Analysis of electronic voting protocols in applied pi calculus

Mark Ryan
University of Birmingham

based on joint work with

Ben Smyth
Steve Kremer
Mounira Kourjieh

IFIP WG 1.3, Udine, Italy
September 2009

Outline

- Electronic voting
- Applied pi calculus
- Privacy properties and verifiability properties
- Case studies

Voting system: desired properties

- Eligibility: only legitimate voters can vote, and at most once (This also implies that the voting authorities cannot insert votes)
- Fairness: no early results can be obtained
- Privacy: the fact that a particular voter in a particular way is not revealed to anyone
- △ Receipt-freeness: a voter cannot later prove to a coercer that she voted in a certain way
- Coercion-resistance: a voter cannot interactively cooperate with a coercer to prove that she voted in a certain way
- △ Individual verifiability: a voter can verify that her vote was really counted
- Universal verifiability: a voter can verify that the published outcome really is the sum of all the votes

. . . and all this even in the presence of corrupt election authorities!

Electronic voting: current situation

Country	*Status*
UK	

Electronic voting: current situation

Country	*Status*
UK	Worrying

Electronic voting: current situation

Country	*Status*
UK	Worrying
Germany	

Electronic voting: current situation

Country	*Status*
UK	Worrying
Germany	Abandoned

Electronic voting: current situation

Country	*Status*
UK	Worrying
Germany	Abandoned
Netherlands	

Electronic voting: current situation

Country	*Status*
UK	Worrying
Germany	Abandoned
Netherlands	Abandoned

Electronic voting: current situation

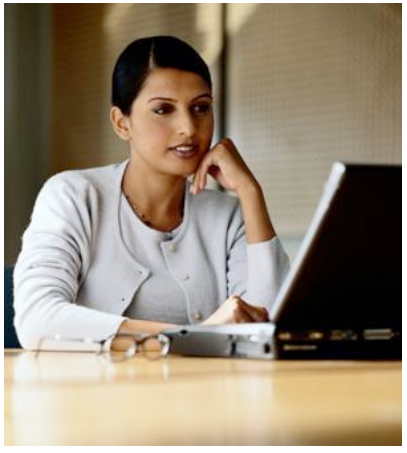
Country	*Status*
UK	Worrying
Germany	Abandoned
Netherlands	Abandoned
USA	

Electronic voting: current situation

Country	*Status*
UK	Worrying
Germany	Abandoned
Netherlands	Abandoned
USA	Disaster

How could it be secure?

Security by trusted client software



$\longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow$

- trusted by user
- does not need to be trusted by authorities or other voters

- not trusted by user
- doesn't need to be trusted by anyone

The applied π-calculus

Applied pi-calculus: [Abadi & Fournet, 01]

basic programming language with constructs for concurrency and communication

- based on the π-calculus [Milner *et al.*, 92]
- in some ways similar to the spi-calculus [Abadi & Gordon, 98], but more general w.r.t. cryptography

Advantages:

- naturally models a Dolev-Yao attacker
- allows us to model less classical cryptographic primitives
- both reachability-bases and equivalence-based specification of properties
- automated proofs using ProVerif tool [Blanchet]
- powerful proof techniques for hand proofs
- successfully used to analyze a variety of security protocols

Equations to model the cryptography: examples

1. *Encryption and signatures*

```
   decrypt( encrypt(m,pk(k)), k )   =   m
checksign( sign(m,k), m, pk(k) )   =   ok
```

2. *Blind signatures*

```
unblind( sign( blind(m,r), sk ), r )   =   sign(m,sk)
```

3. *Designated verifier proof of re-encryption*
   The term dvp(x,renc(x,r),r,pkv) represents a proof designated for the owner of pkv that x and renc(x,r) have the same plaintext.

```
checkdvp(dvp(x,renc(x,r),r,pkv),x,renc(x,r),pkv) = ok
       checkdvp( dvp(x,y,z,skv), x, y, pk(skv) ) = ok.
```

4. *Zero-knowledge proofs of knowledge*
   pf(k,x,y) represents proof that I know k such that dec(x,k)=y.

```
checkpf( pf(k,x,dec(x,k)), x, dec(x,k) ) = ok.
```

