Formalizing and Proving Privacy Properties of Voting Protocols using Alpha-Beta Privacy

Formal methods and security protocol

Sébastien Gondron and Sebastian A. Mödersheim

DTU Compute
Danmarks Tekniske Universitet

Mai 23, 2019

Outline

1. Running example: FOO'92
2. Defining Privacy Goals
3. Alpha-Beta Privacy
4. FOO'92 in Alpha-Beta Privacy
5. Stronger Privacy Properties
6. Conclusion

Introduction

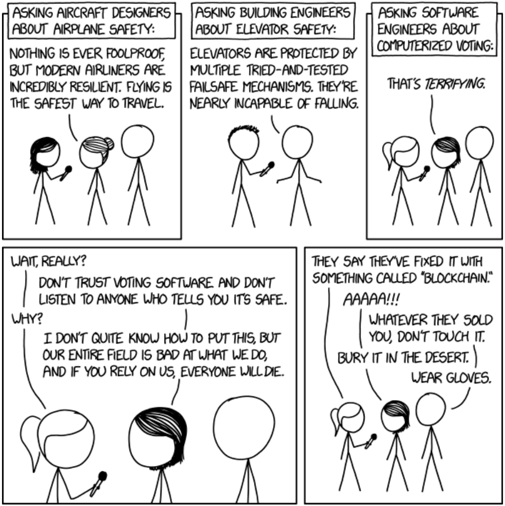
ASKING AIRCRAFT DESIGNERS ABOUT AIRPLANE SAFETY:
NOTHING IS EVER FOOLPROOF, BUT MODERN AIRLINERS ARE INCREDIBLY RESILIENT. FLYING IS THE SAFEST WAY TO TRAVEL.
ASKING BUILDING ENGINEERS ABOUT ELEVATOR SAFETY:
ELEVATORS ARE PROTECTED BY MULTIPLE TRIED-AND-TESTED FAILSAFE MECHANISMS. THEY'RE NEARLY INCAPABLE OF FALLING.
ASKING SOFTWARE ENGINEERS ABOUT COMPUTERIZED VOTING:
THAT'S TERRIFYING.
WAIT, REALLY?
DON'T TRUST VOTING SOFTWARE AND DON'T LISTEN TO ANYONE WHO TELLS YOU IT'S SAFE.
WHY?
I DON'T QUITE KNOW HOW TO PUT THIS, BUT OUR ENTIRE FIELD IS BAD AT WHAT WE DO, AND IF YOU RELY ON US, EVERYONE WILL DIE.
THEY SAY THEY'VE FIXED IT WITH SOMETHING CALLED "BLOCKCHAIN."
AAAAA!!!
WHATEVER THEY SOLD YOU, DON'T TOUCH IT.
BURY IT IN THE DESERT.
WEAR GLOVES.

Figure: xkcd - Voting Software

FOO'92 Protocol

Setup

- A population of voters $V_1, \ldots, V_N$.
- Each voter V_i has decided his or her vote $v_i \in \{0, 1\}$.
- r_i (and later b_i) are secret random numbers chosen by voter V_i.
- There is an administration A that controls who is a valid voter and issues the ballots.
- There is a counter C who collects all the ballots. C then publishes all ballots in a random order.
- We assume anonymous channels similar to onion routing like TOR, and write $[A]\bullet\!\!\rightsquigarrow\!\!\bullet B$ for
    - A has a secure channel with B, but with respect to a pseudonym of A, so B does not know A but can send a reply that only A receives.
    - The intruder cannot observe that A and B have communicated.

Cryptographic Primitives

- Blind signatures: *m* is a message and *b* is a blinding factor
  - *unblind*(*blind*(m,b),b) = *blind*(m,b)
  - *sign*(*priv*(A),*blind*(m,b))
  - *unblind*(*sign*(*priv*(A),*blind*(m,b)),b) = *sign*(*priv*(A),m)
- Bit-commitments: *v* is a message (a vote) and *r* is a randomization value
  - *open*(*commit*(v,r),r) = v

