

Proving more observational equivalences with ProVerif (Work in progress)

B.Blanchet & V.Cheval

23 February 2012

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

- Some security properties

CONTEXT

- Some security properties



Reachability properties

- Secrecy, Authentication, ...

CONTEXT

- Some security properties

Reachability properties

- Secrecy, Authentication, ...

Equivalence properties

- Anonymity, Privacy, Receipt-Freeness, ...

CONTEXT

■ Formalism



Alice



Intruder



Bob

The intruder can

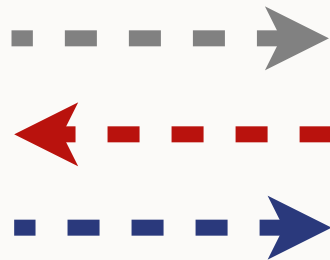
- intercept all messages
- transmit or modify messages
- test equality between messages
- initiate several sessions

CONTEXT

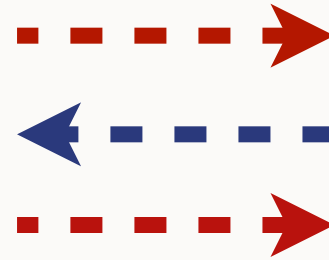
■ Formalism



Alice



Intruder



Bob

The intruder can

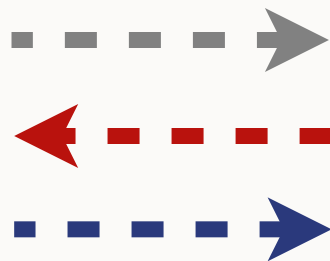
- intercept all messages
- transmit or modify messages
- test equality between messages
- initiate several sessions

CONTEXT

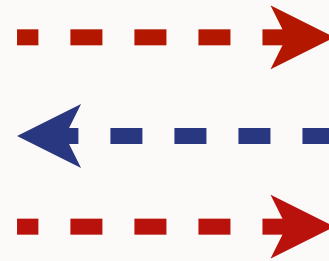
- Equivalence properties : strong secret, anonymity,...



Alice



Intruder



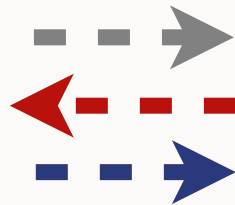
Unknown

CONTEXT

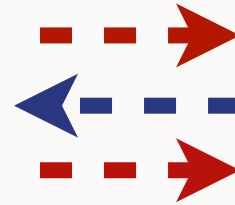
- Equivalence properties : strong secret, anonymity,...



Alice



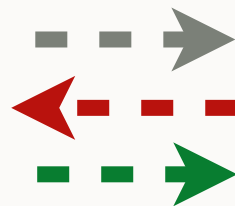
Intruder



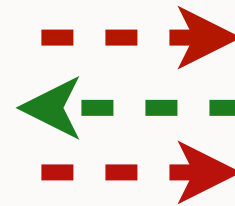
Unknown



Alice



Intruder



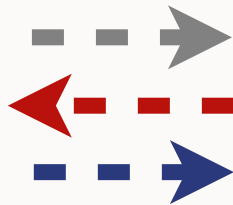
Unknown

CONTEXT

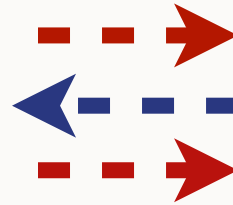
- Equivalence properties : strong secret, anonymity,...



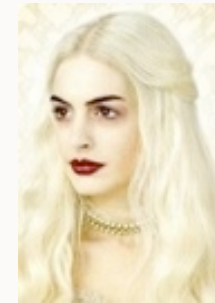
Alice



Intruder



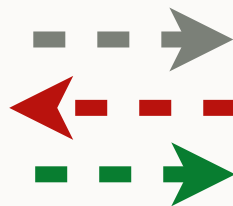
Unknown



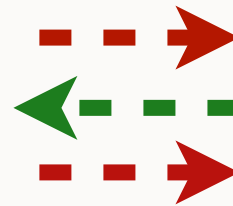
Charlene



Alice



Intruder



Unknown



Bob

PREVIOUS WORKS

Observational 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

- V. Cheval, H. Comon-Lundh, and S. Delaune. *Trace equivalence decision: Negative tests and non-determinism.*

➔ Tool : *Datep*, In development

MOTIVATION

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

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.*

ProVerif

Datep

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions

Datep

- Bounded number of sessions

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions
- Any cryptographic primitives

Datep

- Bounded number of sessions
- Fixed set of cryptographic primitives

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions
- Any cryptographic primitives
- Sound

Datep

- Bounded number of sessions
- Fixed set of cryptographic primitives
- Sound

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions
- Any cryptographic primitives
- Sound
- Possible false attack

Datep

- Bounded number of sessions
- Fixed set of cryptographic primitives
- Sound
- Complete

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions
- Any cryptographic primitives
- Sound
- Possible false attack
- Does not always terminate

Datep

- Bounded number of sessions
- Fixed set of cryptographic primitives
- Sound
- Complete
- Always terminate

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions
- Any cryptographic primitives
- Sound
- Possible false attack
- Does not always terminate

Useful to prove

Datep

- Bounded number of sessions
- Fixed set of cryptographic primitives
- Sound
- Complete
- Always terminate

MOTIVATION

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.*

ProVerif

- Unbounded number of sessions
- Any cryptographic primitives
- Sound
- Possible false attack
- Does not always terminate

Useful to prove

Datep

- Bounded number of sessions
- Fixed set of cryptographic primitives
- Sound
- Complete
- Always terminate

Useful to find attacks

MOTIVATION

- Example : Private authentication protocol


MOTIVATION

- Example : Private authentication protocol



Alice

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

A dashed arrow pointing from Alice to Bob, indicating the transmission of the encrypted data.

Bob

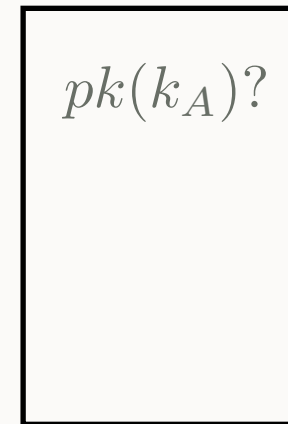
MOTIVATION

- Example : Private authentication protocol



Alice

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



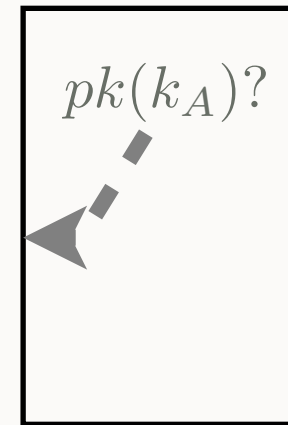
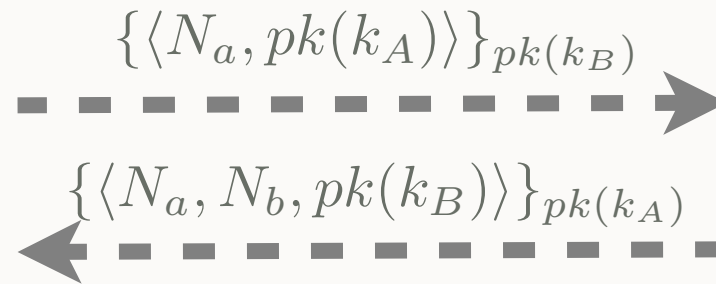
Bob

MOTIVATION

- Example : Private authentication protocol



Alice



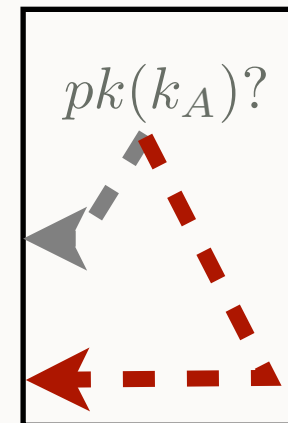
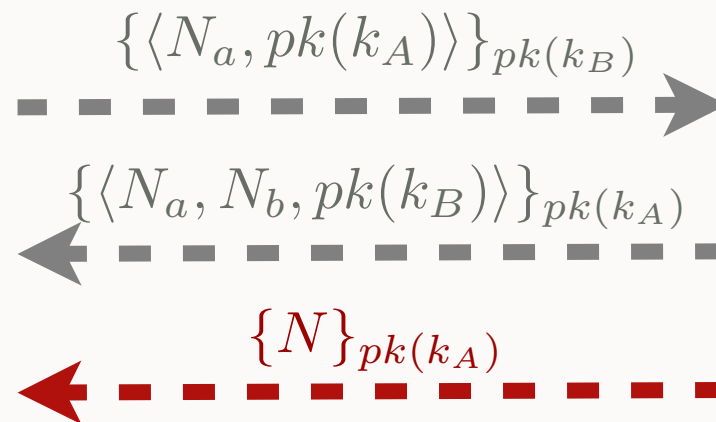
Bob

MOTIVATION

- Example : Private authentication protocol



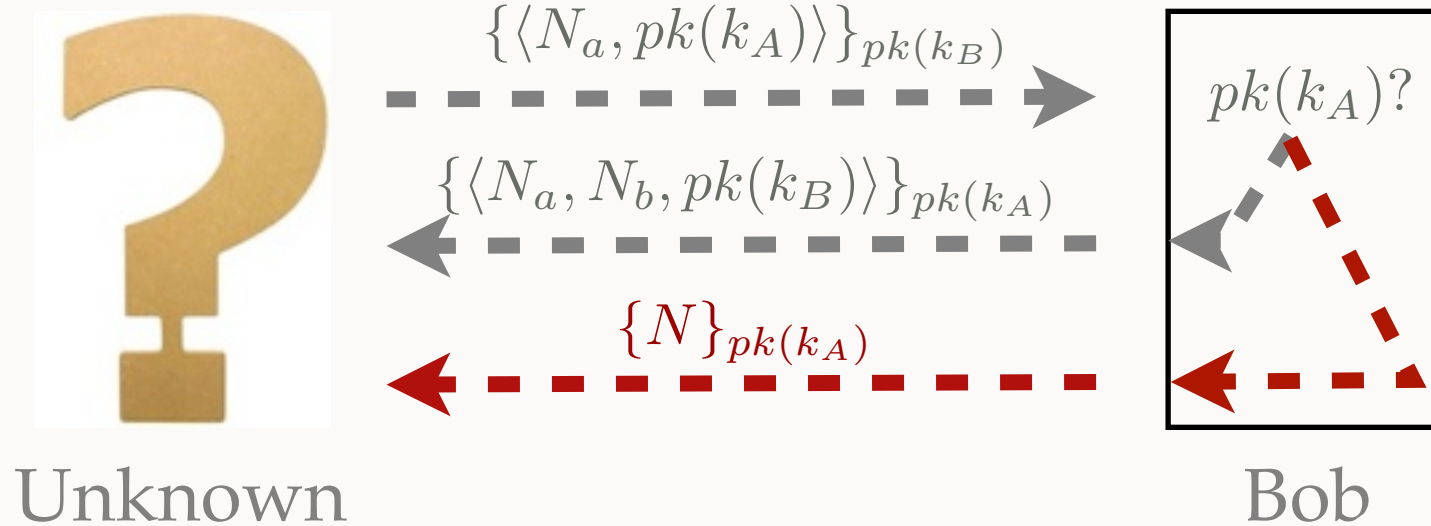
Alice



Bob

MOTIVATION

- Example : Private authentication protocol



MOTIVATION

- Example



Alice



Intruder



Bob



Charlene



Intruder



Bob

MOTIVATION

- Example



Alice



Bob



Charlene



Bob

MOTIVATION

- Example



Alice

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



Bob



Charlene



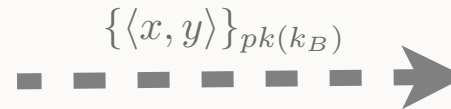
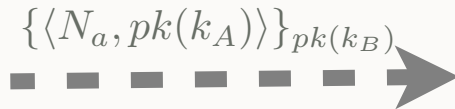
Bob

