Automatic Verification of Privacy Protection for Unbounded Sessions

Shonan

Lucca Hirschi

October 28, 2015

joint work with David Baelde (LSV) *and* Stéphanie Delaune (LSV)

Introduction

⇝ we need formal verification of crypto protocols covering privacy

Introduction

⇝ we need formal verification of crypto protocols covering privacy

Goal:

- checking privacy (unlinkability and anonymity)
- in the symbolic model (Dolev-Yao)
- for unbounded sessions

Introduction

⇝ we need formal verification of crypto protocols covering privacy

Goal:

- checking privacy (unlinkability and anonymity)
- in the symbolic model (Dolev-Yao)
- for unbounded sessions

- *Unlinkability (=untraceability)* [ISO/IEC 15408]:

  Ensuring that a user may make multiple uses of a service or resource without others being able to link these uses together.

- *Anonymity* [ISO/IEC 15408]:

  Ensuring that a user may use a service or resource without disclosing the user's identity. [...]

Context

Strong unlinkability [Ryan *et al.* CSF'10]:

$$\underbrace{!\, \nu \vec{k}\, !\, \nu \vec{n}(T \mid R)}_{\mathcal{M}} \approx \underbrace{!\, \nu \vec{k}.\nu \vec{n}(T \mid R)}_{\mathcal{S}}$$

Intuition: $\mathcal{M} \sqsubseteq \mathcal{S}$

$\forall$  and behaviour of ($\mathcal{M} \|$ ) producing observable $\mathcal{D}$
$\Rightarrow \exists$ behaviour of ($\mathcal{S} \|$ ) producing observable $\mathcal{D}' \sim \mathcal{D}$

Context

Strong unlinkability [Ryan *et al.* CSF'10]:

$$\underbrace{!\ \nu \vec{k}\ !\ \nu \vec{n}(T \mid R)}_{\mathcal{M}} \approx \underbrace{!\ \nu \vec{k}.\nu \vec{n}(T \mid R)}_{\mathcal{S}}$$

Intuition: $\mathcal{M} \sqsubseteq \mathcal{S}$

$\forall$ and behaviour of $(\mathcal{M} \|)$ producing observable $\mathcal{D}$
$\Rightarrow \exists$ behaviour of $(\mathcal{S} \|)$ producing observable $\mathcal{D}' \sim \mathcal{D}$

- Checking this is undecidable (because of !)

Existing approaches:

- manual: need to exhib huge bisimulations
- automatic (ProVerif/Maude-NPA/Tamarin):
  rely on abstraction (diff-equivalence) not enough precise
  $\rightsquigarrow$ always fail to prove unlinkability

Context

Strong unlinkability [Ryan *et al.* CSF'10]:

$$\underbrace{!\,\nu\,\vec{k}\,!\,\nu\,\vec{n}(T \mid R)}_{\mathcal{M}} \approx \underbrace{!\,\nu\,\vec{k}.\nu\,\vec{n}(T \mid R)}_{\mathcal{S}}$$

Intuition: $\mathcal{M} \sqsubseteq \mathcal{S}$

$\forall$ and behaviour of $(\mathcal{M} \|)$ producing observable $\mathcal{D}$
$\Rightarrow \exists$ behaviour of $(\mathcal{S} \|)$ producing observable $\mathcal{D}' \sim \mathcal{D}$

- Checking this is undecidable (because of !)

Existing approaches:

- manual: need to exhib huge bisimulations
- automatic (ProVerif/Maude-NPA/Tamarin):
  rely on abstraction (diff-equivalence) not enough precise
  $\rightsquigarrow$ always fail to prove unlinkability

$\rightsquigarrow$ there is a need for dedicated abstraction targeting privacy

Contribution

We identify:

- 2 conditions implying unlinkability and anonymity
- for a class of 2-agents protocols including our target case studies

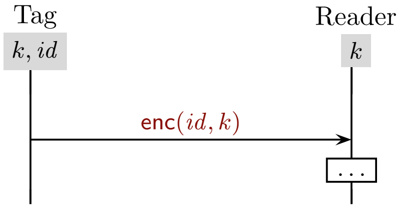
We make sure:

- our conditions can be checked automatically using ProVerif
- they correspond to good design practices

$\rightsquigarrow$ sound approach to check automatically privacy properties working well in practice

