A Formal Taxonomy of Privacy in Voting Protocols

Jannik Dreier, Pascal Lafourcade, Yassine Lakhnech

A Formal Taxonomy of Privacy in Voting Protocols

Jannik Dreier
Université Grenoble 1, CNRS, Verimag
jannik.dreier@imag.fr

Pascal Lafourcade
Université Grenoble 1, CNRS, Verimag
pascal.lafourcade@imag.fr

Yassine Lakhnech
Université Grenoble 1, CNRS, Verimag
yassine.lakhnech@imag.fr

***Abstract*—Privacy is one of the main issues in electronic voting. We propose a family of symbolic privacy notions that allows to assess the level of privacy ensured by a voting protocol. Our definitions are applicable to protocols featuring multiple votes per voter and special attack scenarios such as vote-copying or forced abstention. Finally we employ our definitions on several existing voting protocols to show that our model allows to compare different types of protocols based on different techniques, and is suitable for automated verification using existing tools.**

I. Introduction

Electronic voting systems have been designed and employed in practice for several years. However their use in general elections is controversial due to their security issues [1], [2], [3]. Researchers have identified numerous security properties that are required for secure voting systems and protocols. In this paper we will concentrate on privacy, which is often split into three properties:

- *Vote-Privacy*: The votes are kept private. This can also be modeled as an unlinkability between the voter and his vote.
- *Receipt-Freeness*: A voter cannot construct a receipt which allows him to prove to a third party that he voted for a certain candidate. This is to prevent vote-buying.
- *Coercion-Resistance*: Even when a voter interacts with a coercer during the entire voting process, the coercer cannot be sure whether he followed his instructions or actually voted for another candidate.

However the design of protocols to fulfill the high requirements in electronic voting is notoriously difficult and error-prone. To avoid bugs and analyze protocols, formal verification methods are an ideal tool and have been used in security and safety critical system for several years. In the area of voting protocols, many different formal models and definitions of the above mentioned properties have been proposed and used successfully to discover bugs (e.g. in Helios [4]). However, since the structure, setting and basic design of voting protocols can be quite different depending on the primitives used, many of these definitions are tailored to fit a specific (sub-)group of protocols. Sometimes a protocol can be proved secure in one model, but not in another. This hinders the objective comparisons of protocols.

In particular in the area of privacy properties there is a great variety of models. Moreover, recent research shows that some existing definitions might be insufficient: Smyth and Cortier [4] pointed out that the ability to copy another voter's vote can enable attacks on privacy.

Our Contributions.

1) We propose a new family of privacy notions which allows to assess the level of privacy provided by a voting protocol. These notions are based on formal definitions of the classical notions (Vote-Privacy, Receipt-Freeness and Coercion-Resistance) in the applied pi calculus [5], with a refined protocol model and including new attacks. The resulting family gives a deep understanding of the different levels and requirements for privacy. Additionally – by including a generalization of the notion of "Vote-Independence" [6] – our notions deal with attacks based on vote-copying that were not captured in the model by Delaune et al. [7].
2) A deep understanding of privacy properties, and in particular the relationship between the different notions, is a prerequisite for the correct design of voting protocols. We provide a thorough comparison of existing and new notions.
3) Using several case studies [8], [9], [10], [11] we show that our model allows the analysis of different types of protocols. To automatically analyze protocols we use ProSwapper [12] and ProVerif [13]. Using these tools we can automate some proofs of Vote-Independence which were done by hand previously [6], and check a protocol supporting multiple votes (a variant of [9]).

Related Work. Although the informal specifications of the properties are very general, most of the formal models and definitions in the literature are tailored to a specific type of protocols. Many protocols were in fact developed together with their own definitions (e.g. [14], [10]) and analyzed by hand in the original paper.

Juels et al. [10] (which became the bases for Civitas [15]) were the first to give a formal, but computational definition of coercion-resistance. It was later translated to the applied pi calculus and automated using ProVerif [16]. However – as their protocol is based on voting "credentials" – credentials also appear in the definition. Their model is thus unsuitable for protocols that do not use credentials (e.g. Bingo Voting [11] or the protocol by Lee et al. [17]).

This work was supported by ANR SeSur AVOTE.

DOI http://dx.doi.org/10.1109/ICC.2012.6364938

More general definitions were developed by Delaune, Kremer and Ryan [7]. They express different levels of privacy as observational equivalence in the applied pi calculus [5]. An attacker should not be able to distinguish one case in which the voter complies with the coercer's instructions and another in which he only pretends to do so and votes as he wishes. However their definitions are too strong for the protocol by Juels et al.: since in one case the targeted voter complies and posts only one correct ballot, and in the other he secretly posts his actual ballot and a fake one to cheat the coercer, both cases can be distinguished by counting the ballots.