MOTIVATION

- Example



Alice



Bob



Charlene



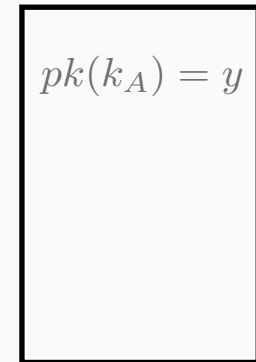
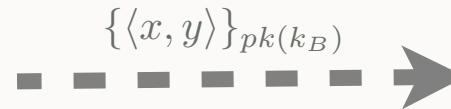
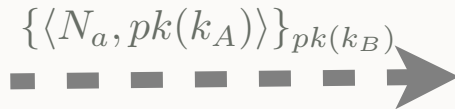
Bob

MOTIVATION

- Example



Alice



Bob



Charlene



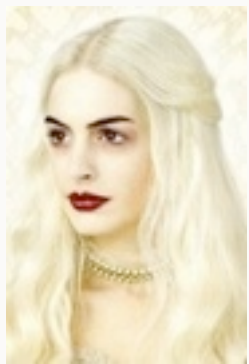
Bob

MOTIVATION

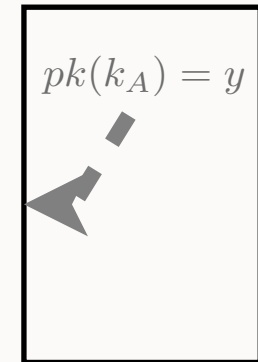
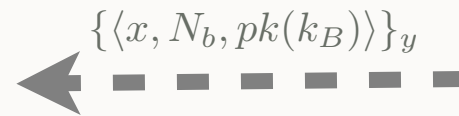
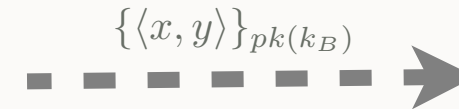
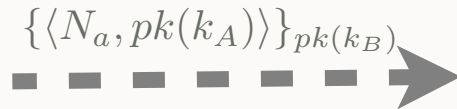
■ Example



Alice



Charlene



Bob



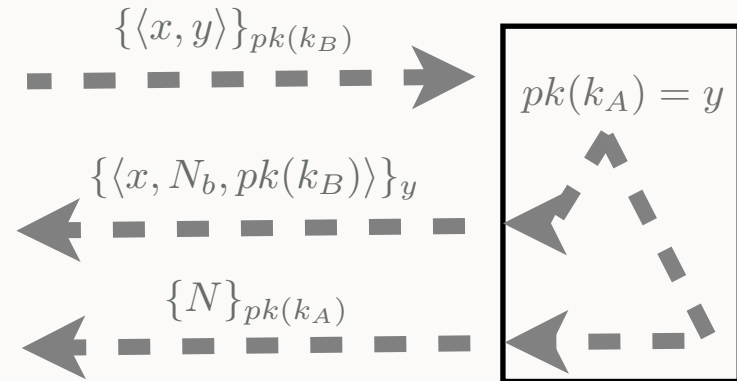
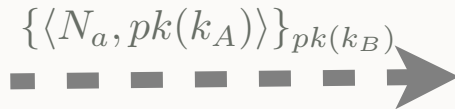
Bob

MOTIVATION

■ Example



Alice



Bob



Charlene



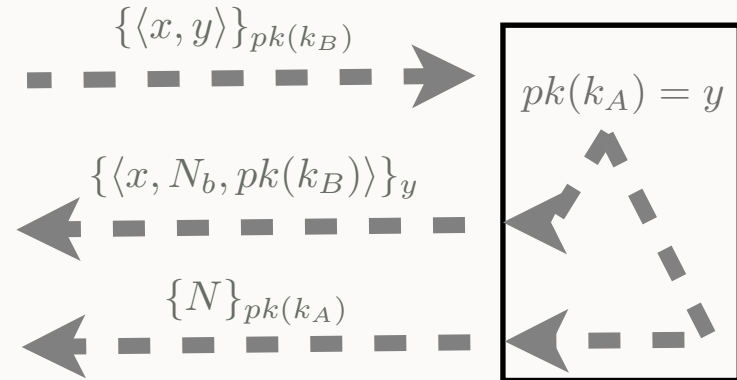
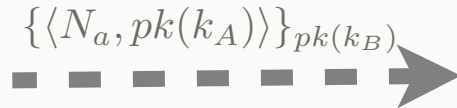
Bob

MOTIVATION

■ Example



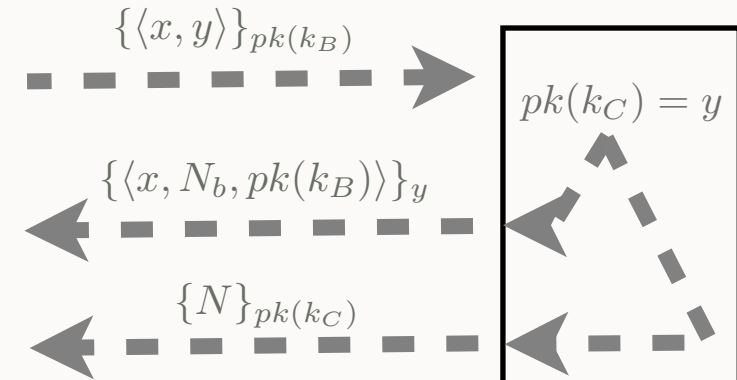
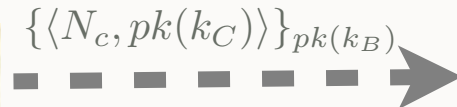
Alice



Bob



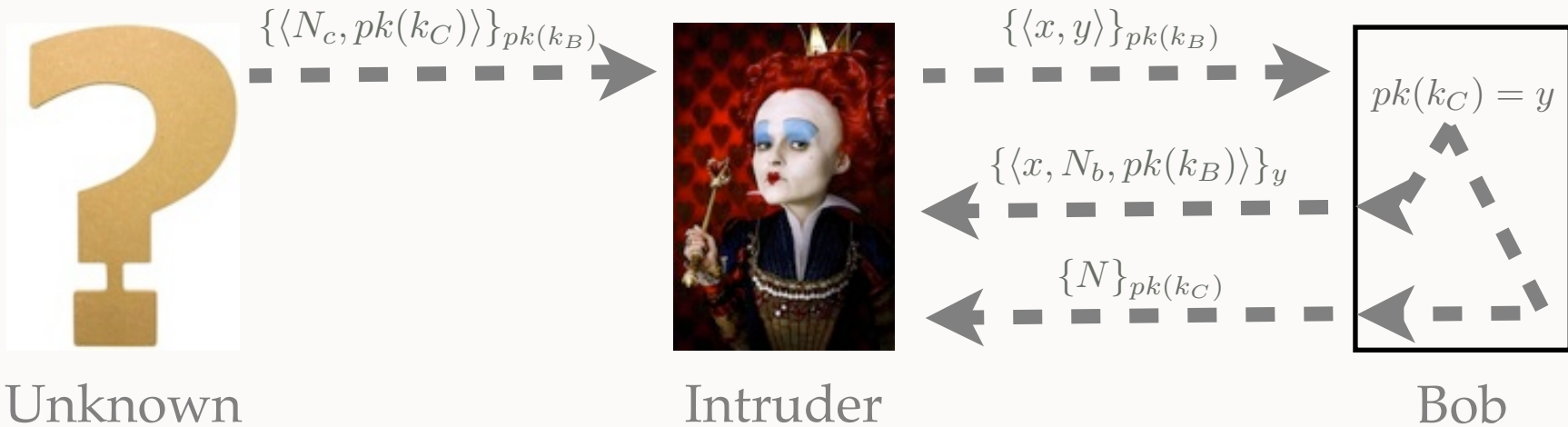
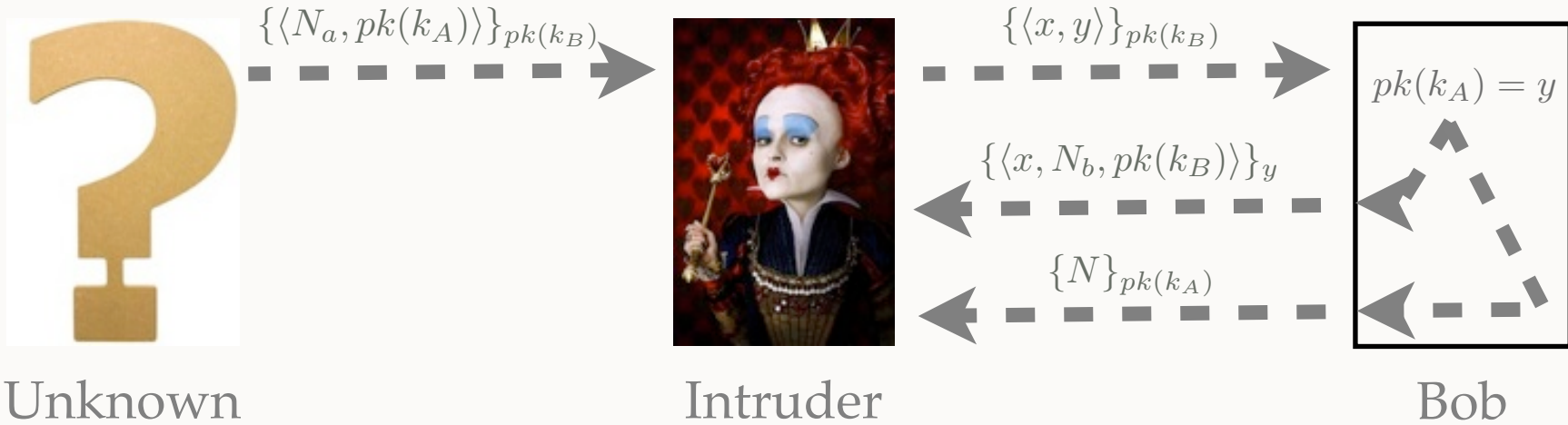
Charlene



