Modelling and Verifying Electronic Voting Protocols in the Applied Pi Calculus

Mark Ryan
University of Birmingham

joint work with Steve Kremer and Stéphanie Delaune
LSV Cachan, France

IFIP WG 1.3, La Roche-en-Ardenne 4 June 2006

Outline

Electronic voting

- Promises convenient, efficient and secure facility for recording and tallying votes
- Suitable for variety of types of elections: from small committees or on-line communities through to full-scale national elections
- But carries risk of large-scale and undetectable fraud.
- Current situation in USA is far from ideal. . .
  [KohnoStubblefieldRubinWallach2004] analysed source code of electronic voting machines sold by the second largest and fastest-growing vendor, used in 37 US states

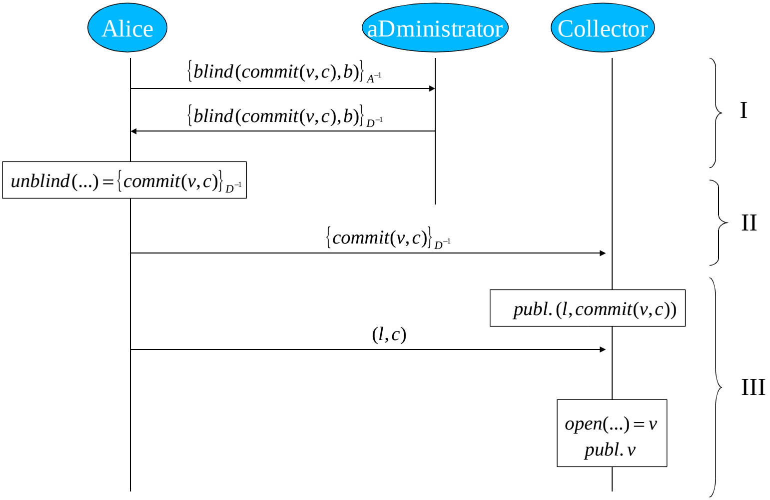
  > *"A 15-year-old in a garage could manufacture smart cards and sell them on the Internet that would allow for multiple votes"* Avi Rubin

- Formal protocols offer possibility of abstract analysis of the protocol against formally-stated properties

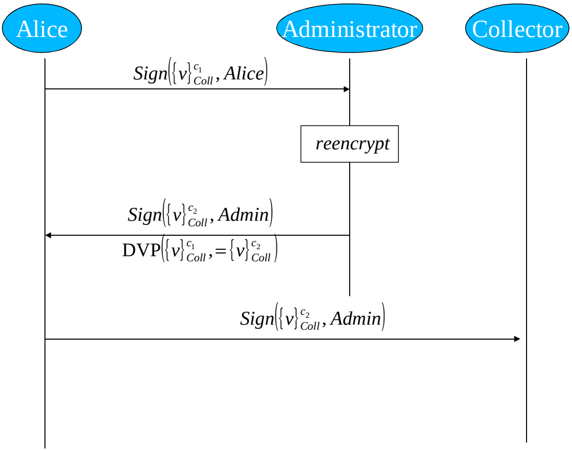
Expected properties

- Eligibility: only legitimate voters can vote, and only once
- Fairness no early results can be obtained which could influence the remaining voters
- Privacy: the fact that a particular voted 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

FOO 92 protocol [FujiokaOkamotoOhta92]



LBDKYY'03 protocol [LeeBoydDawsonKimYangYoo]



Some unusual cryptographic primitives

- Anonymous channels
  - implemented using mixnets, onion routing, . . .
- Commitment
  - To commit to m, I invent a new random r and send you $\text{commit}(m, r)$.
  - Later, I'll send you r, which you can use to reveal m.
  - it is binding: one cannot find some other r', such that the commitment opens correctly to some other m'
- Blind signatures
  - I want you to sign m but I don't want you to see its value.
  - I send you $\text{blind}(m, r)$. You sign it. I use r to extract your signature on m.
- Re-encryption
  - From $\{m\}_K^{c_1}$, compute $\{m\}_K^{c_2}$ without knowing m, c_1, c_2 or K.
- Designated verifier proofs of re-encryptions
  - Prove "$\text{decrypt}(\{m\}_{K_A}^{c_1}, K_A) = \text{decrypt}(\{m\}_{K_A}^{c_2}, K_A)$" in a way convincing only to owner of K_B.

Specification challenge

How to specify

- The protocol, which has that encryption stuff in it;
- The properties, such as
  - privacy
  - receipt-freeness
  - coercion-resistance

. . . in such a way that we can

- verify satisfaction of the properties by the protocol?