Smyth and Cortier [4] showed that being able to copy votes can compromise privacy if the number of participants is small or a noticeable fraction of voters can be corrupted. For example in the case of three voters, the third voter can try to copy the first voter's vote and submit it as his vote. This will result in (at least) two votes for the candidate chosen by the first voter and his choice can thus be inferred from the result. They also formally analyzed ballot secrecy in Helios using an adaption of the model by Delaune, Kremer and Ryan [7]. However it was shown that, in general, the DKR model is not sufficient to capture vote-independence. For example the protocol by Lee et al. [17] was shown to be coercion-resistant in this model, despite its vulnerability to vote-copy attacks [6].

Küsters and Truderung [18] proposed a first model independent definition of coercion-resistance for voting protocols. Their definition has to be instantiated using a concrete formal model. The exact security level can be defined with respect to certain chosen goals, and excluding explicit special cases. In contrast to our family of notions, their definition is based on traces and not bisimulations.

Computational definitions of receipt-freeness [19] and coercion-resistance [20] that can be applied to other applications than voting have also been proposed. Completely application-independent anonymity notions were proposed by Bohli and Pashalidis [21]. Although their definitions are very general, the application on voting protocols results in – for this context – rather unusual privacy notions (Pseudonymity etc.), compared to the classic properties such as receipt-freeness or coercion-resistance.

Outline of the Paper. In the next section, we give a brief introduction of the applied pi calculus and develop our model of a voting process. Section III starts by explaining informally our privacy notions and subsequently gives the formal definitions. In Section IV, we discuss the relationship between the different notions, explain the hierarchy implied by the definitions and analyze several case studies to illustrate our definitions. In the last section, we conclude and discuss future work.

II. Preliminaries

A. The Applied Pi Calculus

The applied pi calculus [5] is a formal language to describe concurrent processes (in particular cryptographic protocols) with tool support [12], [13]. The calculus consists of *names* (which typically correspond to data or channels), *variables*, and a *signature* Σ of *function symbols* which can be used to build *terms*. Functions typically include encryption and decryption (for example $\mathtt{enc}(message,\ key)$, $\mathtt{dec}(message, key)$), hashing, signing etc. Terms are correct (i.e. respecting arity and sorts) combinations of names and functions. Equalities are modeled using an equational theory E which defines a relation $=_E$. A classical example which describes the correctness of symmetric encryption is $\mathtt{dec}(\mathtt{enc}(message, key), key) =_E message$. Plain processes are constructed using the following grammar:

$P,\ Q,\ R :=$	processes
0	null process
$P \vert Q$	parallel composition
$!P$	replication
$\nu n.P$	name/variable restriction
$\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q$	conditional
$\mathtt{in}(u, x)$	message input
$\mathtt{out}(u, x)$	message output

Extended processes are plain processes or active substitutions $\{^M/_x\}$. The substitution $\{^M/_x\}$ replaces the variable x with term M. We denote by $fv(A)$, $bv(A)$, $fn(A)$, $bn(A)$ the free variables, bound variables, free names or bound names respectively. A process is *closed* if all variables are bound or defined by an active substitution. A context $C[\_]$ denotes a process with a hole for a process.

More details and the semantics are given in the original paper [5]. To reason about the equivalence or bisimilarity of processes, we use the following bisimilarity relation.

Definition 1 (Labeled Bisimilarity ($\approx_l$) [5]): Labeled bisimilarity is the largest symmetric relation $\mathcal{R}$ on closed processes, such that $A\ \mathcal{R}\ B$ implies

1) $A \approx_s B$,
2) if $A \to A'$, then $B \to B'$ and $A'\ \mathcal{R}\ B'$ for some B',
3) if $A \xrightarrow{\alpha} A'$ and $fv(\alpha) \subseteq \mathrm{dom}(A)$ and $bn(\alpha) \cap fn(B) = \emptyset$, then $B \to^* \xrightarrow{\alpha} \to^* B'$ and $A'\ \mathcal{R}\ B'$ for some B'.

Each interaction on one side can be simulated by the other side, and the processes are statically equivalent (see [5]) at each step during the execution, thus an attacker cannot distinguish both sides.

To formally describe the interaction between a voter and the attacker, we use the following two definitions. The first one turns a process P into another process P^{ch} that reveals all its inputs and secret data on the channel ch.