Bob

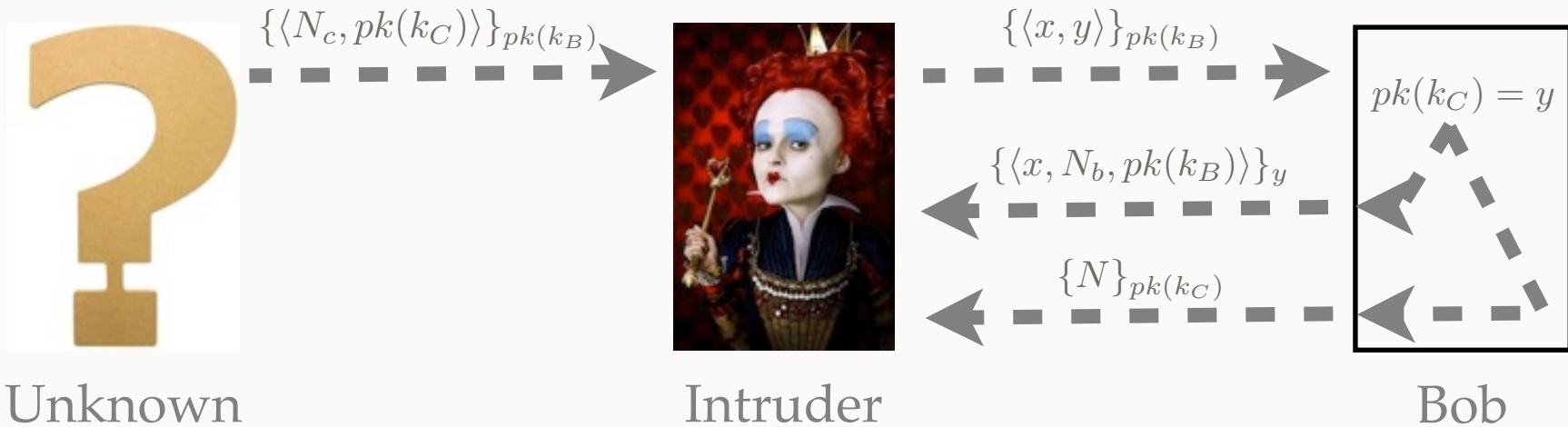
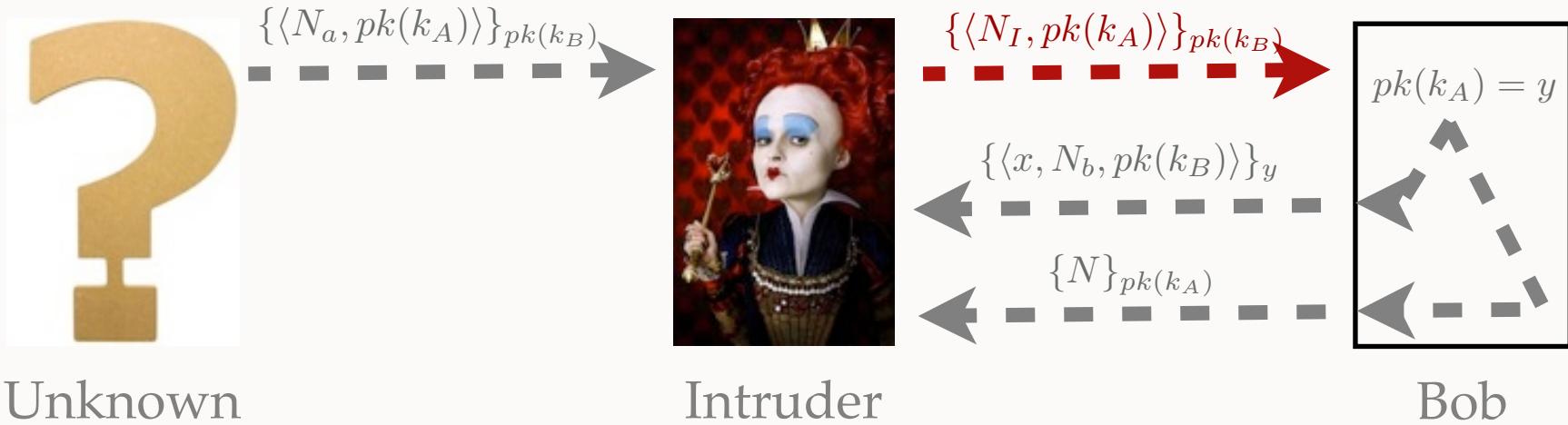
MOTIVATION

■ Example



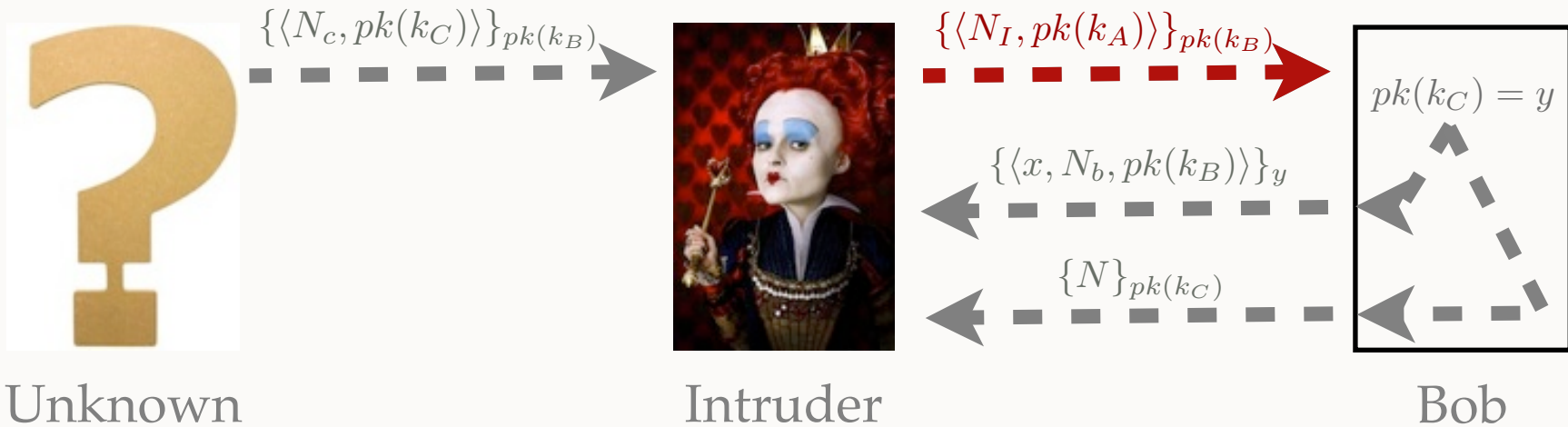
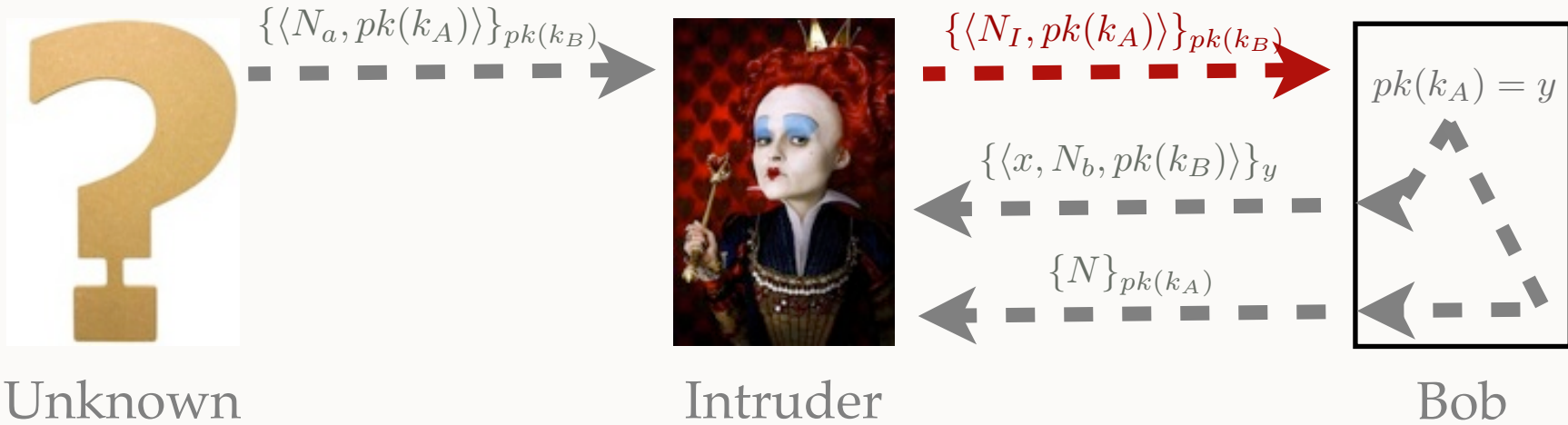
MOTIVATION

■ Example



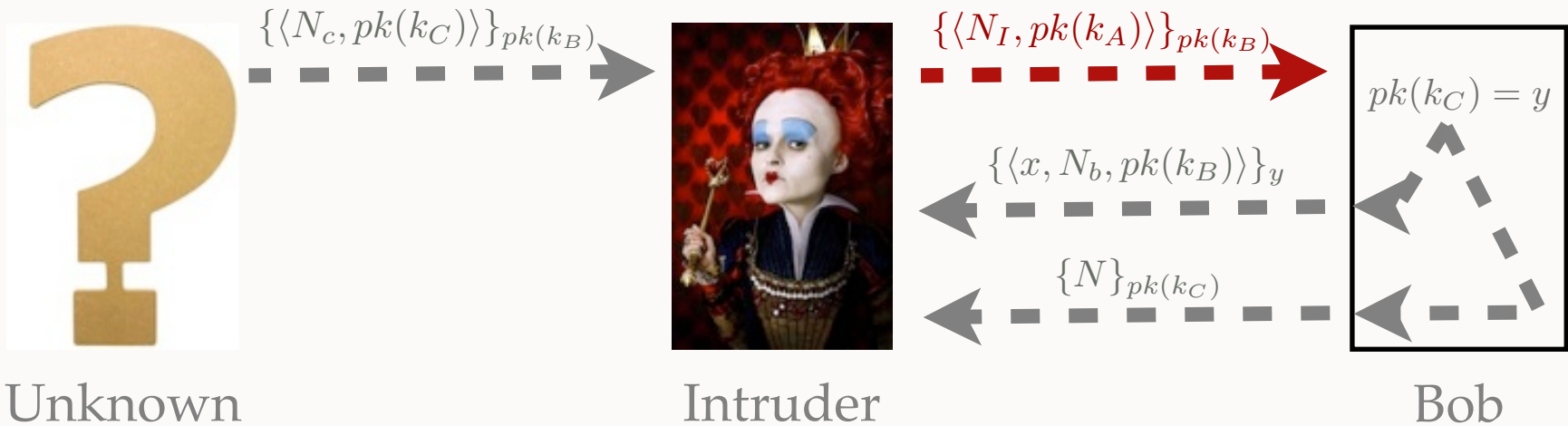
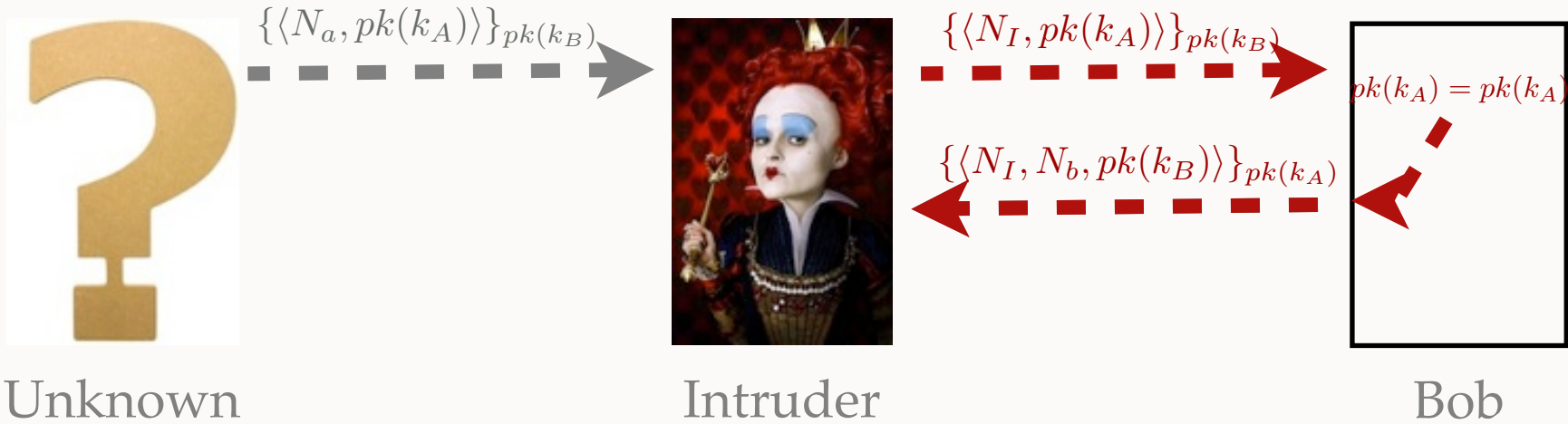
MOTIVATION

