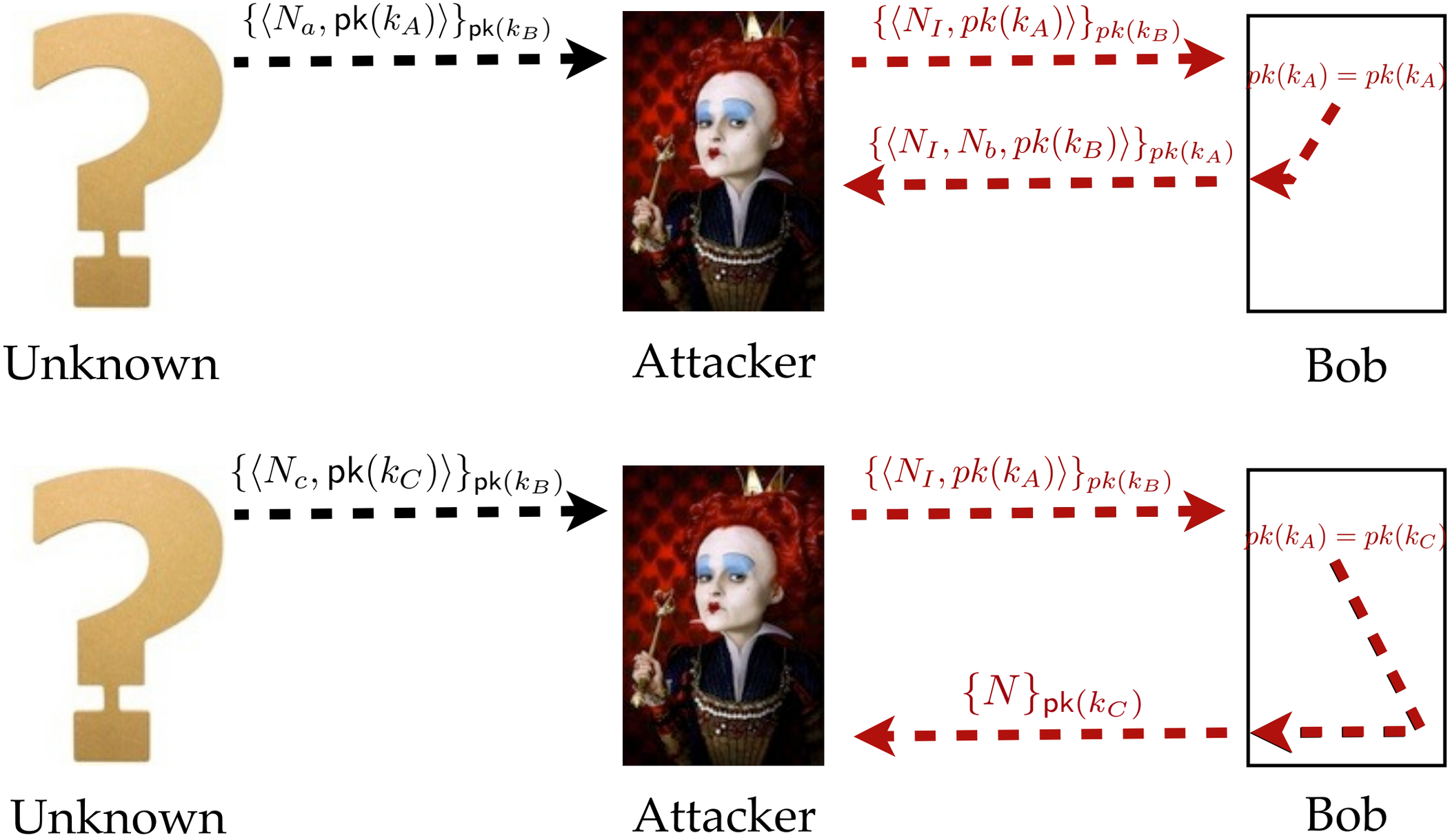
Motivation

The private authentication protocol



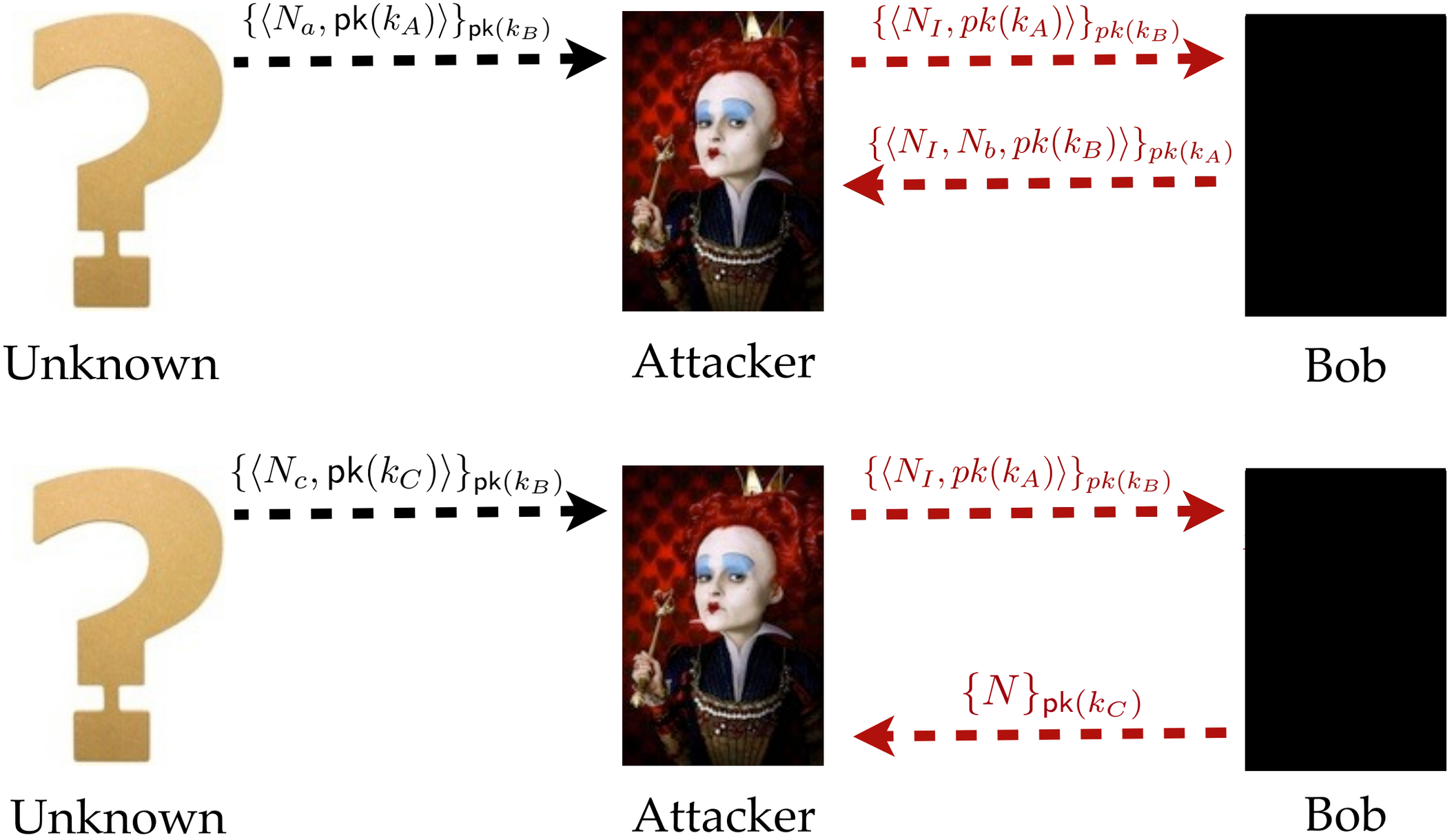
Motivation

The private authentication protocol



Motivation

The private authentication protocol



Contribution

Introduction of destructors with tests between terms

Contribution

Introduction of destructors with tests between terms

The *If-then-else* destructor:

$$\mathsf{ifthenelse}(x, x, z, t) \to z$$

$$\mathsf{ifthenelse}(x, y, z, t) \to t \quad \text{with} \quad x \neq y$$

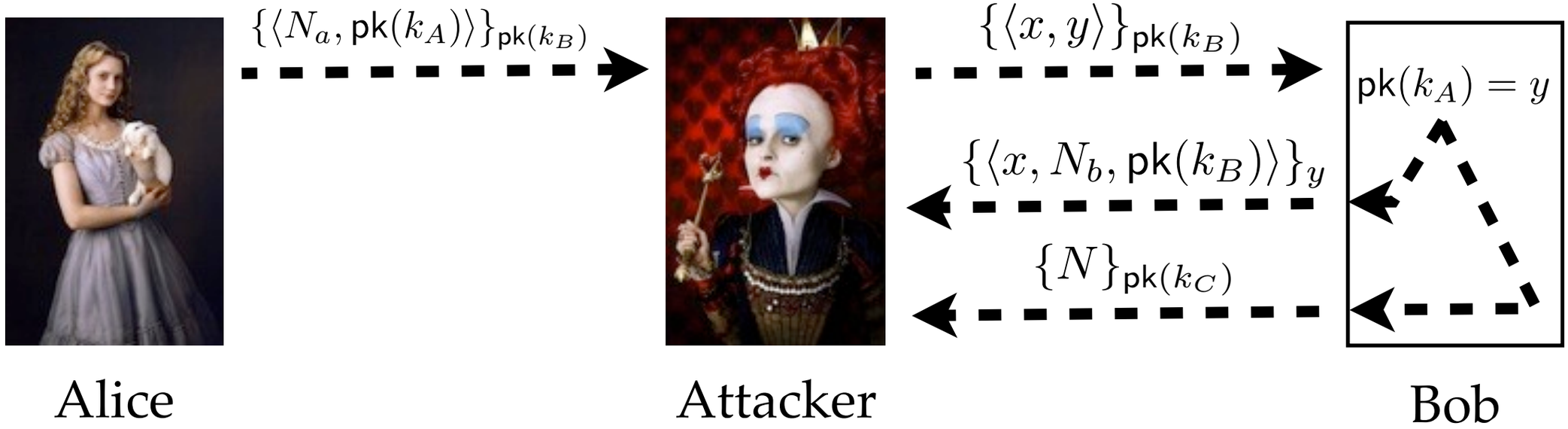
Contribution

Introduction of destructors with tests between terms

The *If-then-else* destructor:

$$\mathsf{ifthenelse}(x, x, z, t) \to z$$

$$\mathsf{ifthenelse}(x, y, z, t) \to t \quad \text{with} \quad x \neq y$$



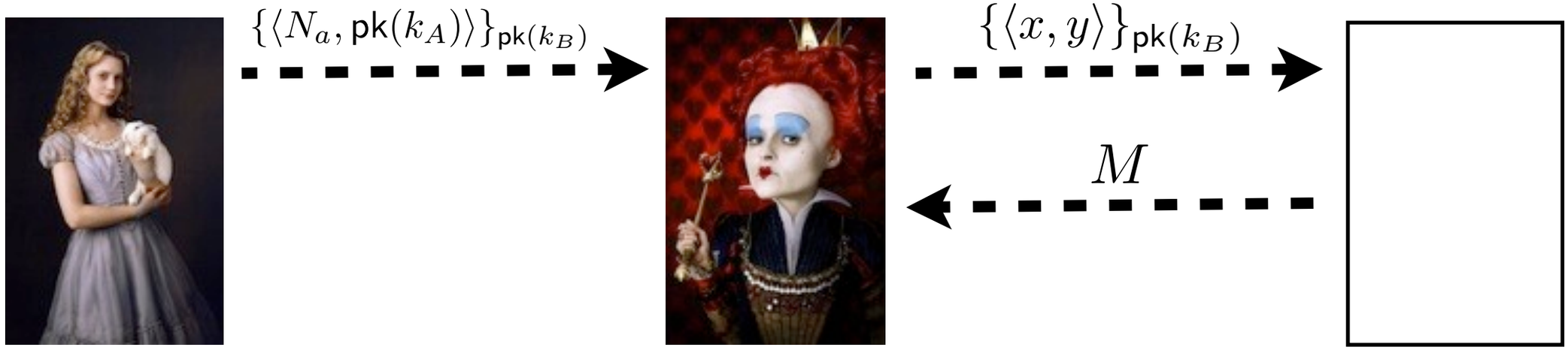
Contribution

Introduction of destructors with tests between terms

The *If-then-else* destructor:

$$\mathsf{ifthenelse}(x, x, z, t) \to z$$

$$\mathsf{ifthenelse}(x, y, z, t) \to t \quad \text{with} \quad x \neq y$$



$$M = \mathsf{ifthenelse}(y, \mathsf{pk}(k_A), \{x, N_b, \mathsf{pk}(k_B)\}_y, \{N\}_{\mathsf{pk}(k_A)})$$

Contribution

Introduction of destructors with tests between terms

The *If-then-else* destructor:

$$\mathsf{ifthenelse}(x, x, z, t) \to z$$

$$\mathsf{ifthenelse}(x, y, z, t) \to t \quad \text{with} \quad x \neq y$$

Automatic transformation: $simpl$

For all processes P, $simpl(P) \approx P$

Implementation

Beta release:

ProVerif version 1.87beta6

http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

Content:

- Rewrite rules with tests
- Automatic transformation of biprocesses
- Equivalence between processes with different control structures

Results:

- Prove anonymity for private authentication protocol (unbounded number of sessions)