Protocol Narration

Phase 1	
$[V_i] \bullet\!\!\leadsto\!\!\bullet A$	$: sign(priv(V_i), blind(commit(v_i, r_i), b_i))$
$A \bullet\!\!\leadsto\!\!\bullet [V_i]$	$: sign(priv(A), blind(commit(v_i, r_i), b_i))$
Phase 2	
$[V_i] \bullet\!\!\leadsto\!\!\bullet C$	$: sign(priv(A), commit(v_i, r_i))$
$C \rightarrow$ all	$: sign(priv(A), commit(v_{\pi(j)}, r_{\pi(j)}))$ for each $j \in \{1, \ldots, N\}$
Phase 3	
$[V_i] \bullet\!\!\leadsto\!\!\bullet C$	$: r_i$
$C \rightarrow$ all	$: r_{\pi(j)}$ for each $j \in \{1, \ldots, N\}$

Table: Protocol description for FOO'92 in a style of an AnB language

The Goals

The Original Goals

From the paper [FOO'92]:

- Completeness: All valid votes are counted correctly.
- Soudness: The dishonest voter cannot disrupt the voting.
- Privacy: All votes must be secret.
- Unreusability: No voter can vote twice.
- Eligibility: Nothing must affect the voting.
- Verifiability: No one can falsify the result of voting.

The Goals

Privacy

Figure: "2015 Election Ballot Counting" by City of Fort Collins, CO

Voting privacy: the number of votes and the result of the election are finally published. The intruder should not find out more than that about voters and votes.

Encoding of frames

Definition ($\phi_{gen}(D)$ and $\phi_{frame}(F)$)

For a frame $F = \{\!|m_1 \mapsto t_1, \ldots, m_l \mapsto t_l|\!\}$ with domain $D = \{m_1, \ldots, m_l\}$, a unary predicate *gen* and an interpreted unary function symbol kn_F, we define the Herbrand logic formulae:

$$\phi_{gen}(D) \equiv \forall r.gen(r) \iff$$
$$(r \in D \vee \bigvee_{f^n \in \Sigma_{op}} \exists r_1, \ldots, r_n.\, r = f(r_1, \ldots, r_n) \wedge gen(r_1) \wedge \ldots \wedge gen(r_n))$$

$$\phi_{frame}(F) \equiv kn_F[m_1] = t_1 \wedge \ldots \wedge kn_F[m_l] = t_l \wedge$$
$$\bigwedge_{f^n \in \Sigma_{op}} \forall r_1, \ldots, r_n \colon gen.\, kn_F[f(r_1, \ldots, r_n)] = f(kn_F[r_1], \ldots, kn_F[r_n])$$

Static equivalence of Frames

Definition (Static Equivalence of Frames)

A common approach is based on formulating pairs of worlds and the goal that look the same to the intruder, written $\sim$.

Static equivalence of Frames

Definition (Static Equivalence of Frames)

A common approach is based on formulating pairs of worlds and the goal that look the same to the intruder, written $\sim$.

We encode static equivalence of frames in Herbrand Logic:

Definition ($\phi_{\sim}(F_1, F_2)$)

Let F_1 and F_2 be frames with the same domain.

$$\phi_{\sim}(F_1, F_2) \equiv \forall r, s\colon gen.\, kn_{F_1}[r] = kn_{F_1}[s] \iff kn_{F_2}[r] = kn_{F_2}[s]$$

Static Equivalence of Frames

The structural information

The intruder knows the structure of the messages, i.e. the specification of the protocol is public:

$$\begin{aligned} struct = \{\!| & m_0 \mapsto pub(A), m_1 \mapsto pub(V_1), \ldots, m_n \mapsto pub(V_N), \\ & m_{N+1} \mapsto sign(priv(A), commit(v[\pi[1]], r[\pi[1]])), \ldots, \\ & m_{2N} \mapsto sign(priv(A), commit(v[\pi[N]], r[\pi[N]])), \\ & m_{2N+1} \mapsto r[\pi[1]], \ldots, m_{3N} \mapsto r[\pi[N]] |\!\} \end{aligned}$$

Static Equivalence of Frames

The concrete information

The intruder also knows the concrete messages that he observes:

$$\begin{aligned}
concr = \{\!| & m_0 \mapsto pub(A), m_1 \mapsto pub(V_1), \ldots, m_n \mapsto pub(V_N), \\
& m_{N+1} \mapsto sign(priv(A), commit(\theta_0(v_{\pi_0(1)}), r_{\pi_0(1)})), \ldots, \\
& m_{2N} \mapsto sign(priv(A), commit(\theta_0(v_{\pi_0(N)}), r_{\pi_0(N)})), \\
& m_{2N+1} \mapsto r_{\pi_0(1)}, \ldots, m_{3N} \mapsto r_{\pi_0(N)} |\!\}
\end{aligned}$$