■ Example



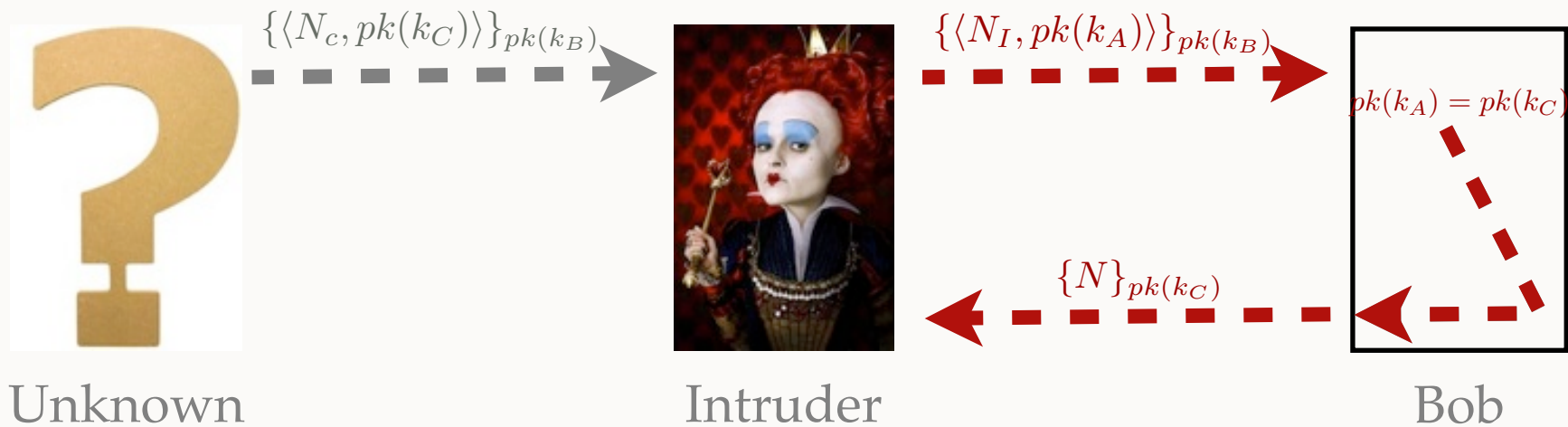
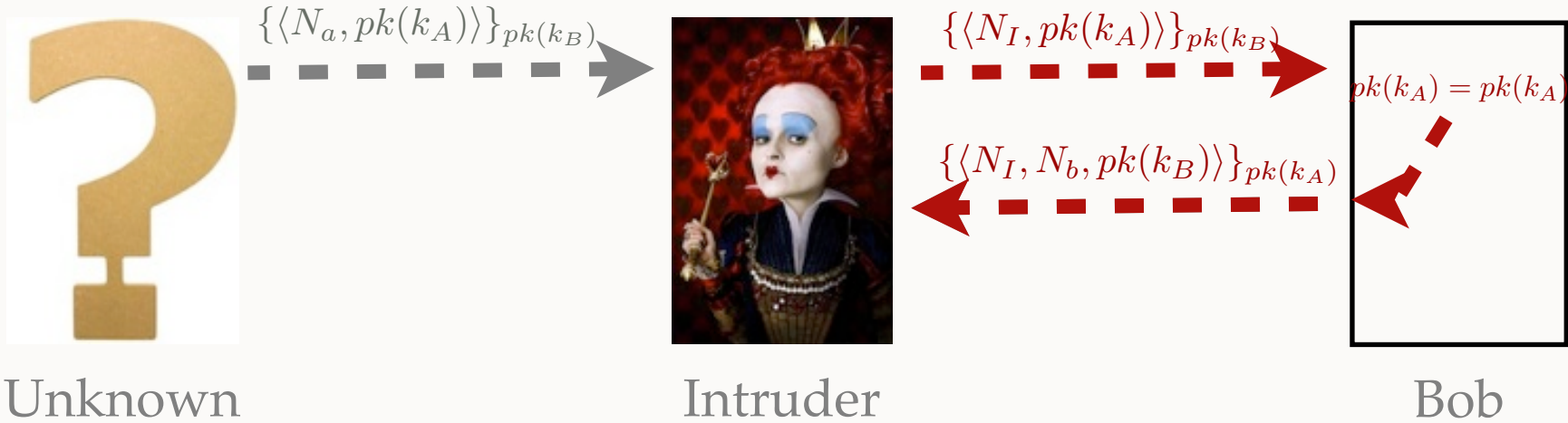
MOTIVATION

■ Example



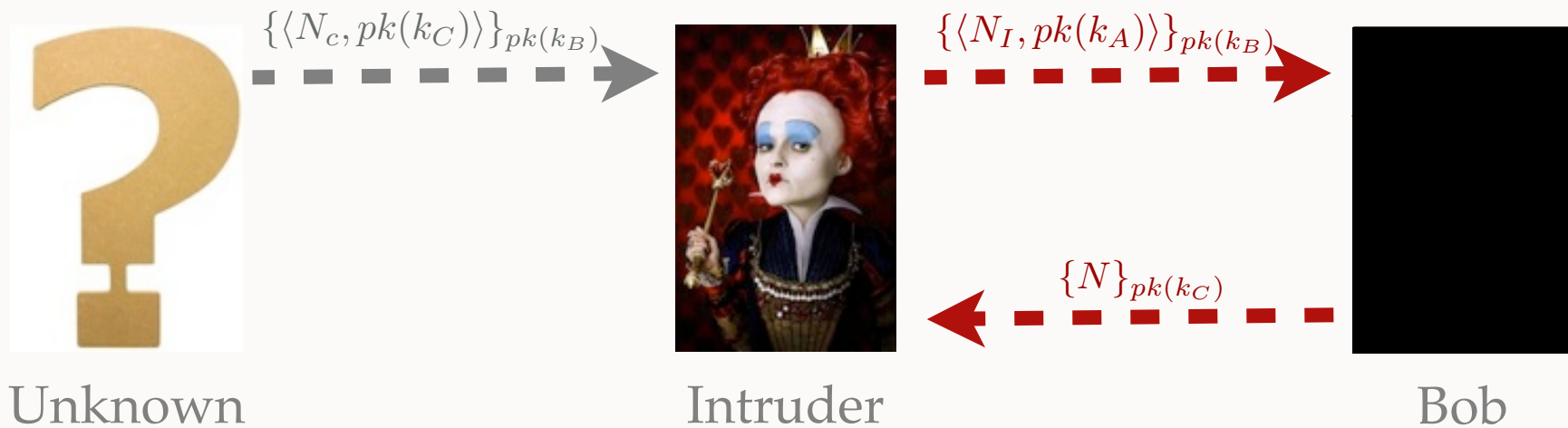
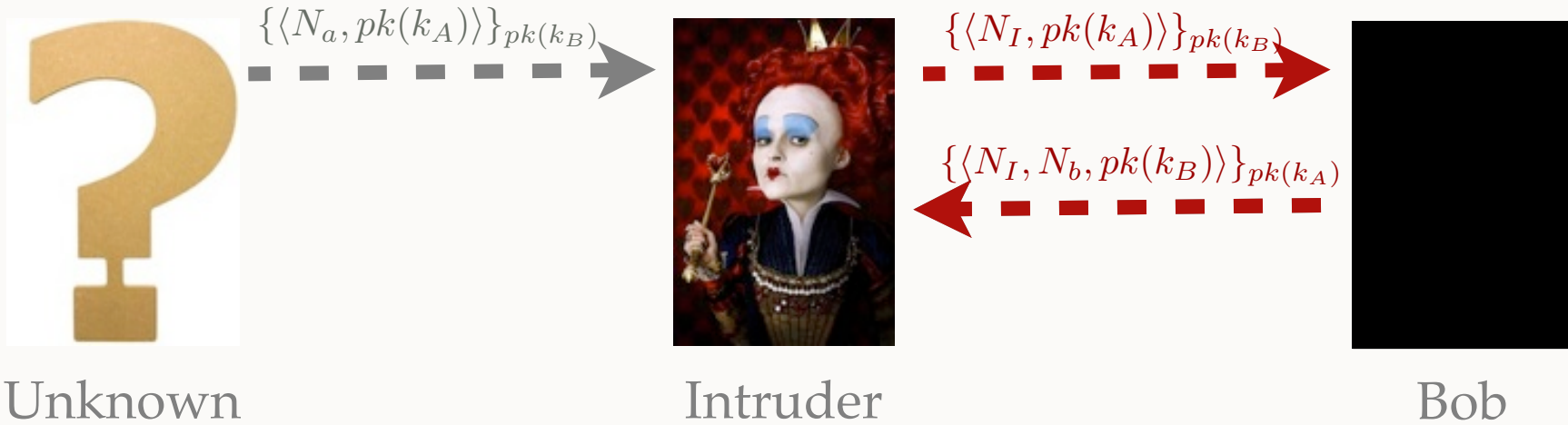
MOTIVATION

■ Example



MOTIVATION

■ Example



PROVERIF

- Was first an analyzer for reachability properties
- Based on Horn clauses
- Equational theory : constructors

$$\text{dec}(\{x\}_y, y) =_E x$$

$$\{\text{dec}(x, y)\}_y =_E x$$

- Rewrite rules : destructors

$$\text{dec}(\{x\}_y, y) \rightarrow x$$

$$\text{equal}(x, x) \rightarrow x$$

CONTRIBUTION

We introduce destructor with tests between terms

$$g(M_1, \dots, M_n) \rightarrow M \text{ with } \Phi$$

CONTRIBUTION

We introduce destructor with tests between terms

$$g(M_1, \dots, M_n) \rightarrow M \text{ with } \Phi$$

- The *If-then-else* destructor

$$\text{ifthenelse}(x, x, z, t) \rightarrow z$$

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

CONTRIBUTION

We introduce destructor with tests between terms

$$g(M_1, \dots, M_n) \rightarrow M \text{ with } \Phi$$

- The *If-then-else* destructor

$$\text{ifthenelse}(x, x, z, t) \rightarrow z$$

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

- The *dec-else* destructor

$$\text{dec}(\{x\}_y, y, t) \rightarrow x$$

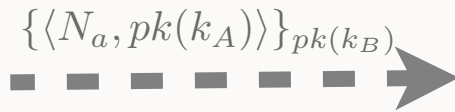
$$\text{dec}(x, y, t) \rightarrow t \quad \text{with } \forall z. x \neq \{z\}_y$$

CONTRIBUTION

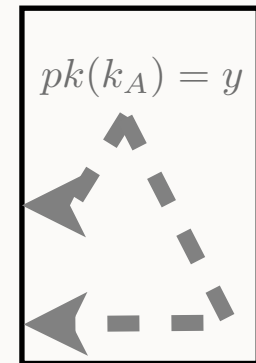
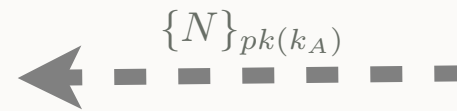
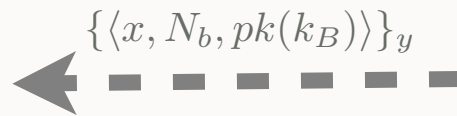
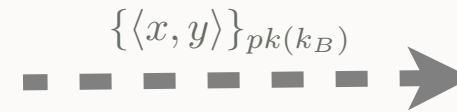
- Protocol transformation