Definition 2 (Process P^{ch} [7]): Let P be a plain process and ch be a channel name. We define P^{ch} as follows:

- $0^{ch} \triangleq 0$,
- $(P|Q)^{ch} \triangleq P^{ch}|Q^{ch}$,
- $(\nu n.P)^{ch} \triangleq \nu n.\mathtt{out}(ch, n).P^{ch}$ when n is a name of base type,
- $(\nu n.P)^{ch} \triangleq \nu n.P^{ch}$ otherwise,
- $(\mathtt{in}(u, x).P)^{ch} \triangleq \mathtt{in}(u, x).\mathtt{out}(ch, x).P^{ch}$ when x is a variable of base type,
- $(\mathtt{in}(u, x).P)^{ch} \triangleq \mathtt{in}(u, x).P^{ch}$ otherwise,
- $(\mathtt{out}(u, M).P)^{ch} \triangleq \mathtt{out}(u, M).P^{ch}$,
- $(!P)^{ch} \triangleq !P^{ch}$,

- $(\texttt{if } M = N \texttt{ then } P \texttt{ else } Q)^{ch} \doteq \texttt{if } M = N \texttt{ then } P^{ch}$ $\texttt{else } Q^{ch}$.

In the remainder we assume that $ch \notin fn(P) \cup bn(P)$ before applying the transformation. The second definition does not only reveal the secret data, but also takes orders from an outsider before sending a message or branching, i.e. the process is under complete remote control.

Definition 3 (Process P^{c_1,c_2} [7]): Let P be a plain process and c_1, c_2 be channel names. We define P^{c_1,c_2} as follows:

- $0^{c_1,c_2} \doteq 0$,
- $(P|Q)^{c_1,c_2} \doteq P^{c_1,c_2}|Q^{c_1,c_2}$,
- $(\nu n.P)^{c_1,c_2} \doteq \nu n.\texttt{out}(c_1, n).P^{c_1,c_2}$ when n is a name of base type,
- $(\nu n.P)^{c_1,c_2} \doteq \nu n.P^{c_1,c_2}$ otherwise,
- $(\texttt{in}(u, x).P)^{c_1,c_2} \doteq \texttt{in}(u, x).\texttt{out}(c_1, x).P^{c_1,c_2}$ when x is a variable of base type and x is a fresh variable,
- $(\texttt{in}(u, x).P)^{c_1,c_2} \doteq \texttt{in}(u, x).P^{c_1,c_2}$ otherwise,
- $(\texttt{out}(u, M).P)^{c_1,c_2} \doteq \texttt{in}(c_2, x).\texttt{out}(u, x).P^{c_1,c_2}$,
- $(!P)^{c_1,c_2} \doteq\, !P^{c_1,c_2}$,
- $(\texttt{if } M = N \texttt{ then } P \texttt{ else } Q)^{c_1,c_2} \doteq \texttt{in}(c_2, x).\texttt{if } x =$ true $\texttt{then } P^{c_1,c_2} \texttt{ else } Q^{c_1,c_2}$ where and x is a fresh variable and true is a constant.

The following definition hides the output of a process.

Definition 4 (Process $A^{\backslash out(ch,\cdot)}$ [7]): Let A be an extended process. We define the process $A^{\backslash out(ch,\cdot)}$ as $\nu ch.(A|!\text{in}(ch, x))$.

B. Voting Protocol and Process

First of all, we define the notion of a *voting protocol*. Informally, a voting protocol specifies the processes executed by voters and authorities.

Definition 5 (Voting Protocol): A voting protocol is a tuple $(V, A_1, \ldots, A_m, \tilde{n})$ where V is the process that is executed by the voter, the A_j's are the processes executed by the election authorities, and $\tilde{n}$ is a set of private channels.

Note that the protocol only defines one process V which will be instantiated for each voter. Yet here may be several authorities, for example a registrar, a bulletin board, a mixer, a tallier, ... In our privacy definitions we reason about privacy using concrete instances of a voting protocol. An instance is called a*Voting Process*.

Definition 6 (Voting Process): A voting process of a voting protocol $(V, A_1, \ldots, A_m, \tilde{n})$ is a closed plain process

$$\nu\tilde{m}.(V\sigma_{id_1}\sigma_{f_1}\sigma_{v_1}| \ldots |V\sigma_{id_n}\sigma_{f_n}\sigma_{v_n}|A_1| \ldots |A_l)$$

where $l \leq m$, $\tilde{m}$ includes the secret channel names, $V\sigma_{id_i}\sigma_{v_i}\sigma_{f_i}$ are the processes executed by the voters where:

- σ_{id_i} is a substitution assigning the identity to a process (this determines for example the secret keys),
- σ_{v_i} specifies the vote(s) and if the voter abstains,
- and σ_{f_i} defines the other behavior, in particular if fake votes are issued,

and the A_js are the honest election authorities.