The applied π-calculus

- The applied pi calculus [AbadiFournet01] models concurrent processes which send and receive messages on channels. Channels may be private, or public (= accessible to the attacker). Message sending on channels is anonymous (if wanted, identification of the sender may be given as part of the message).
- Applied pi calculus is based on the pi calculus [Milner++92], and in some ways similar to the spi calculus [AbadiGordon98] but with definable crypto constructors instead of a limited set of builtins.

The applied π-calculus

- Messages are terms constructed from a signature. An equational theory is used to model cryptographic primitives, e.g.

$$\mathrm{decrypt}(\mathrm{encrypt}(m, \mathrm{pk}(sk)), sk) = m$$

- The behaviour of a process may depend on the environment, which is assumed to be controlled by the attacker. Process may expose terms, by writing to public channels. The attacker can apply functions to terms thus exposed, constructing new terms, modulo the equational theory.
- Thus, the attacker controls the public channels, and may read, intercept and inject messages on them. But the attacker can only apply functions (e.g., encrypt and decrypt) if he has the necessary arguments (e.g., the keys).

Signature and equational theory: FOO'92

Signature

`commit/2.`	*commitment*
`open/2.`	*open commitment*
`sign/2.`	*digital signature*
`checksign/2.`	*open digital signature*
`pk/1.`	*get public key from private key*
`blind/2.`	*blinding*
`unblind/2.`	*undo blinding*

Equational theory

```
             open(commit(m,r),r)  =  m.
      checksign(sign(m,sk),pk(sk))  =  m.
unblind(sign(blind(m,r),sk),r)  =  sign(m,sk).
```

Re-encryption and designated verifier proofs

- Re-encryption

```
  decrypt(pencrypt(m,pk(sk),r),sk)   =   m.
rencrypt(pencrypt(m,pk(sk),r1),r2)   =   pencrypt(m,pk(sk),f(r1,r2)).
```

- Designated verifier proofs of re-encryptions

  The term `dvp(x,rencrypt(x,r),r,pkv)` represents a proof designated for pkv that x and rencrypt(x,r) have the same plaintext.

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

Voter process: FOO'92

```
processV =
     new b; new r;
     let blindedcommitedvote=blind(commit(v,r),b) in
     out(ch,(hostv,sign(blindedcommitedvote,skv)));
     in(ch,m2);
     if checksign(m2,pka)=blindedcommitedvote then
     let signedcommitedvote=unblind(m2,b) in
     phase 1;
     out(ch,signedcommitedvote);
     in(ch,(l,=signedcommitedvote));
     phase 2;
     out(ch,(l,r)).
```

How to specify properties in the applied π-calculus

- Properties of protocols can be expressed as
  - reachability conditions
    e.g. is there an execution leading to a state in which a certain message is known to the attacker?

  and
  - observational equivalences,
    e.g. can the attacker distinguish two given runs of the system?

- Privacy and receipt-freeness will be expressed as observational equivalences.

Static equivalence

As a process evolves, it may expose the values of its variables to the environment. In applied pi, this is modelled as a "frame", e.g.

$$\nu n \{^{a}/_{x},\ ^{g(b)}/_{y},\ ^{f(c,n)}/_{z}\}$$

Static equivalence on frames ($\approx_s$) [passive adversary]

$\varphi \approx_s \psi$ when $dom(\varphi) = dom(\psi)$, and for all terms U, V, $(U = V)\varphi$ iff $(U = V)\psi$

Example: Suppose we have the equations

```
fst(pair(x,y))   =   x
snd(pair(x,y))   =   y
```

Then

$$\nu n\{^{f(n,a)}/_{x}\} \approx_s \nu n\{^{f(n,b)}/_{x}\}$$
$$\nu n\{^{pair(n,a)}/_{x}\} \not\approx_s \nu n\{^{pair(n,b)}/_{x}\}$$

because $snd(x)=a$ succeeds only on the left-hand side

Observational equivalence

Observational equivalence ($\approx$) [active adversary]

Largest symmetric relation R between closed extended processes with the same domain such that $A\ R\ B$ implies:

1. if $A \Downarrow a$ then $B \Downarrow a$ ($\Downarrow \equiv$ *"can send a message on"*)
2. if $A \rightarrow^* A'$ then $B \rightarrow^* B'$ and $A'\ R\ B'$ for some B'
3. $C[A]\ R\ C[B]$ for closing evaluation contexts C

$\Updownarrow$ [AbadiFournet2001]

Labeled bisimilarity ($\approx_\ell$)

labeled bisimilarity $\equiv$ usual bisimilarity $+ \approx_s$ at each step

Modelling properties: privacy

Privacy

VP satisfies privacy if

$$S[V_A\{^a/_v\} \mid V_B\{^b/_v\}] \quad \approx \quad S[V_A\{^b/_v\} \mid V_B\{^a/_v\}].$$

Results. FOO'92 and LBDKYY'03 satisfy privacy.