Alice



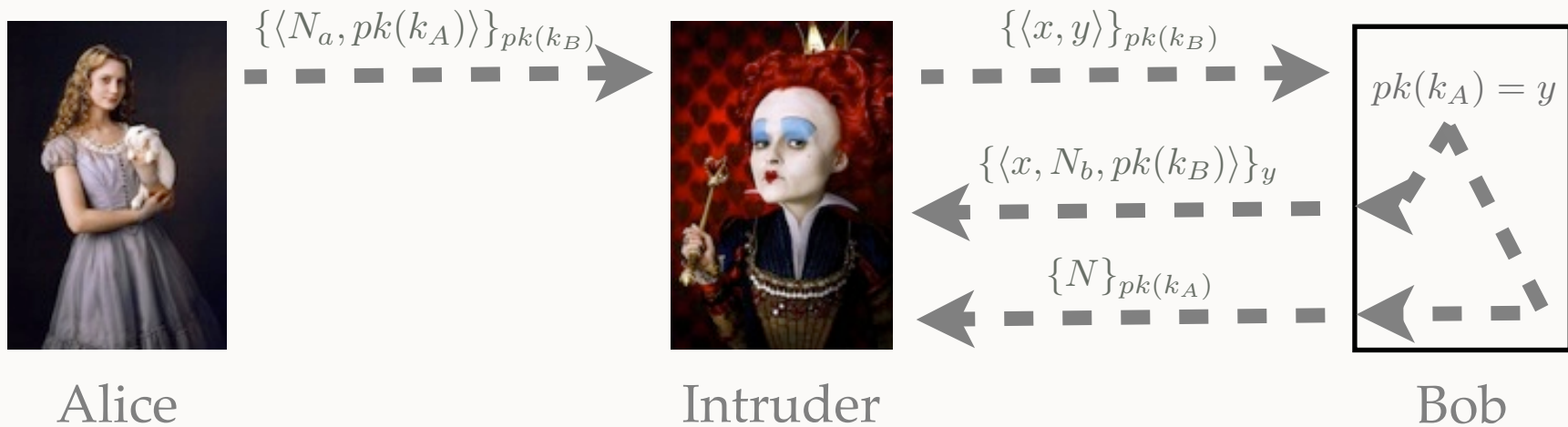
Intruder



Bob

CONTRIBUTION

- Protocol transformation



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

PROVERIF

What are the modifications in ProVerif algorithm ?

CURRENT PROVERIF

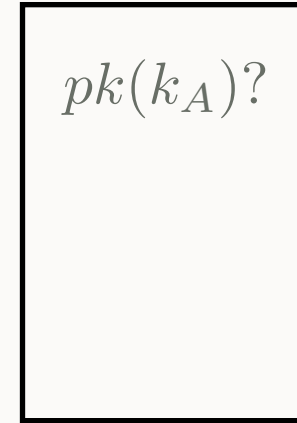
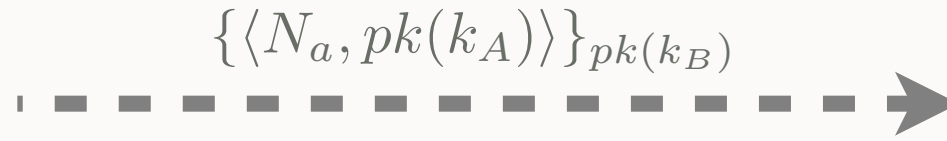
- Horn clauses

CURRENT PROVERIF

- Horn clauses



Alice



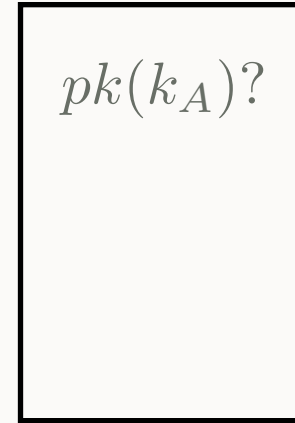
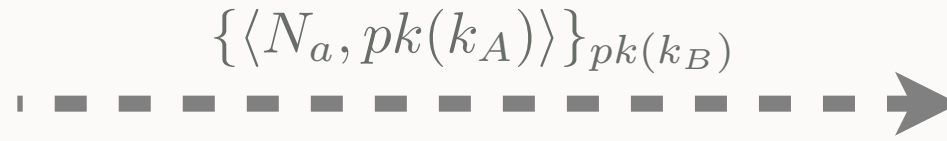
Bob

CURRENT PROVERIF

- Horn clauses



Alice



Bob

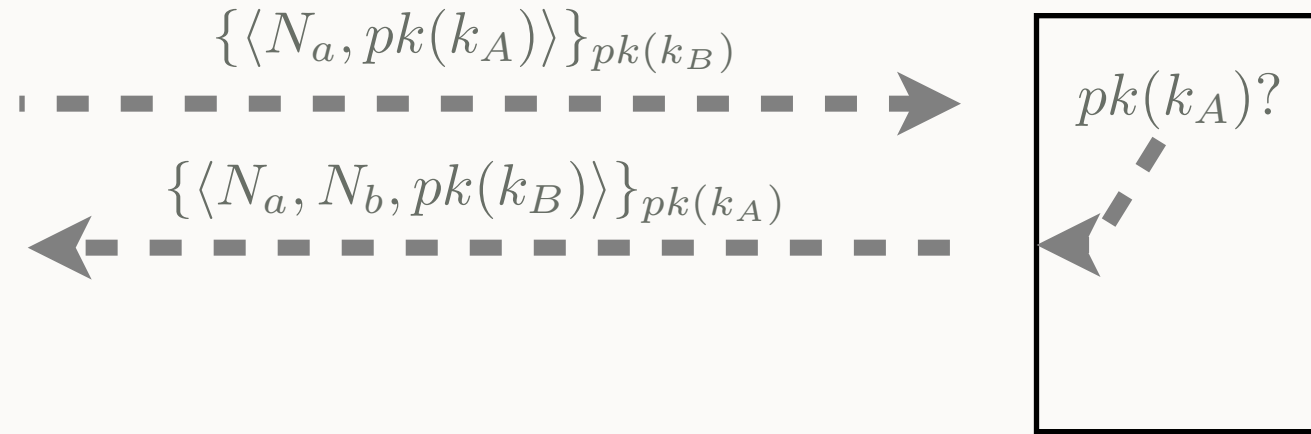
$\rightarrow \text{att}(\{N_a, pk(k_A)\}_{pk(k_B)})$

CURRENT PROVERIF

- Horn clauses



Alice



Bob

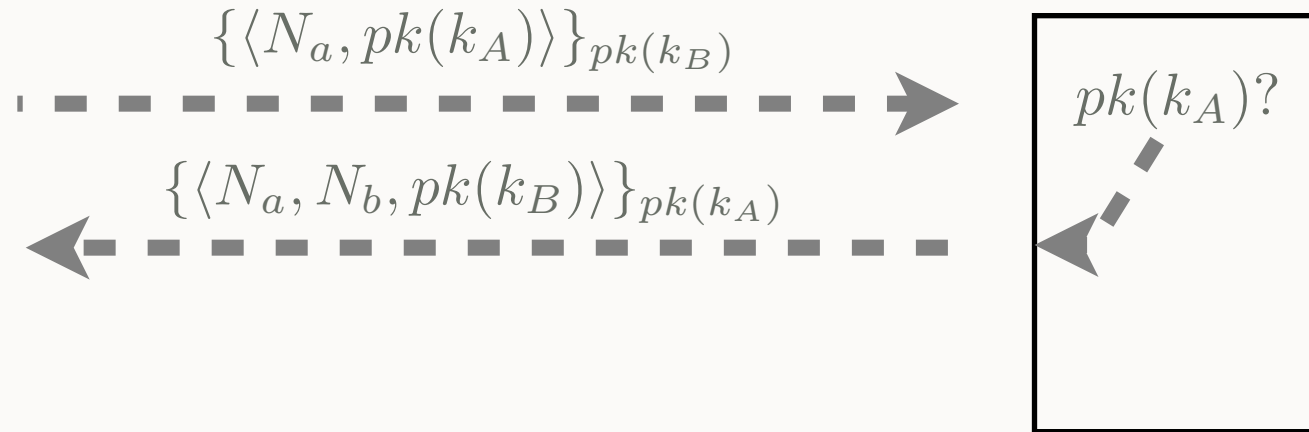
$$\rightarrow \text{att}(\{N_a, pk(k_A)\}_{pk(k_B)})$$

CURRENT PROVERIF

- Horn clauses



Alice



Bob

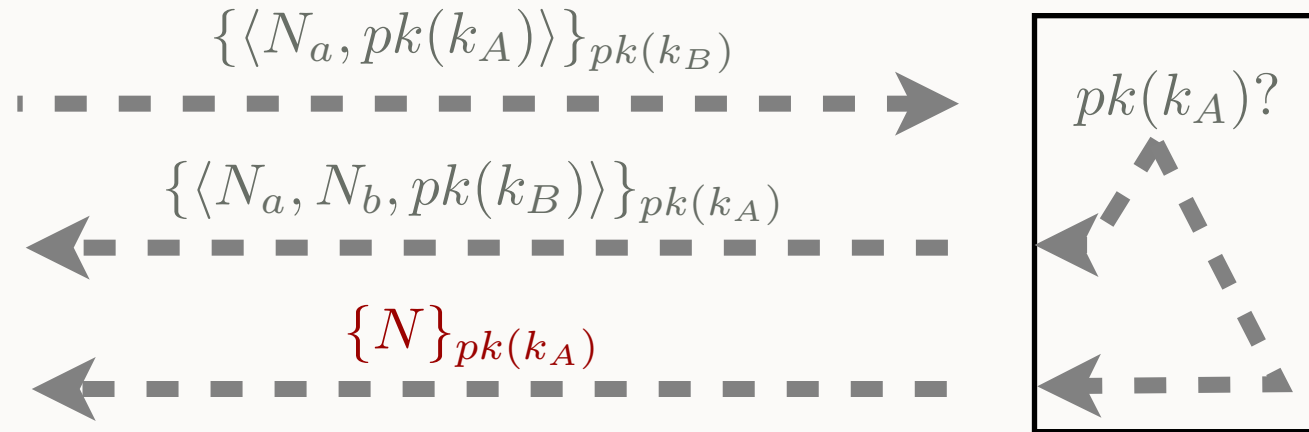
$$\begin{aligned} & \rightarrow \text{att}(\{N_a, pk(k_A)\}_{pk(k_B)}) \\ \text{att}(\{x, pk(k_A)\}_{pk(k_B)}) & \rightarrow \text{att}(\{x, N_b, pk(k_B)\}_{pk(k_A)}) \end{aligned}$$