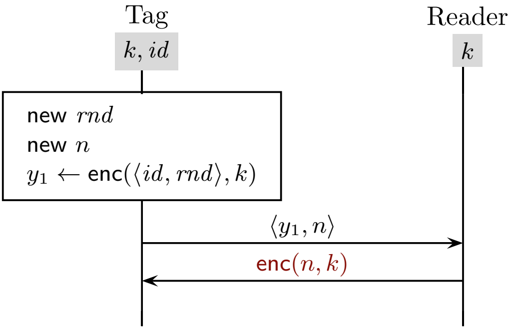
Messing with messages & C_{data}

C_{data}: "Messages are without relations"



Messing with messages & C_{data}

C_{data}: "Messages are without relations"

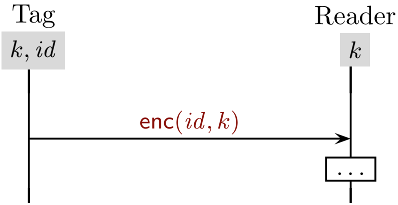


Practical examples (RFID protocols): HB^+, DM, KCL, LBV, LD, ...

Messing with messages & C_{data}

C_{data}: “Messages are without relations”

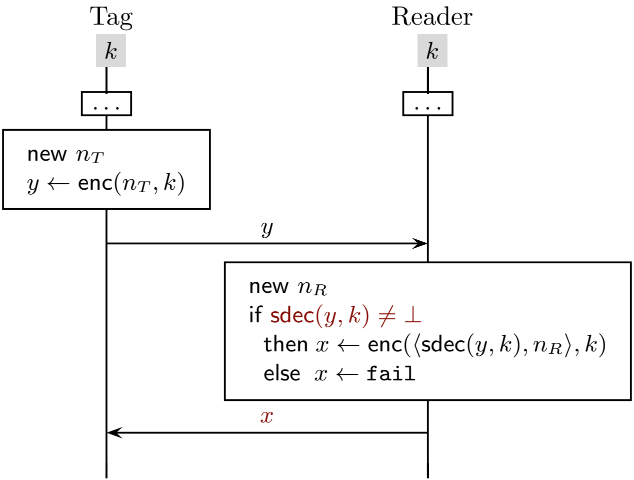
- Goal: messages do not leak info about involved agents
- Intuitively: outputs are (statically) indistiguishable from $\neq$ nonces



$$\{\text{enc}(\text{ok}, k), \text{enc}(\text{ok}, k)\} \not\sim \{n_1^f, n_2^f\}$$

Messing with conditionals & C_{test}

C_{test}: "Conditionals hold only for honest interactions"



Practical examples: BAC (ePassport), some versions of PACE (new version of ePassport), LAK, CH

Messing with conditionals & C_{test}

C_{test}: "Conditionals hold only for honest interactions"

- Goal: conditionals do not leak info about involved agents
- Intuitively: if Tag goes to a Then branch then the attacker just forwarded messages between this Tag and some Reader

A taste of $C_{\text{data}} \& C_{\text{test}}$

	Equivalence?	Active Attacker?
UK/ANO	☑	☑

A taste of C_{data}&C_{test}

	Equivalence?	Active Attacker?
UK/ANO	☑	☑

⇑ **Theorem: implies** ⇑

C_{data}	"Messages are without relations"
C_{test}	"Conditionals hold only for honest interactions"

A taste of C_{data} & C_{test}

	Equivalence?	Active Attacker?
UK/ANO	☑	☑

⇑ **Theorem: implies** ⇑

	Equivalence?	Active Attacker?
C_{data}	☑	☐
C_{test}	☐	☑

A taste of C_{data} & C_{test}

	Equivalence?	Active Attacker?
UK/ANO	☑	☑

⇑ **Theorem: implies** ⇑

	Equivalence?	Active Attacker?
C_{data}	☑	☐
C_{test}	☐	☑

⇑ can be **checked** ⇑

- C_{data}: automatic check of diff-equivalence using Proverif
- C_{test}: automatic check of correspondence prop. using Proverif

Applications

We wrote a tool on top of ProVerif that automatically checks our two sufficient conditions

New proofs of Unlinkability & Anonymity for:

- BAC+PA+AA (ePassport);
- PACE+PA+AA (ePassport v2);
- (fixed) LAK (RFID auth.);
- Hash-Lock (RFID auth.).

Applications

We wrote a tool on top of ProVerif that automatically checks our two sufficient conditions

New proofs of Unlinkability & Anonymity for:

- BAC+PA+AA (ePassport);
- PACE+PA+AA (ePassport v2);
- (fixed) LAK (RFID auth.);
- Hash-Lock (RFID auth.).

When conditions fail to hold: no direct attacks but still...

Flaws/attacks discovered:

- some versions of PACE ($\neg$ UK);
- LAK ($\neg$ UK).

... still looking for other case studies ...

Thank You!