Applied pi calculus: Grammar [Abadi/Fournet 02]

$L, M, N, T, U, V ::=$	terms
$a, b, c, k, m, n, s, t, r, \ldots$	name
x, y, z	variable
$g(M_1, \ldots, M_l)$	function

$P, Q, R ::=$	processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu\, n.P$	name restriction
$u(x).P$	message input
$\overline{u}\langle M\rangle.P$	message output
$\text{if } M = N \text{ then } P \text{ else } Q$	conditional

$A, B, C ::=$	extended processes
P	plain process
$A \mid B$	parallel composition
$\nu\, n.A$	name restriction
$\nu\, x.A$	variable restriction
$\{^M/_x\}$	active substitution

Applied pi calculus: Operational semantics I [Abadi/Fournet 02]

$$
\begin{array}{lrcl}
\textsc{Par-0} & A & \equiv & A \mid 0 \\
\textsc{Par-A} & A \mid (B \mid C) & \equiv & (A \mid B) \mid C \\
\textsc{Par-C} & A \mid B & \equiv & B \mid A \\
\textsc{Repl} & !P & \equiv & P \mid !P \\
\\
\textsc{New-0} & \nu\, n.0 & \equiv & 0 \\
\textsc{New-C} & \nu\, u.\nu\, w.A & \equiv & \nu\, w.\nu\, u.A \\
\textsc{New-Par} & A \mid \nu\, u.B & \equiv & \nu\, u.(A \mid B) \\
 & & & \text{where } u \notin \mathrm{fv}(A) \cup \mathrm{fn}(A) \\
\\
\textsc{Alias} & \nu\, x.\{^{M}/_{x}\} & \equiv & 0 \\
\textsc{Subst} & \{^{M}/_{x}\} \mid A & \equiv & \{^{M}/_{x}\} \mid A\{^{M}/_{x}\} \\
\textsc{Rewrite} & \{^{M}/_{x}\} & \equiv & \{^{N}/_{x}\} \\
 & & & \text{where } M =_E N
\end{array}
$$

$$
\begin{array}{ll}
\textsc{Comm} & \overline{c}\langle x\rangle.P \mid c(x).Q \to P \mid Q \\
\textsc{Then} & \text{if } N = N \text{ then } P \text{ else } Q \to P \\
\textsc{Else} & \text{if } L = M \text{ then } P \text{ else } Q \to Q \\
 & \quad \text{for ground terms } L, M \text{ where } L \neq_E M
\end{array}
$$

Applied pi calculus: Operational semantics II [Abadi/Fournet 02]

IN
$$c(x).P \xrightarrow{c(M)} P\{^{M}/_{x}\}$$

OUT-ATOM
$$\overline{c}\langle u\rangle.P \xrightarrow{\overline{c}\langle u\rangle} P$$

OPEN-ATOM
$$\frac{A \xrightarrow{\overline{c}\langle u\rangle} A' \qquad u \neq c}{\nu\, u.A \xrightarrow{\nu\, u.\overline{c}\langle u\rangle} A'}$$

SCOPE
$$\frac{A \xrightarrow{\alpha} A' \qquad u \text{ does not occur in } \alpha}{\nu\, u.A \xrightarrow{\alpha} \nu\, u.A'}$$

PAR
$$\frac{A \xrightarrow{\alpha} A' \qquad \mathrm{bv}(\alpha) \cap \mathrm{fv}(B) = \mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset}{A \mid B \xrightarrow{\alpha} A' \mid B}$$

STRUCT
$$\frac{A \equiv B \qquad B \xrightarrow{\alpha} B' \qquad B' \equiv A'}{A \xrightarrow{\alpha} A'}$$

Receipt-freeness

Receipt-freeness: leaking secrets to the coercer

To model receipt-freeness we need to specify that a coerced voter cooperates with the coercer by leaking secrets on a channel ch

$P ::=$
0
$P \mid P$
$\nu n.P$
$\mathsf{in}(u, x).P$
$\mathsf{out}(u, M).P$
$\mathsf{if}\ M = N\ \mathsf{then}\ P\ \mathsf{else}\ P$
$!P$
$\ldots$