CURRENT PROVERIF

- Horn clauses



Alice



Bob

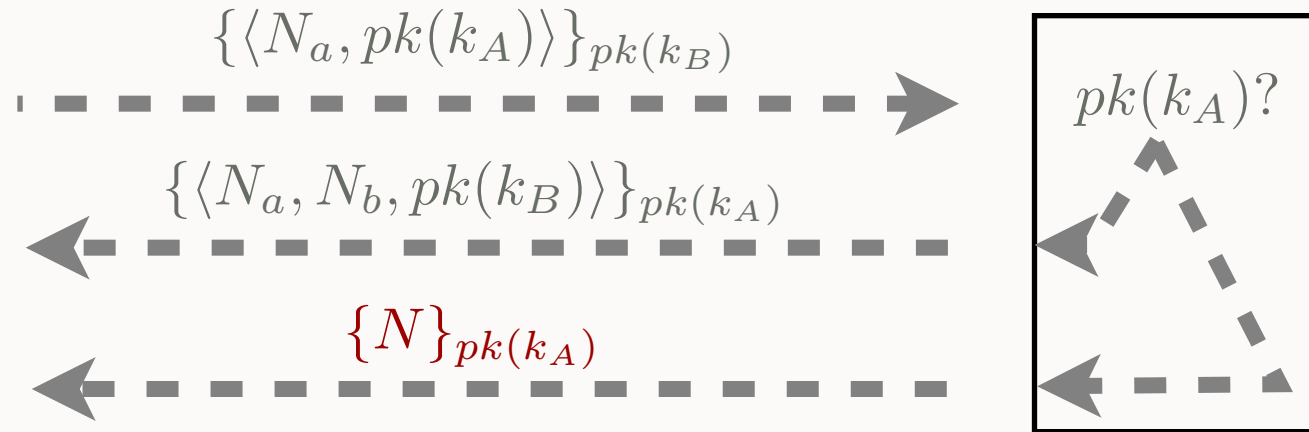
$$\begin{aligned} & \rightarrow \text{att}(\{N_a, pk(k_A)\}_{pk(k_B)}) \\ \text{att}(\{x, pk(k_A)\}_{pk(k_B)}) & \rightarrow \text{att}(\{x, N_b, pk(k_B)\}_{pk(k_A)}) \end{aligned}$$

CURRENT PROVERIF

- Horn clauses



Alice



Bob

$$\rightarrow att(\{N_a, pk(k_A)\}_{pk(k_B)})$$

$$att(\{x, pk(k_A)\}_{pk(k_B)}) \rightarrow att(\{x, N_b, pk(k_B)\}_{pk(k_A)})$$

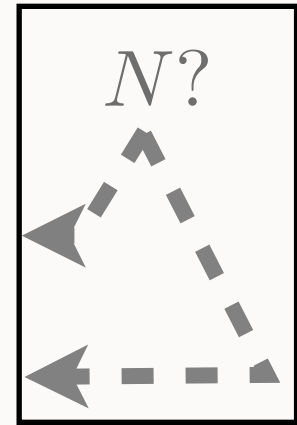
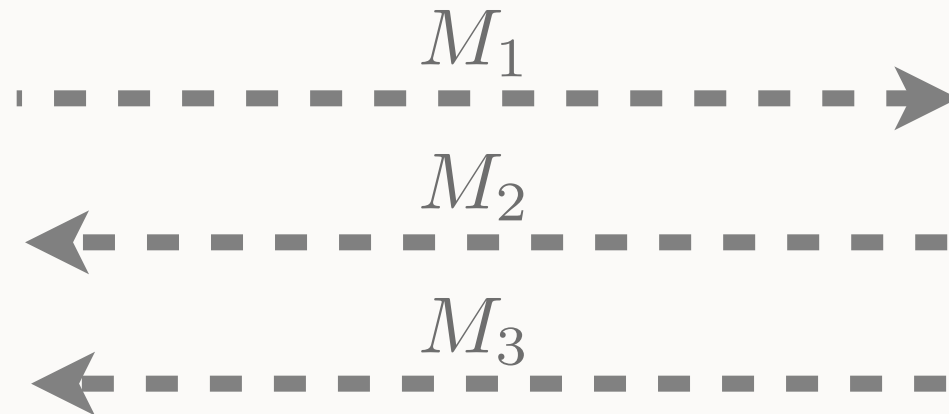
$$att(x) \wedge \forall y. x \neq \{y, pk(k_A)\}_{pk(k_B)} \rightarrow att(\{N\}_{pk(k_A)})$$

CURRENT PROVERIF

- Biprocess



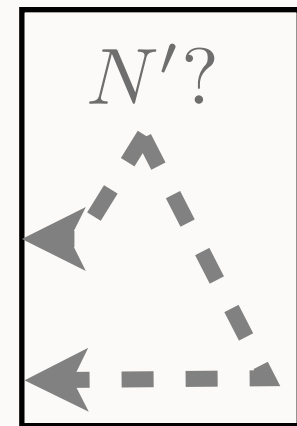
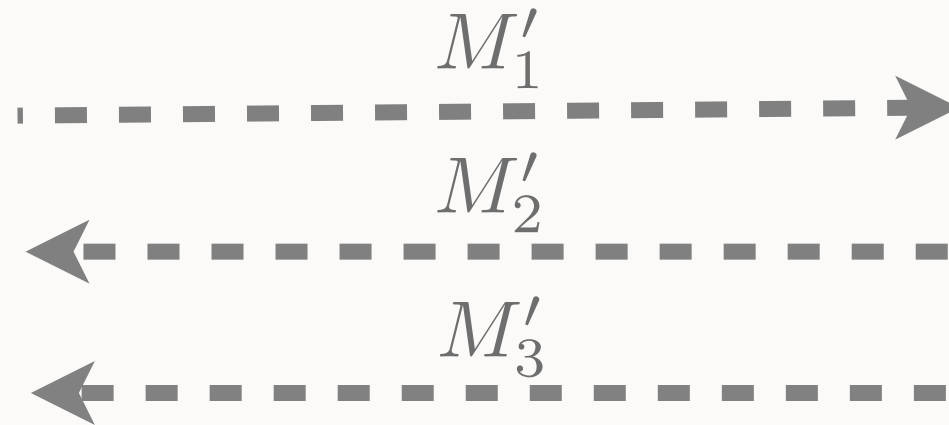
Alice