Model-Theoretical Alpha-Beta Privacy

We specify two formulae:

- α the high-level we deliberately reveal to the intruder/verifier/public
- β the technical information like cryptographic messages that are observable (including α)

Model-Theoretical Alpha-Beta Privacy

We specify two formulae:

- α the high-level we deliberately reveal to the intruder/verifier/public
- β the technical information like cryptographic messages that are observable (including α)

Definition (Model-theoretical (α, β)-privacy)

We say that (α, β)*-privacy holds (model-theoretically)* iff every Σ_0-model of α can be extended to a Σ-model of β. Here a Σ-interpretation $\mathcal{I}'$ is an *extension* of a Σ_0-interpretation $\mathcal{I}$ if they agree on all variables and all the interpreted function and relation symbols of Σ_0.

Example

Formula $x_1, x_2 \in \{0,1\} \wedge x_1 + x_2 = 1$
What are the models? (values of x_1 x_2 that makes the formula true).
$\theta_0 = \{x_1 \mapsto 0, x_2 \mapsto 1\}$ and $\theta_1 = \{x_1 \mapsto 1, x_2 \mapsto 0\}$.

The two visions of the world

- $\theta_0 \models \alpha$: an interpretation of the v_i with $\{0, 1\}$ that is a model of α, i.e. the truve vote of every voter
- $\theta_I \models \alpha$: an arbitratry model called an *intruder's hypothesis*, i.e. that maps the v_i to $\{0, 1\}$ so that their sum is R

The two visions of the world

- $\theta_0 \models \alpha$: an interpretation of the v_i with $\{0,1\}$ that is a model of α, i.e. the truve vote of every voter
- $\theta_I \models \alpha$: an arbitratry model called an *intruder's hypothesis*, i.e. that maps the v_i to $\{0,1\}$ so that their sum is R

Thus we can find a permutation $\psi\colon \{1,\ldots,N\} \to \{1,\ldots,N\}$ such that $\theta_I(v_i) = \theta_0(v_{\psi(i)})$ for all $i \in \{1,\ldots,N\}$.

Example

Given three voters, i.e. $N = 3$ and the result of the vote is $R = 2$, the true result of the vote $\theta_0 = \{v_1 \mapsto 1, v_2 \mapsto 1, v_3 \mapsto 0\}$ and the actual permutation be $\pi_0 = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{smallmatrix}\right)$, the bulletin board is then:

Bulletin board	j	1	2	3
	$v_{\pi_0(j)}$	1	0	1

Let us consider an intruder's hypothesis $\theta_I = \{v_1 \mapsto 0, v_2 \mapsto 1, v_3 \mapsto 1\}$. One possible permutation ψ is then $\psi = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{smallmatrix}\right)$. Then $\pi_I = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{smallmatrix}\right)$.

Message-Analysis Problem

Definition (Message-analysis problem)

Let α be combinatoric, *struct* and *concr* be two frames with domain D. We say that β is a *message-analysis* problem if $\beta \equiv \mathrm{MsgAna}(D, \alpha, \mathit{struct}, \mathit{concr})$ with:

$$\mathit{MsgAna}(D, \alpha, \mathit{struct}, \mathit{concr}) \equiv \ \alpha \wedge \phi_{\mathit{gen}}(D) \wedge \phi_{\mathit{frame}}(\mathit{struct}) \\ \wedge \phi_{\mathit{frame}}(\mathit{concr}) \wedge \phi_{\sim}(\mathit{struct}, \mathit{concr})$$

The goals encoded in Alpha-Beta Privacy

$$\alpha \equiv v_1 \in \{0,1\} \wedge \ldots \wedge v_N \in \{0,1\} \wedge \sum_{i=1}^{N} v_i = R\,, \tag{1}$$

$$\beta \equiv \bigwedge_{i=1}^{N} \Big(v[i] = v_i \wedge r[i] = r_i \Big) \wedge MsgAna(D, \alpha, struct, concr) \tag{2}$$

Defining the interpretation

We have to define an interpretation for the voting, commitment and the permutation functions:

Definition (A model of the functions)