P^{ch} in terms of P

- $0^{ch} = 0$
- $(P \mid Q)^{ch} = P^{ch} \mid Q^{ch}$
- $(\nu n.P)^{ch} = \nu n.\mathsf{out}(ch, n).P^{ch}$
- $(\mathsf{in}(u, x).P)^{ch} = \mathsf{in}(u, x).\mathsf{out}(ch, x).P^{ch}$
- $(\mathsf{out}(u, M).P)^{ch} = \mathsf{out}(u, M).P^{ch}$
- $\ldots$

We denote by $P^{\backslash \mathsf{out}(chc,\cdot)}$ the process $\nu chc.(P \mid !\mathsf{in}(chc, x))$.

Lemma: $(P^{ch})^{\backslash \mathsf{out}(chc,\cdot)} \approx_\ell P$

Receipt-freeness: definition

Intuition

There exists a process V' which

- votes a,
- leaks (possibly fake) secrets to the coercer,
- looks indistinguishable to coercer from situation in which she voted c

Definition (Receipt-freeness)

A voting protocol is receipt-free if there exists a process V', satisfying

- $V'\backslash^{out(chc,\cdot)} \approx_\ell V_A\{^a/_v\}$,
- $S[V_A\{^c/_v\}^{chc} \mid V_B\{^a/_v\}] \approx_\ell S[V' \mid V_B\{^c/_v\}]$.

Case study: Lee *et al.* protocol
We prove receipt-freeness by

- exhibiting V'
- showing that $V'\backslash^{out(chc,\cdot)} \approx_\ell V_A\{^a/_v\}$
- showing that $S[V_A\{^c/_v\}^{chc} \mid V_B\{^a/_v\}] \approx_\ell S[V' \mid V_B\{^c/_v\}]$

end-to-end verifiability

- Election results can be fully verified by voters/observers
- The software provided by election authorities does not need to be trusted
- The software used to perform the verification can be sourced independently


Where is my
VOTE?
GETTY IMAGES


Démocratie
& Liberté
WHERE
IS
MY VOTE


Where is
my vote ?


WHERE
IS MY
VOTE?

Election verifiability

Individual verifiability

A voter can check her own vote is included in the tally.

Universal verifiability

Anyone can check that the declared outcome corresponds to the tally.

Eligibility verifiability

Anyone can check that only eligible votes are included in the declared outcome.

Remarks

- Verifiability $\neq$ correctness
- What system components need to be trusted in order to carry out these checks?

Individual verifiability

Intuition: a protocol satisfies individual verifiability if there is a test

$$R^{IV}(\text{my\_vote}, \text{my\_data}, \text{bb\_entry})$$

that a voter can apply after the election.

The test succeeds iff the bulletin board entry corresponds to the voter's vote and data.

Acceptability conditions for R^{IV}

- For all votes s, there is an execution of the protocol that produces $\tilde{M}$ such that some bulletin board entry T satisfies $R^{IV}(s, \tilde{M}, T)$.
- The bulletin board entry determines the vote, that is:

$$\forall s, t, \tilde{M}, \tilde{N}, T \ \left(R^{IV}(s, \tilde{M}, T) \wedge R^{IV}(t, \tilde{N}, T) \Rightarrow s = t \right)$$

Universal verifiability

Intuition: a protocol satisfies universal verifiability if there is a test

$$R^{UV}\big(\text{declared\_outcome}\,,\ \text{bb\_entries}\,,\ \text{proof}\big)$$

that an observer can apply after the election.

The test succeeds iff the declared outcome is correct w.r.t. the bb entries and the proof.