Bob



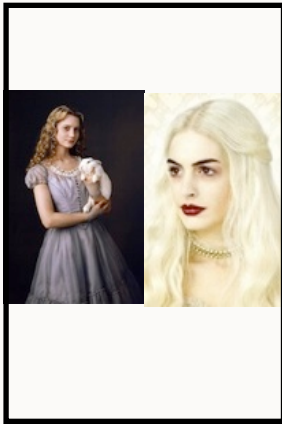
Charlene



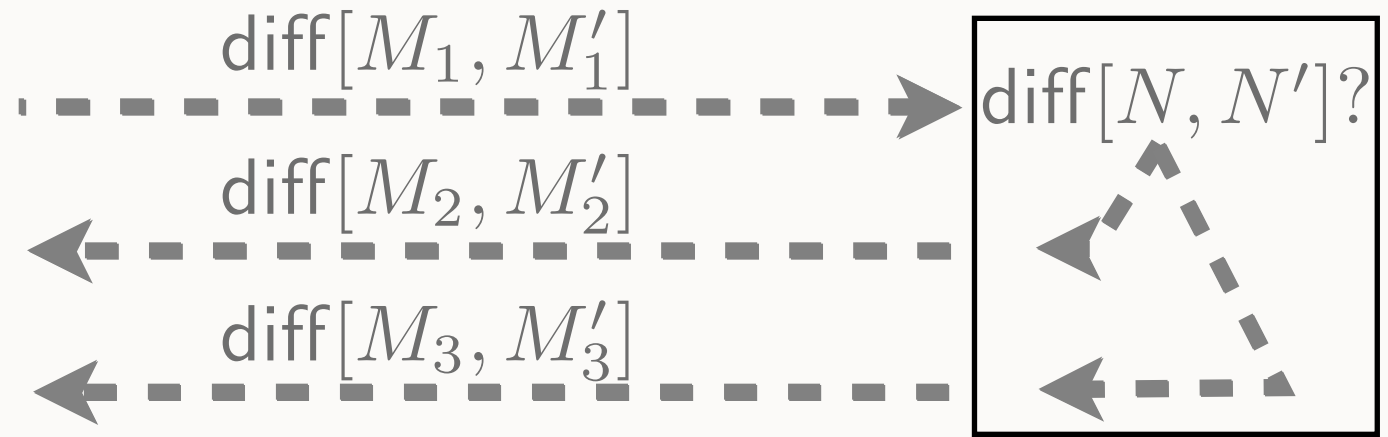
Bob

CURRENT PROVERIF

- Biprocess



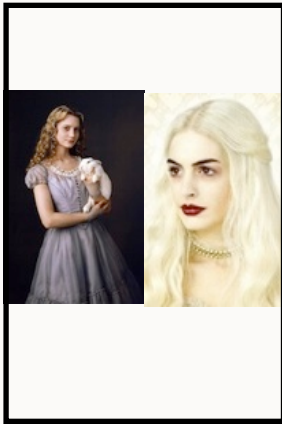
Alice Charlene



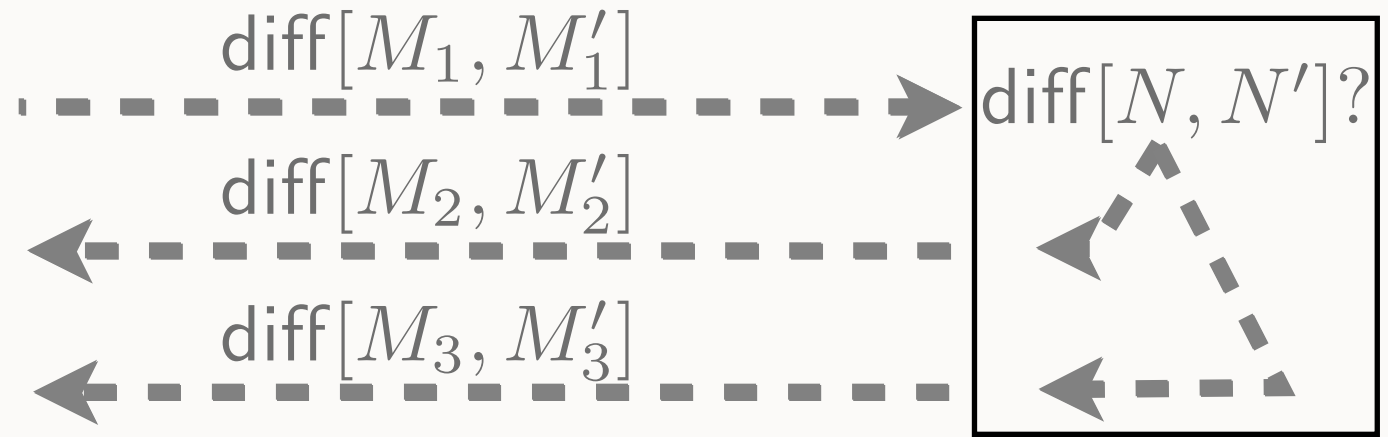
Bob

CURRENT PROVERIF

- Biprocess



Alice Charlene

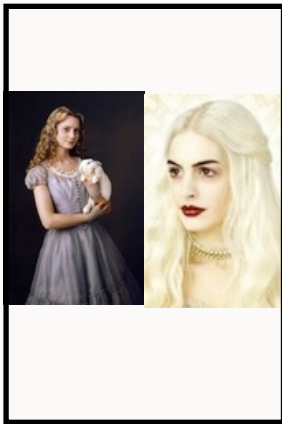


Bob

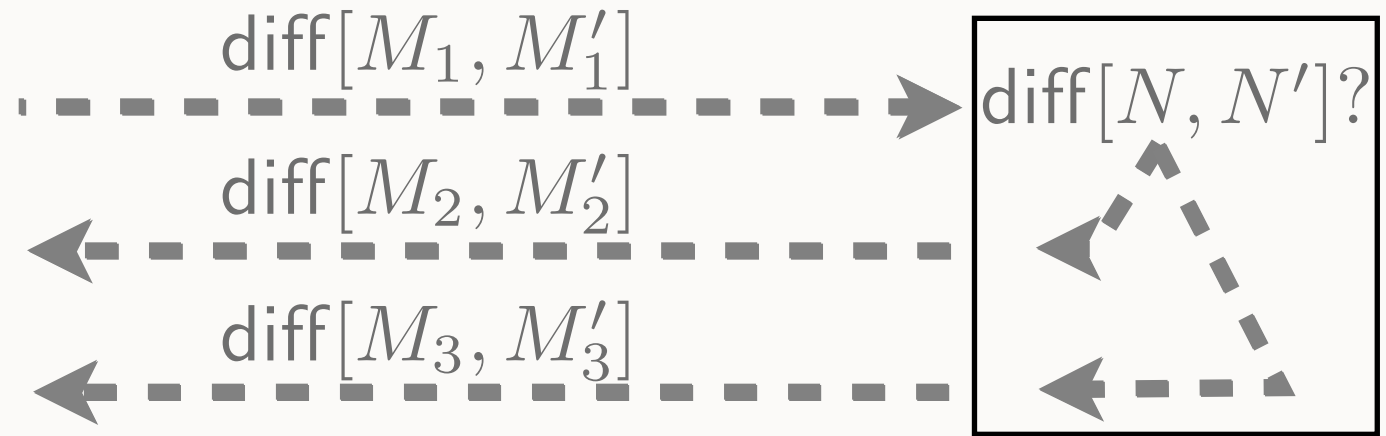
$\rightarrow \text{att}(M_1, M'_1)$

CURRENT PROVERIF

- Biprocess



Alice Charlene



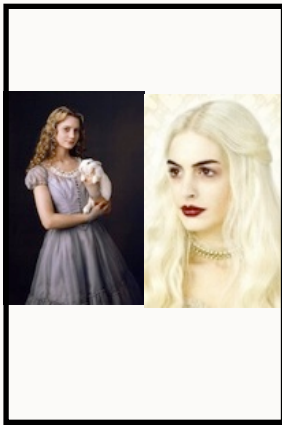
Bob

$$\rightarrow \text{att}(M_1, M'_1)$$

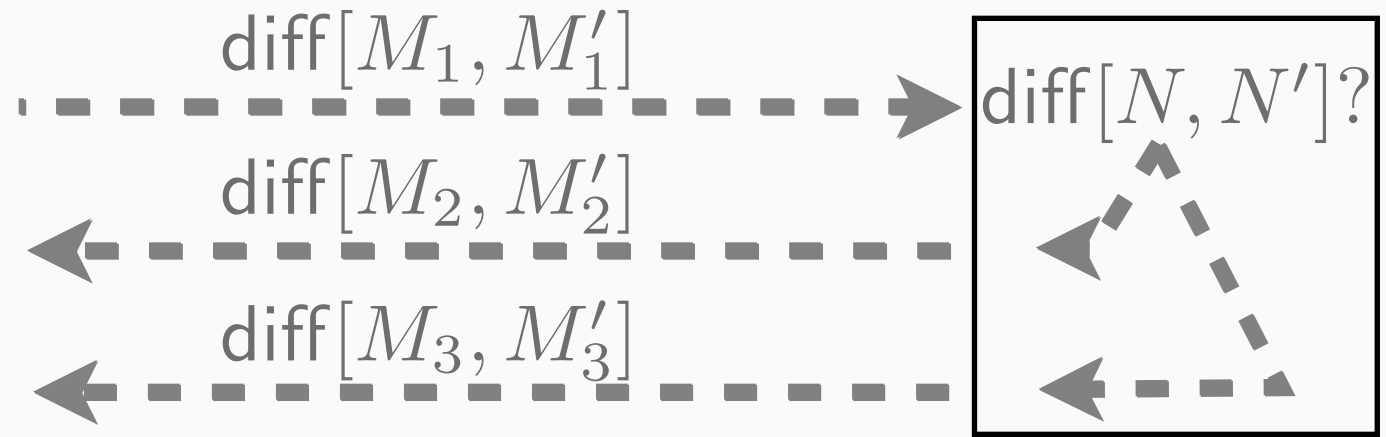
$$\text{att}(N, N') \rightarrow \text{att}(M_2, M'_2)$$

CURRENT PROVERIF

■ Biprocess



Alice Charlene



Bob

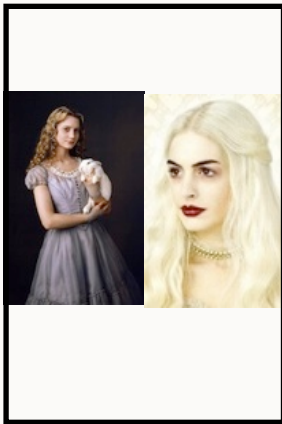
$$\rightarrow \text{att}(M_1, M'_1)$$

$$\text{att}(N, N') \rightarrow \text{att}(M_2, M'_2)$$

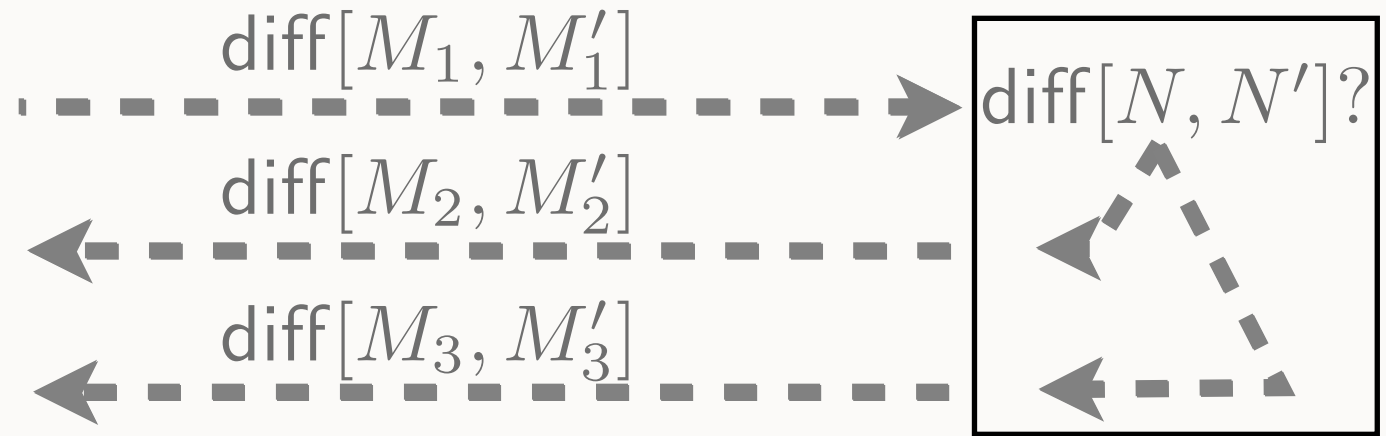
$$\text{att}(x, x') \wedge \forall y, y'. (x, x') \neq (N, N') \rightarrow \text{att}(M_2, M'_2)$$

CURRENT PROVERIF

- Biprocess



Alice Charlène



Bob

$$\rightarrow \text{att}(M_1, M'_1)$$

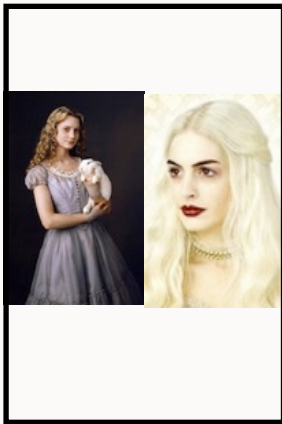
$$\text{att}(N, N') \rightarrow \text{att}(M_2, M'_2)$$

$$\text{att}(x, x') \wedge \forall y, y'. (x, x') \neq (N, N') \rightarrow \text{att}(M_2, M'_2)$$

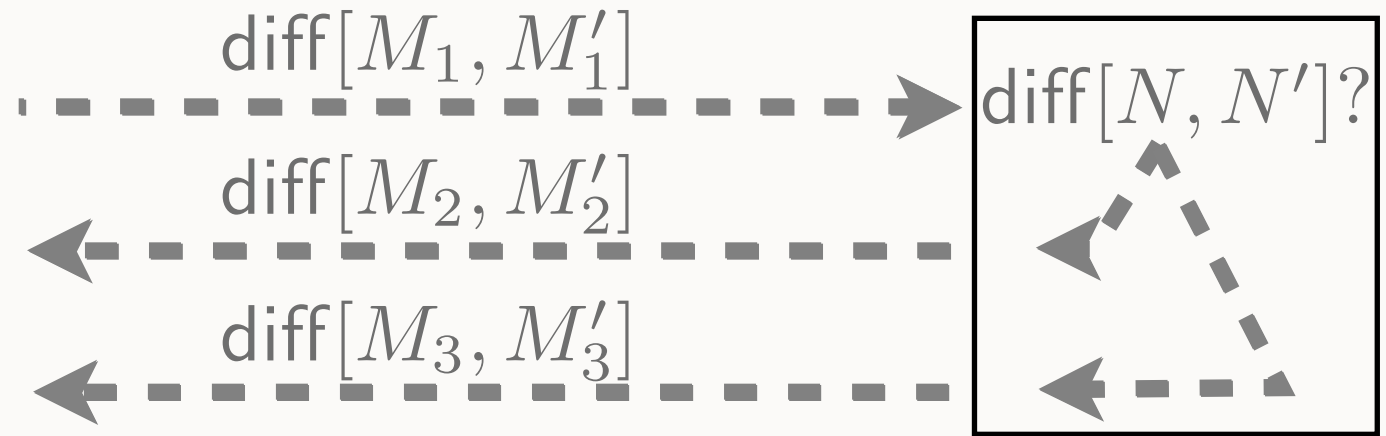
$$\text{att}(N, x') \wedge \forall y. x' \neq N' \rightarrow \text{bad}$$

CURRENT PROVERIF

- Biprocess



Alice Charlene



Bob

$$\rightarrow \text{att}(M_1, M'_1)$$

$$\text{att}(N, N') \rightarrow \text{att}(M_2, M'_2)$$

$$\text{att}(x, x') \wedge \forall y, y'. (x, x') \neq (N, N') \rightarrow \text{att}(M_2, M'_2)$$

$$\text{att}(N, x') \wedge \forall y. x' \neq N' \rightarrow \text{bad}$$

$$\text{att}(x, N') \wedge \forall y. x \neq N \rightarrow \text{bad}$$

CURRENT PROVERIF

- Attacker clauses

$$\rightarrow \text{att}(a, a) \quad \text{with } a \text{ public}$$

$$\text{att}(x, x') \wedge \text{att}(y, y') \rightarrow \text{att}(\{x\}_y, \{x'\}_{y'})$$

- New destructor reduction

$$\text{att}(\{x\}_y, \{x'\}_{y'}) \wedge \text{att}(y, y') \rightarrow \text{att}(x, x')$$

- Destructor failure

$$\text{att}(\{x\}_y, x') \wedge \text{att}(y, y') \wedge \forall z. x' \neq \{z\}_{y'} \rightarrow \text{bad}$$

$$\text{att}(x, x') \wedge \text{att}(x, y') \wedge x' \neq y' \rightarrow \text{bad}$$

