Inductive Analysis of Security Protocols in Isabelle/HOL with Applications to Electronic Voting

Denis Butin
Security Layers, Protocols and Formal Methods

Isabelle/HOL and the Inductive Method

Analysis of Composed Protocols

ISO/IEC 9798-3 and AIBS

Extensions for E-voting Protocols

Contributions & Perspectives
Introduction

- Network communication sensitive: banking, private correspondence, business-critical data
- Cryptography contributes to network security...
- ...But not sufficient in itself!
Security Layers

Several levels at which attacks can and have been led:

- Hardware (e.g. side-channel attacks)
- Cryptographic primitives
- Security protocols
- Ceremonies
Security Protocol Goals

- Classically: authentication, secret sharing, electronic payment...
- New, more complex needs: electronic voting, secure multiparty computation, electronic cash...
Analysing Security Protocols

Many methods:

- Model checking
- Automated / interactive theorem proving
- Static analysis, applied pi calculus, strand spaces...

Tools with automation: ProVerif, AVISPA, Scyther, AKiSs...
Interactive Theorem Proving

- Uses mathematical reasoning to determine if protocol reaches its security goals
- Unlike model checking, population unbounded
- Doesn’t provide explicit attack but may give clues
- Interactive
- Our choice — Isabelle
The Inductive Method

- Application of Isabelle ("generic proof assistant")! to security protocol verification
- ★ Paulson 1996, then Bella
- Uses mathematical induction to model and verify protocols + goals
Principles of the Inductive Method

- Unbounded number of agents
- Dedicated datatypes (keys, hashes, nonces...)
- Events for message sending, reception, agent knowledge
- Inductive reasoning over network event lists (traces)
- Cryptographic algorithms idealised
Threat Model

- Attacker = “Spy”
- Controls network (Dolev-Yao)
- Eavesdropping + dynamic behaviour, can also act like normal agent
Goal Definition and Proving

- Protocol security goals $\iff$ predicates over all possible traces
- User specifies techniques to use: basic induction, rewriting, automatic prover...
- In most cases, several subgoals generated and user input required again
Modelling Properties — Example

Authentication of an agent:

\[\left[ A \notin \text{bad}; B \notin \text{bad}; \text{evs} \in \text{ns}_{-}\text{public}\right] \implies \]

Crypt (pubEK A) \{Nonce NA, Nonce NB, Agent B\} \in parts (spies evs) \implies

Says A B (Crypt (pubEK B) \{Nonce NA, Agent A\}) \in set evs \implies

Says B A (Crypt (pubEK A) \{Nonce NA, Nonce NB, Agent B\}) \in set evs
## Protocols Verified in Isabelle So Far

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

- Adapt Isabelle theory framework (specifications of messages, events, keys, knowledge...)
- Model protocol steps
- Formalise novel guarantees: sometimes hardest step
- Proofs (interactive)
Analysing Composed Protocols

- Typical real-world scenario of security protocol use
- Analysis issue not solved in general, partially supported by Scyther
- Not done before in the Inductive Method
Protocol Composition Paradigm

- Certificate distribution sequenced with authentication
- Specified by two linked inductive models
- Better guarantee availability (implicit public key binding)
Protocol Composition — Discussion

- Scalable semantics, not limited to two protocols
- No compositionality theorem as for Scyther
- Case study extendable to detailed PKI
Auditable Identity-Based Signatures

- Proposed by David Gray in 2007
- Provide stronger non-repudiation than “standard” IBS (mitigate key escrow)
- Separate audit step allows third party to ensure signature origin
- Relies on additional audit key-pair; private one required to sign and registered with KGC
2010 Amendment presents new authentication protocols
We study *Five-pass mutual authentication with TTP, initiated by A*
Side-by-side specification of IBS and AIBS versions
Focus is not on the protocol itself but on AIBS
Auditable Identity-Based Signatures – Theories
Key package datatype: \( \textit{datatype pack} = \text{Pack key key} \)

Auditable signature structure:

\[
\text{Crypt (priSK A) \{ Crypt (priEK A) M, M \}}
\]