Let $\mathcal{I}$ map v to the function $\mathcal{I}(v)\colon A \to A$, r to the function $\mathcal{I}(r)\colon A \to A$ and π to the function $\mathcal{I}(\pi)\colon A \to A$:

$$\begin{array}{rcll}
\mathcal{I}(v)([\![t]\!]_\approx) & = & [\![\theta_I(v_t)]\!]_\approx & \text{if } t \in [\![\{1, \ldots, N\}]\!]_\approx \\
\mathcal{I}(r)([\![t]\!]_\approx) & = & [\![r_{\psi(t)}]\!]_\approx & \text{if } t \in [\![\{1, \ldots, N\}]\!]_\approx \\
\mathcal{I}(\pi)([\![t]\!]_\approx) & = & [\![\pi_I(t)]\!]_\approx & \text{if } t \in [\![\{1, \ldots, N\}]\!]_\approx
\end{array}$$

Defining the interpretation

We also have to define an interpretation for the symbols *gen*, *struct* and *concr*. They are independent from the considered protocol:

Definition (A model of *gen*, *struct* and *concr*)

Let D be the domain of the considered frames. Then we define

$$\begin{array}{rcl} \mathcal{I}(gen) & = & \{[\![t]\!]_{\approx} \mid t \in \mathcal{T}_{\Sigma_{op} \cup D}\} \\ \mathcal{I}(struct)([\![t]\!]_{\approx}) & = & \mathcal{I}(struct\{\!|t|\!\}) \text{ for all } t \in \mathcal{T}_{\Sigma_f} \\ \mathcal{I}(concr)([\![t]\!]_{\approx}) & = & \mathcal{I}(concr\{\!|t|\!\}) \text{ for all } t \in \mathcal{T}_{\Sigma_f} \end{array}$$

Canonic construction

The previous definition gives rise to "canonical" construction independent of the considered protocol:

Lemma

$\mathcal{I} \models \phi_{frame}(struct)$ *and* $\mathcal{I} \models \phi_{frame}(concr)$.

Lemma

If $\mathcal{I}(struct) = \mathcal{I}(concr)$ *then* $\mathcal{I} \models \phi_{\sim}(struct, concr)$.

Voting Privacy

Theorem

Voting privacy holds in the last state of the simplified FOO’92.

Idea of the proof:

- We already proved that $\mathcal{I} \models \phi_{\mathit{frame}}(\mathit{struct})$ and $\mathcal{I} \models \phi_{\mathit{frame}}(\mathit{concr})$
- We have to prove that $\mathcal{I}(\mathit{struct}) = \mathcal{I}(\mathit{concr})$

Proof Sketch

$$\begin{aligned}
\mathcal{I}(v[\pi[i]]) &= \mathcal{I}(v)(\mathcal{I}(\pi)([\![i]\!]_{\approx})) = \mathcal{I}(v)([\![\pi_I(i)]\!]_{\approx}) = \mathcal{I}(v)([\![(\psi^{-1} \circ \pi_0)(i)]\!]_{\approx}) \\
&= [\![\theta_I(v_{\psi^{-1}(\pi_0(i))})]\!]_{\approx} = [\![\theta_0(v_{\pi_0(i)})]\!]_{\approx} \\
\mathcal{I}(r[\pi[i]]) &= \mathcal{I}(r)(\mathcal{I}(\pi)([\![i]\!]_{\approx})) = \mathcal{I}(r)([\![\pi_I(i)]\!]_{\approx}) = \mathcal{I}(r)([\![(\psi^{-1} \circ \pi_0)(i)]\!]_{\approx}) \\
&= [\![r_{(\psi \circ \psi^{-1} \circ \pi_0)(i)}]\!]_{\approx} = [\![r_{\pi_0(i)}]\!]_{\approx}.
\end{aligned}$$

Receipt-freeness

Definition

Receipt-freeness: no voter has a way to prove how they voted. This can be indirectly expressed by saying: for everything that could have happened according to a voting privacy scenario, the voter can make up a consistant "story".

Receipt-freeness

Setup

- We introduce a particular voter: Dan.
- The question is whether Dan can prove to the intuder how he voted by a kind of "receipt".
- FOO'92 is not receipt-free...
- ...but it is in our simplified protocol, i.e. the intruder cannot see the exchanges between the voters and the administrator.