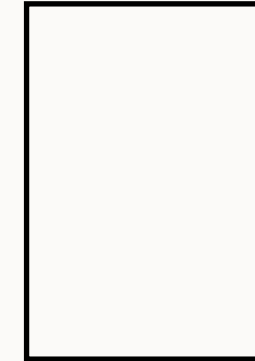
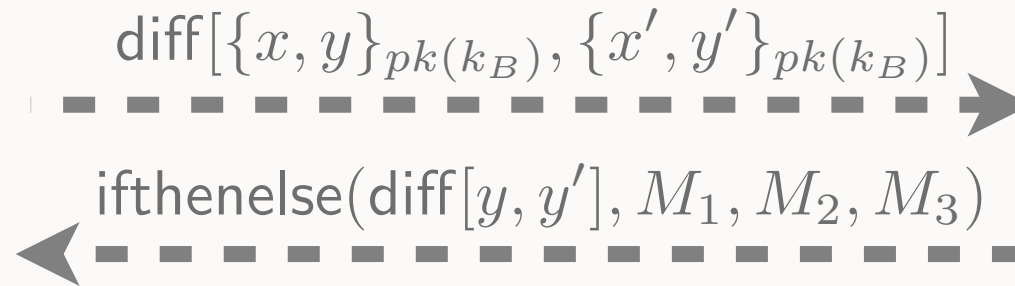
PROVERIF

A biprocess is *equivalent* if its horn clauses cannot derive *bad*

NEW PROVERIF



Intruder



Bob

$$M_1 = \text{diff}[pk(k_A), pk(k_C)]$$

$$M_2 = \text{diff}[\{x, N_b, pk(k_B)\}_y, \{x', N'_b, pk(k_B)\}_y]$$

$$M_3 = \text{diff}[\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}]$$

- Case 1 : Both tests succeed

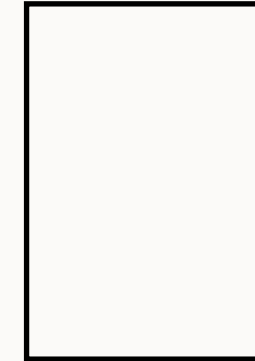
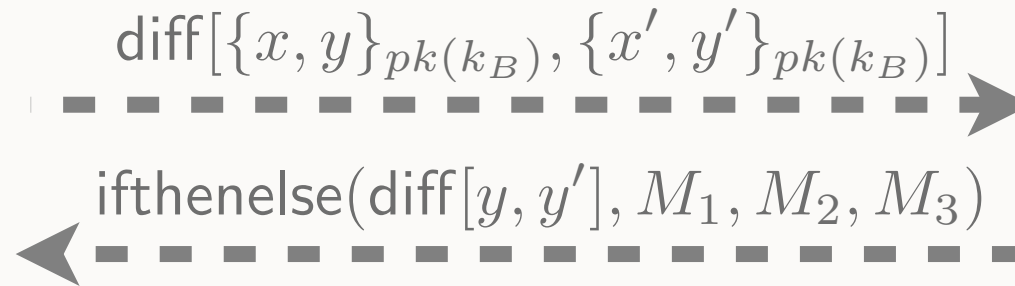
$$\rightarrow \text{att}(\{x, pk(k_A)\}_{pk(k_B)}, \{x', pk(k_C)\}_{pk(k_B)})$$

$$\rightarrow \text{att}(\{x, N_b, pk(k_B)\}_{pk(k_A)}, \{x', N'_b, pk(k_B)\}_{pk(k_C)})$$

NEW PROVERIF



Intruder



Bob

$$M_1 = \text{diff}[pk(k_A), pk(k_C)]$$

$$M_2 = \text{diff}[\{x, N_b, pk(k_B)\}_y, \{x', N'_b, pk(k_B)\}_y]$$

$$M_3 = \text{diff}[\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}]$$

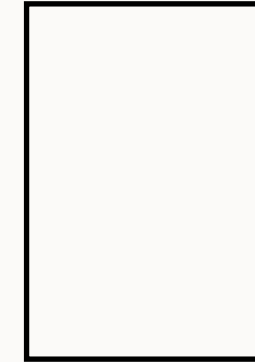
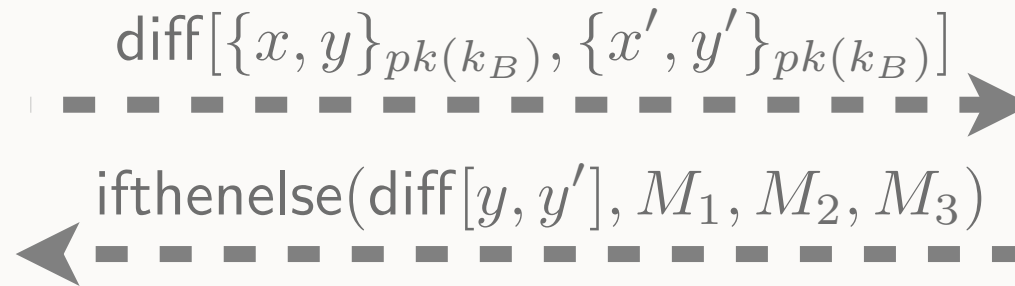
- Case 2 : Both tests fail

$$\begin{aligned} & \text{att}(\{x, y'\}_{pk(k_B)}, \{x', y'\}_{pk(k_B)}) \\ \wedge & \quad y \neq pk(k_A) \wedge y' \neq pk(k_C) \\ \rightarrow & \quad \text{att}(\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}) \end{aligned}$$

NEW PROVERIF



Intruder



Bob

$$M_1 = \text{diff}[pk(k_A), pk(k_C)]$$

$$M_2 = \text{diff}[\{x, N_b, pk(k_B)\}_y, \{x', N'_b, pk(k_B)\}_y]$$

$$M_3 = \text{diff}[\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}]$$

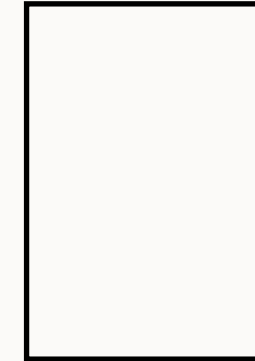
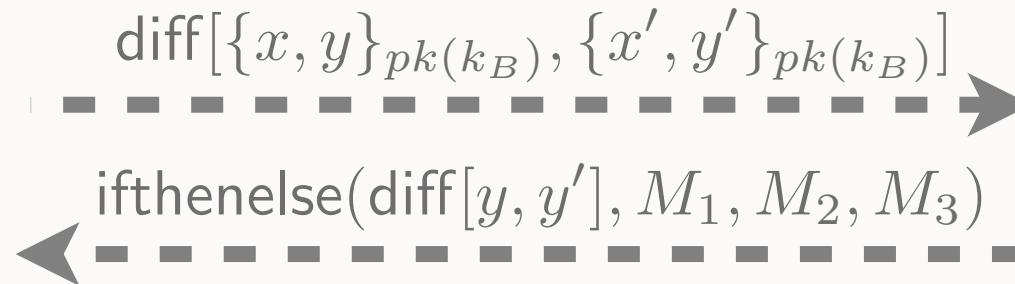
- Old case 3 : Left test succeeds, right test fails

$$\begin{aligned} & \text{att}(\{x, pk(k_A)\}_{pk(k_B)}, \{x', y'\}_{pk(k_B)}) \\ & \wedge y' \neq pk(k_C) \\ & \rightarrow \text{bad} \end{aligned}$$

NEW PROVERIF



Intruder



Bob

$$M_1 = \text{diff}[pk(k_A), pk(k_C)]$$

$$M_2 = \text{diff}[\{x, N_b, pk(k_B)\}_y, \{x', N'_b, pk(k_B)\}_y]$$

$$M_3 = \text{diff}[\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}]$$

- New case 3 : Left test succeeds, right test fails

$$\begin{aligned}
 & \text{att}(\{x, pk(k_A)\}_{pk(k_B)}, \{x', y'\}_{pk(k_B)}) \\
 \wedge & \quad y' \neq pk(k_C) \\
 \rightarrow & \quad \text{att}(\{x, N_b, pk(k_B)\}_{pk(k_A)}, \{N'\}_{pk(k_C)})
 \end{aligned}$$

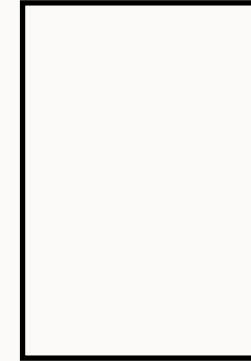
NEW PROVERIF



Intruder

$\text{diff}[\{x, y\}_{pk(k_B)}, \{x', y'\}_{pk(k_B)}]$

$\text{ifthenelse}(\text{diff}[y, y'], M_1, M_2, M_3)$



Bob

$$M_1 = \text{diff}[pk(k_A), pk(k_C)]$$

$$M_2 = \text{diff}[\{x, N_b, pk(k_B)\}_y, \{x', N'_b, pk(k_B)\}_y]$$

$$M_3 = \text{diff}[\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}]$$

- Old case 4 : Left test fails, right test succeeds

$\text{att}(\{x, y\}_{pk(k_B)}, \{x', pk(k_C)\}_{pk(k_B)})$

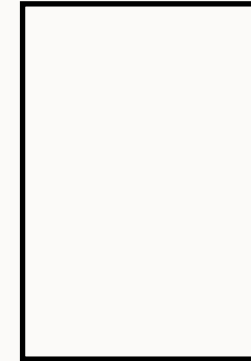
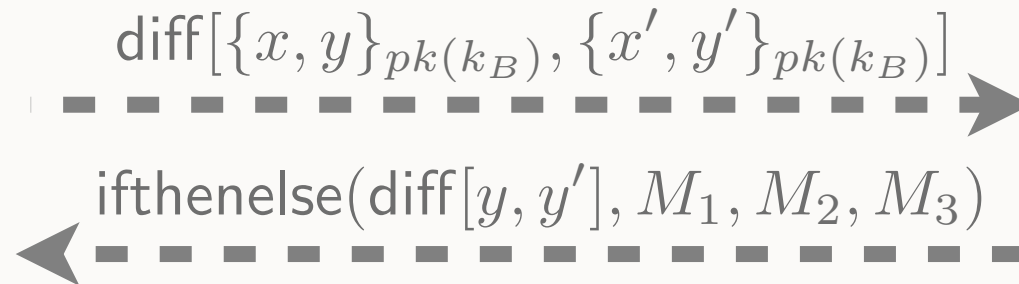
$\wedge y \neq pk(k_A)$

\rightarrow **bad**

NEW PROVERIF



Intruder



Bob

$$M_1 = \text{diff}[pk(k_A), pk(k_C)]$$

$$M_2 = \text{diff}[\{x, N_b, pk(k_B)\}_y, \{x', N'_b, pk(k_B)\}_y]$$

$$M_3 = \text{diff}[\{N\}_{pk(k_A)}, \{N'\}_{pk(k_C)}]$$

- New case 4 : Left test fails, right test succeeds

$$\begin{aligned} & \text{att}(\{x, y\}_{pk(k_B)}, \{x', pk(k_C)\}_{pk(k_B)}) \\ \wedge & \quad y \neq pk(k_A) \\ \rightarrow & \quad \text{att}(\{N\}_{pk(k_A)}, \{x', N'_b, pk(k_B)\}_{pk(k_C)}) \end{aligned}$$

CONCLUSION

■ Contribution

We introduce destructor with tests between terms

$$g(M_1, \dots, M_n) \rightarrow M \text{ with } \Phi$$

- + Prove equivalence on larger class of protocol (e.g. private authentication)
- + Can perform automatic detection and transformation of protocol
- + Could be used for reachability properties
- E-passport protocol is still not handle by ProVerif

■ Future work

- Remove even further false attack due to parameter of nonces
- E-passport protocol ?