Modelling 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

We denote by V^{ch} the process built from the process V as follows:

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

We also define $V^{\setminus out(chc,\cdot)} \widehat{=} \nu chc.(V \mid !\mathrm{in}(chc, x))$.

Receipt-freeness

Definition (Receipt-freeness)

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

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

Intuitively, there exists a process V' which

- votes a,
- leaks (possibly fake) secrets to the coercer,
- and makes the coercer believe he voted c

Some results

Let VP be a voting protocol. We have formally shown that:

$$VP \text{ is receipt-free} \implies VP \text{ respects privacy.}$$

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

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

Interacting with the coercer

To model coercion-resistance, we need to model interaction between the coercer and the voter:

1. secrets are leaked to the coercer on a channel c_1, and
2. outputs are prepared by the coercer and given to the voter via c_2.

We denote by V^{c_1,c_2} the process built from V as follows:

- $0^{c_1,c_2} \mathrel{\widehat{=}} 0$,
- $(P \mid Q)^{c_1,c_2} \mathrel{\widehat{=}} P^{c_1,c_2} \mid Q^{c_1,c_2}$,
- $(\nu n.P)^{c_1,c_2} \mathrel{\widehat{=}} \nu n.\mathrm{out}(c_1, n).P^{c_1,c_2}$,
- $(\mathrm{in}(u,x).P)^{c_1,c_2} \mathrel{\widehat{=}} \mathrm{in}(u,x).\mathrm{out}(c_1,x).P^{c_1,c_2}$,
- $(\mathrm{out}(u,M).P)^{c_1,c_2} \mathrel{\widehat{=}} \mathrm{in}(c_2,x).\mathrm{out}(u,x).P^{c_1,c_2}$ (x is a fresh variable),
- ...

Coercion-resistance (1)

First approximation:

VP is coercion-resistant if there exists a process V' such that

$$S[V_A\{^c/_v\}^{c_1,c_2} \mid V_B\{^a/_v\}] \approx_\ell S[V' \mid V_B\{^c/_v\}].$$

Problem:

- the coercer could oblige $V_A\{^c/_v\}^{c_1,c_2}$ to vote $c' \neq c$,
- the process $V_B\{^c/_v\}$ would not counterbalance the outcome

Solution:

$\hookrightarrow$ a new relation we have called adaptive simulation (A $\preceq_a$ B)

Coercion-resistance (2)

Definition (Coercion-resistance)

A voting protocol is coercion-resistant if there exists a process V' and an evaluation context C satisfying

- $S[V_A\{^c/_v\}^{c_1,c_2} \mid V_B\{^a/_v\}] \preceq_a S[V' \mid V_B\{^x/_v\}]$,
- $\nu c_1, c_2.C[V_A\{^c/_v\}^{c_1,c_2}] \approx_\ell V_A\{^c/_v\}^{chc}$,
- $\nu c_1, c_2.C[V']^{\backslash out(chc,\cdot)} \approx_\ell V_A\{^a/_v\}$,

where x is a fresh free variable.

Intuitively,

- $V_B\{^x/_v\}$ can adapt his vote and counter-balance the outcome,
- we require that when we apply a context C (the coercer requesting $V_A\{^c/_v\}^{c_1,c_2}$ to vote c) the process V' in the same context C votes a.

Some results

Let *VP* be a voting protocol. We have formally shown that:

VP is coercion-resistant $\Longrightarrow$ *VP* respects receipt-free.

$\hookrightarrow$ reflects the intuition but the proof is technical

Case study: Lee *et al.* protocol
Coercion-resistance depends on implementation details:

- encryption with integrity check
  $\hookrightarrow$ fault attack: the protocol is not coercion-resistant
- encryption without integrity check
  $\hookrightarrow$ the protocol is coercion-resistant

Conclusion and future work

Conclusion:

- first formal definitions of receipt-freeness and coercion-resistance
- coercion-resistance $\Rightarrow$ receipt-freeness $\Rightarrow$ privacy,
- a case study giving interesting insights

Future work:

- decision procedure for observational equivalence for processes without replication
- other properties based on *not being able to prove*
- individual/universal verifiability