Acceptability conditions for R^{UV}

- $\tilde{T}$ determines $\tilde{s}$, that is,

$$R^{UV}(\tilde{s_1}, \tilde{T}, p_1) \wedge R^{UV}(\tilde{s_2}, \tilde{T}, p_2) \Rightarrow \tilde{s_1} = \tilde{s_2}$$

- The observer opens the bb entry the same way as the voter:

$$R^{IV}(s, \tilde{M}, T) \wedge R^{UV}(\tilde{s}, \tilde{T}, p') \Rightarrow \exists p'.\, R^{UV}(\tilde{s} \circ s, \tilde{T} \circ T, p')$$

Election verifiability

A voting process $C[!\nu\tilde{a}.(P \mid Q[\overline{c}\langle U\rangle])]$ satisfies *election verifiability* if voter's credentials and bulletin board entries are unique and there exists tests R^{IV}, R^{UV}, R^{EV} with

- $fv(R^{IV}) \subseteq bv(P) \cup \{v, z\}$
- $fv(R^{UV}) \subseteq \{v, z\}$
- $fv(R^{EV}) \subseteq \{y, z\}$
- $(fn(R^{UV}) \cup fn(R^{EV})) \cap bn(P) = \emptyset$

such that the augmented voting process satisfies the following conditions:

- the *un*reachability assertion: $\overline{\texttt{fail}}\langle\text{true}\rangle$.
- the reachability assertion: $\overline{\texttt{pass}}\langle\text{true}, x\rangle$.

Augmented process

Given a voting process $C[!\nu\tilde{a}.(P \mid Q[\overline{c}\langle U\rangle])]$ and tests R^{IV}, R^{UV}, R^{EV}, the *augmented voting process* is

$$\nu b.(C[!\nu\tilde{a}, b'.(\hat{P} \mid \hat{Q})] \mid R \mid R') \mid R'' \mid R'''$$

where

$$\begin{array}{lcl}
\hat{P} & = & b(v).P.c(z).b'(y).(\overline{\texttt{pass}}\langle R^{IV}, z\rangle \mid \overline{\texttt{fail}}\langle\psi\rangle) \\
\hat{Q} & = & Q[\overline{b'}\langle U\rangle \mid \overline{\mathcal{D}}\langle U\rangle \mid \overline{c}\langle U\rangle] \\
R & = & !\nu s.((!\overline{b}\langle s\rangle) \mid \overline{c}\langle s\rangle) \\
R' & = & b(v').b(v'').c(x').c(x'').c(y').c(y'').c(z').\overline{\texttt{fail}}\langle\phi' \vee \phi'' \vee \phi'''\rangle \\
R'' & = & \texttt{pass}(e).\texttt{pass}(e').\overline{\texttt{fail}}\langle e_1 \wedge e_1' \wedge (e_2 = e_2')\rangle \\
R''' & = & \mathcal{D}(e).\mathcal{D}(e').\overline{\texttt{fail}}\langle\neg(e = e')\rangle \\
 & & \\
\psi & = & (R^{IV} \wedge \neg R^{UV}) \vee (R^{IV} \wedge \neg R^{EV}) \vee (\neg R^{IV} \wedge R^{EV}) \\
\phi' & = & R^{IV}\{^{v',\tilde{x'},z'}/_{v,\tilde{x},z}\} \wedge R^{IV}\{^{v'',\tilde{x''},z'}/_{v,\tilde{x},z}\} \wedge \neg\,(v' = v'') \\
\phi'' & = & R^{UV}\{^{v',z'}/_{v,z}\} \wedge R^{UV}\{^{v'',z'}/_{v,z}\} \wedge \neg\,(v' = v'') \\
\phi''' & = & R^{EV}\{^{y',z'}/_{y,z}\} \wedge R^{EV}\{^{y'',z'}/_{y,z}\} \wedge \neg\,(y' =_E y'')
\end{array}$$

Results and trustworthiness requirements

Property	*FOO'92*	*Civitas '08*	*Helios/UCL '09*
Vote-privacy trusted compnts	✓ client	✓ client	✓ client
Receipt-freeness trusted compnts	×	✓ client	×
Coercion resist. trusted compnts	×	✓ client	×
Individual verif. trusted compnts	✓ client	✓ client	✓ client
Universal verif. trusted compnts	✓	✓	✓
Elig. verif. trusted compnts	×	✓	×

Conclusions and future work

Conclusions

- First *generic* formal definitions of election verifiability.
- Suitable for automation.
- Automatic verification for PostalBallot, FOO, Civitas.

Future work

- Completion of homomorphic cases (Helios/UCL)
- Voting systems that are not client-crypto-based.