Can only sign with key package + private key:

\[
\begin{align*}
& [ \text{evss} \in \text{iso}; X \in \text{synth(analz (spies evss))}; \\
& \text{Key (priEK A) } \in \text{analz (spies evss)}; \\
& \text{Pkg (KP A B) } \in \text{analz (spies evss)} ] \\
\implies & \text{Notes Spy Crypt (priSK B) \{ Crypt (priEK A) X, X \} } \not\in \text{evss } \in \text{iso}
\end{align*}
\]
Auditable Identity-Based Signatures – Modelling

- candidates function — input agent name, output set of potential signers who leave a trace
- Classic authentication results + focus on signatures
- Comparative analysis shows operational auditable feature of AIBS
Extensions for E-voting Protocols — Introduction

- E-voting use is spreading quickly in the EU and elsewhere
- Sensitive, need for formal guarantees
- Inductive Method: protocol verification through theorem proving + mathematical induction
- Toolbox built with FOO as example protocol
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
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
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

\[
\begin{align*}
&\exists evsb \in \text{foo}; \text{Crypt} (\text{priSK} V) \ \text{BSBody} \in \text{analz} (\text{spies evsb}); \\
&\text{BSBody} = \text{Crypt} b (\text{Crypt} c (\text{Nonce} N)); \ b \in \text{symKeys}; \\
&\text{Key} b \in \text{analz} (\text{spies evsb})
\end{align*}
\]

\[\rightarrow \text{Notes Spy (Crypt (priSK V) (Crypt c (Nonce N)))} \neq evsb \in \text{foo}\]
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: \textit{aanalz}

\begin{verbatim}
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 isAnns X
                then insert ((\{Agent B\} ∪ (aanlzplus \{X\} (aanlz(knows Spy evs)))) (aanlz Spy evs)
                else insert (\{Agent A'\} Un \{Agent B\} ∪ (aanlzplus \{X\} (aanlz(knows Spy evs)))) (aanlz Spy evs)
            )
        |
        Gets A' X ⇒ aanalz Spy evs
        |
        Notes A' X ⇒ aanalz Spy evs)
      else aanalz A evs)"
\end{verbatim}

Extract associations from honest agent’s messages
Privacy in the Inductive Method: \texttt{asynth}

\begin{verbatim}
\texttt{inductive_set}

\texttt{asynth :: msg set set \Rightarrow msg set set}

\texttt{for as :: msg set set where}

\texttt{asynth\_Build [intro]:}

\[
\exists a1 \in as; a2 \in as; m \in a1; m \in a2; m \neq \text{Agent Adm}; m \neq \text{Agent Col} \]

\[
\implies a1 \cup a2 \in \text{asynth as}
\]

Build up association sets from associations with common elements. Only pairwise so far!
\end{verbatim}
Privacy in the Inductive Method: Theorem Statement

\textbf{Theorem foo\_V\_privacy\_asynth:}

\[ \begin{align*}
\{ \text{Says } V \text{ Adm} \} \{ \text{Agent } V, \\
\text{Crypt (priSK } V \) (\text{Crypt } b \text{ (Crypt } c \text{ (Nonce } Nv))\} \in \text{set evs}; \\
a \in (\text{asynth (aanalz Spy evs))}; \\
\text{Nonce } Nv \in a; \ V \notin \text{bad}; \ V \neq \text{Adm}; \ V \neq \text{Col}; \ evs \in \text{foo}\} \\
\implies \text{Agent } V \neq a
\end{align*} \]

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 synth to possible origins in analz
- Taxonomy of structures of elements in analz
- Divide & conquer
Contributions

- Conference publications:

- Workshop talk:
  - *Electronic Voting Protocol Analysis with the Inductive Method* — 2011 miniWorkshop on Security Frameworks (mWSF11)
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

- Focus on the e-voting part of the work
- Need stronger association synthesis — proof complexity challenge
- Analyse more recent e-voting protocols
- Article on AIBS chapter
- Long-term goal: reengineer message datatype completely for broader primitive support