Receipt-freeness

Dan's knowledge

- Dan's knowledge: $\mathit{concr}_{\mathrm{Dan}}$ and $\mathit{struct}_{\mathrm{Dan}}$ over domain $D_{\mathrm{Dan}} = \{d_1, \ldots, d_l\}$.
- The idea is that what Dan can lie about is $\mathit{concr}_{\mathrm{Dan}}$.

$$\begin{aligned}
&\mathit{struct}_{\mathrm{Dan}} = \{\!| d_0 \mapsto \mathit{pub}(A), d_1 \mapsto \mathit{pub}(V_1), \ldots, d_n \mapsto \mathit{pub}(V_N),\\
&d_{N+1} \mapsto \mathit{sign}(\mathit{priv}(A), \mathit{commit}(v[\pi[1]], r[\pi[1]])), \ldots,\\
&d_{2N} \mapsto \mathit{sign}(\mathit{priv}(A), \mathit{commit}(v[\pi[N]], r[\pi[N]])),\\
&d_{2N+1} \mapsto r[\pi[1]], \ldots, d_{3N} \mapsto r[\pi[N]], d_{3N+1} \mapsto \mathit{priv}(\mathrm{Dan}), d_{3N+2} \mapsto v[1],\\
&d_{3N+3} \mapsto r[1], d_{3N+4} \mapsto b_1 |\!\}
\end{aligned}$$

The Axiom of Lying

$$
\begin{aligned}
&\phi_{lie}(struct, concr, struct_{\mathrm{Dan}}, concr_{\mathrm{Dan}}) \equiv \\
&struct[d_1] = struct_{\mathrm{Dan}}[d_1] \wedge \cdots \wedge struct[d_l] = struct_{\mathrm{Dan}}[d_l] \\
&\wedge \exists s_1, \ldots, s_l : gen_{D_{\mathrm{Dan}}}.(concr[d_1] = concr_{\mathrm{Dan}}[d_1] \wedge \cdots \wedge concr[d_l] = concr_{\mathrm{Dan}}
\end{aligned}
$$

The Axiom of Lying

$$\phi_{lie}(struct, concr, struct_{\mathrm{Dan}}, concr_{\mathrm{Dan}}) \equiv$$
$$struct[d_1] = struct_{\mathrm{Dan}}[d_1] \wedge \cdots \wedge struct[d_l] = struct_{\mathrm{Dan}}[d_l]$$
$$\wedge \exists s_1, \ldots, s_l : gen_{D_{\mathrm{Dan}}}.(concr[d_1] = concr_{\mathrm{Dan}}[d_1] \wedge \cdots \wedge concr[d_l] = concr_{\mathrm{Dan}}$$

Definition (Receipt-freeness problem)

$$RcpFree(D, D_{\mathrm{Dan}}, \alpha, struct, concr, struct_{\mathrm{Dan}}, concr_{\mathrm{Dan}})$$
$$\equiv \phi_{gen_{D_{\mathrm{Dan}}}}(D_{\mathrm{Dan}}) \wedge \phi_{frame}(struct_{\mathrm{Dan}}) \wedge \phi_{frame}(concr_{\mathrm{Dan}})$$
$$\wedge\ MsgAna(D \cup D_{\mathrm{Dan}}, \alpha, struct, concr)$$
$$\wedge\ \phi_{lie}(struct, concr, struct_{\mathrm{Dan}}, concr_{\mathrm{Dan}})$$

Receipt-freeness

The lying strategy

Dan can choose any vote on the bulletin board consistant with the intruder's hypothesis!

$$s_{3N+2} = \mathit{open}(\mathit{retrieve}(d_{N+\psi(1)}), d_{2N+\psi(1)}) \text{ and } s_{3N+3} = d_{2N+\psi(1)}$$

Coercion-resistance

Coercion-resistance: no voter has a way to prove how they voted even when the intruder can additionally require some values to be used in advance. In other word, for everything that could have happened according to a voting privacy scenario, the voter can make up a consistent "story" even though the intruder has fixed part of the "story".

Conclusion

- Privacy goals are more subtle than standard secrecy!
  - Relatively complicated notions like (observational) equivalence.
  - Both hard for the modeler and automated tools.
- α-β-privacy as an new way to specify more declaratively:
  - what high-level information α we publish (or reveal to an intruder)
  - and what low-level/cryptographic information β can be observed.
- Privacy as a reachability problem: can we reach a state where β allows for an interesting derivation that α does not imply?