Verifying Privacy by Little Interaction and No Process Equivalence

Denis Butin and Giampaolo Bella

Introduction

- E-voting protocols increasingly used
- Key property: voter privacy / ballot secrecy
- Inductive Method: protocol verification through theorem proving
- Extended for e-voting privacy analysis
- Example: FOO'92

Extensions for E-voting Protocols — Motivation

- Analysis of e-voting dominated by ProVerif automatic verifier
- Powerful, but sometimes limited
- Motivation to fill in the gaps with complementary, alternative approach

Privacy in e-voting

- aaa

Related Work

- Ryan / Kremer / Delaune: applied pi calculus, partially mechanized through ProVerif
- Observational equivalence: traces in which two voters swap their votes are equivalent in a sense
- Parts of the proof done by hand

Method: the Inductive approach

- Mathematical induction on protocol steps
- Dolev-Yao threat model
- Tool support: Isabelle/HOL interactive theorem prover

Protocols Verified in Isabelle So Far

Protocol	**Class**	**Year**	**Author(s)**
Yahalom	Key sharing, authentication	1996	Paulson
NS symmetric	Key sharing	1996	Paulson & Bella
Otway-Rees (with variants)	Authentication	1996	Paulson
Woo-Lam	Authentication	1996	Paulson
Otway-Bull	Authentication	1996	Paulson
NS asymmetric	Authentication	1997	Paulson
TLS	Multiple	1997	Paulson
Kerberos IV	Mutual authentication	1998	Bella
Kerberos BAN	Mutual authentication	1998	Paulson & Bella
SET suite	Multiple	2000+	Bella *et al.*
Abadi *et al.* certified e-mail	Accountability	2003	Bella *et al.*
Shoup-Rubin smartcard	Key distribution	2003	Bella
Zhou-Gollmann	Non-repudiation	2003	Paulson & Bella
Kerberos V	Mutual authentication	2007	Bella
TESLA	Broadcast authentication	2009	Schaller *et al.*
Meadows distance bounding	Physical	2009	Basin *et al.*
Multicast NS symmetric	Key sharing	2011	Martina
Franklin-Reiter	Byzantine	2011	Martina
Onion routing	Anonymising	2011	Li & Pang

E-voting Protocols

- New properties : privacy, verifiability, coercion-resistance. . .
- Partially studied with applied pi calculus, but with little mechanisation
- Often require modelling new crypto primitives

E-voting protocols: properties

- Eligibility
- Fairness
- Privacy / Receipt freeness / Coercion resistance – linkability concept (hard)
- Individual / Universal verifiability

The FOO Protocol

- Fujioka, Okamoto and Ohta, 1992
- Two election officials, bit commitment, blind signatures
- Signed, blinded commitment on a vote
- 6 steps

Specifying Blind Signatures

- Directly in `Message.thy` — limitation of operators interplay
- Solution: as part of inductive model

$$\llbracket \mathit{evsb} \in \mathit{foo};\ \mathit{Crypt}\ (\mathit{priSK}\ V)\ \mathit{BSBody} \in \mathit{analz}\ (\mathit{spies}\ \mathit{evsb});$$
$$\mathit{BSBody} = \mathit{Crypt}\ b\ (\mathit{Crypt}\ c\ (\mathit{Nonce}\ N));\ b \in \mathit{symKeys};$$
$$\mathit{Key}\ b \in \mathit{analz}\ (\mathit{spies}\ \mathit{evsb})\rrbracket$$
$$\Longrightarrow \mathit{Notes}\ \mathit{Spy}\ (\mathit{Crypt}\ (\mathit{priSK}\ V)\ (\mathit{Crypt}\ c\ (\mathit{Nonce}\ N)))\ \#\ \mathit{evsb} \in \mathit{foo}$$

What Is Privacy in E-Voting?

- Crucial point: privacy is NOT confidentiality of vote...
- ...But unlinkability of voter and vote
- In Pro-Verif, done with observational equivalence between swapped votes

Privacy in the Inductive Method: *aanalz*

```
primrec aanalz :: "agent => event list => msg set set"
where
  aanalz_Nil:   "aanalz A [] = {}"
| aanalz_Cons:
"aanalz A (ev # evs) =
 (if A = Spy then
  (case ev of
    Says A' B X ⇒
     (if A' ∈ bad  then aanalz Spy evs
      else if isAnms X
           then insert                ({Agent B} ∪ (analzplus {X} (analz(knows Spy evs)))) (aanalz Spy evs)
           else insert ({Agent A'} Un {Agent B} ∪ (analzplus {X} (analz(knows Spy evs)))) (aanalz Spy evs)
 )
  | Gets A' X ⇒ aanalz Spy evs
  | Notes A' X ⇒ aanalz Spy evs)
  else aanalz A evs)"
```

Extract associations from honest agent's messages

Privacy in the Inductive Method: *asynth*

inductive_set
asynth :: *msg set set* ⇒ *msg set set*
for as :: *msg set set where*
asynth_Build [intro]:
$[\![a1 \in as;\ a2 \in as;\ m \in a1;\ m \in a2;\ m \neq Agent\ Adm;\ m \neq Agent\ Col]\!]$
$\Longrightarrow a1 \cup a2 \in asynth\ as$

Build up association sets from associations with common elements. Only pairwise so far!

Privacy in the Inductive Method: Theorem Statement

$$
\begin{array}{l}
\textit{theorem foo\_V\_privacy\_asynth:} \\
[\![\textit{Says } V \textit{ Adm } \{\!|\textit{Agent } V, \\
\qquad\qquad\qquad \textit{Crypt (priSK } V\textit{) (Crypt } b \textit{ (Crypt } c \textit{ (Nonce Nv)))}|\!\} \in \textit{set evs;} \\
\ a \in \textit{(asynth (aanalz Spy evs));} \\
\ \textit{Nonce Nv} \in a\textit{; } V \notin \textit{bad; } V \neq \textit{Adm; } V \neq \textit{Col; evs} \in \textit{foo}]\!] \\
\ \Longrightarrow \textit{Agent } V \neq a
\end{array}
$$

If a regular voter started the protocol, the corresponding vote and identity are unlinkable.

Privacy in the Inductive Method: Proving Process

- Genericity of steps 2 and 4 yields proof complexity
- Genericity is natural consequence of respecting guarantee availability
- Strategy: map components in asynth to possible origins in aanalz
- Taxonomy of structures of elements in aanalz
- Divide & conquer

Conclusions

- Flexibility of Inductive Method confirmed. . .
- . . . but limitations related to message datatype extension
- Very different approach from most used tools (ProVerif, Scyther). . .
- . . . hence potential for complementarity!

Future Work

- Need stronger association synthesis — proof complexity challenge
- Analyse more recent e-voting protocols

Questions?