We notice that if an authority is not supposed to be honest, it is not modeled and left to the context, i.e. the attacker (thus $l \leq m$). Note also that each voter runs the same process V, which is instantiated with a different σ_{id_i} (his identity), σ_{v_i} (his vote(s)) and σ_{f_i} (the fakes). If a protocol does not allow fakes, σ_{f_i} is empty.

This model allows us to reason about more than one correct behavior, which is necessary if for example a voter can decide to abstain from voting or if – in case of multiple votes[1] – he can vote between 0 and n times in the same election. In this case V defines all the possible executions, and σ_{v_i} and σ_{f_i} will determine which of them is executed. Another application are protocols where voters can submit fake ballots and/or several real ballots, even if only one of them is actually counted (like in the one by Juels et al. [10]). In that case σ_{v_i} determines those who are actually counted, and σ_{f_i} the others.

Example 1: As a running example, we consider the following simple voting protocol.

Informal description: To construct a ballot, each voter encrypts his vote with the administrator's public key and signs it. The resulting ballot is posted on the bulletin board. After the voting deadline is over, the administrator checks if each ballot is signed by an eligible voter. He then decrypts the correct ballots and publishes the result.

Formal description in our model: The protocol is a tuple $(V, A, \emptyset)$ where

$$\begin{aligned} A &= \texttt{in}(ch, (sig, vote)). \\ &\ \texttt{if checksign}(sig, pkv) = vote \\ &\ \texttt{then sync } 1.\texttt{out}(chR, \texttt{dec}(vote, ska)) \\ V &= \nu r.\texttt{let } evote = \texttt{enc}(v, pka, r) \texttt{ in} \\ &\ \texttt{out}(ch, (\texttt{sign}(evote, skv), evote)) \end{aligned}$$

where – by abuse of notation – $\texttt{sync } 1$ is a synchronization point as implemented by ProSwapper [12]. A substitution determining the identity of a voter would in this case assign the secret key, e.g. $\sigma_{id_k} = \{^{sk_k}/_{skv}\}$. The substitution specifying the vote as for example a vote for candidate a would be $\sigma_{v_k} = \{^{a}/_{v}\}$. As the protocol does not specify the possibility to create fakes, σ_{v_k} is the empty substitution.

To facilitate notation we denote by S and S' two contexts which are like voting processes but with holes for two and three voters respectively.

Definition 7 (S and S'): We define evaluation contexts S and S' such that

- S is like a voting process, but has a hole instead of three processes among $(V\sigma_{id_i}\sigma_{v_i}\sigma_{f_i})_{1\leq i\leq n}$
- S' which is like as voting process, but has a hole instead of two processes among $(V\sigma_{id_i}\sigma_{v_i}\sigma_{f_i})_{1\leq i\leq n}$.

Finally, we formally define what it means for a voting process to abstain. An abstaining voter does not send any message on any channel, in particular no ballot, which would correspond to a voter that does not even go to polling station. This is stronger than just voting for a particular "null" candidate $\perp$, which will still result in sending a ballot (a *blank* vote).

Definition 8 (Abstention): A substitution σ_{v_i} makes a voter abstain if $V\sigma_{id_i}\sigma_{v_i} \approx_l 0$.

Note that abstention is determined by σ_{v_i} only, so the voter abstains for any σ_{f_i}.

[1] By multiple votes we mean a protocol where each voter can vote several times in the same election, and each vote is transmitted in a separate ballot.

III. Defining Privacy: A Multidimensional Approach

In our setting, the attacker targets one voter (the *targeted voter*) and tries to extract information about the targeted voter's vote(s). If the attacker knows the votes of all other voters, he can infer the targeted voter's vote from the result. Thus we suppose that he is unsure about the vote(s) of one other voter (the *counterbalancing voter*).

We express privacy as an observational equivalence. Intuitively, an attacker should not be able to distinguish between an execution in which the targeted voter behaves and votes as the attacker wishes, and another execution where he only pretends to do so and votes differently. To ensure that the attacker cannot tell the difference by just comparing the result, the counterbalancing voter will compensate the different vote.

Starting from the definitions of Coercion-Resistance, Receipt-Freeness and Vote-Privacy in the literature [7], [16], [10] we propose extensions in the four following dimensions: Communication between attacker and targeted voter, Vote-Independence, security against forced-abstention-attacks and knowledge about the behavior of the counterbalancing voter.

1) *Communication between the attacker and the targeted voter:* We define three different levels:
   a) In the simplest case, the attacker only observes publicly available data and communication. We call this case Vote-Privacy, denoted VP.
   b) In the second case, the targeted voter tries to convince the attacker that he voted for a certain candidate by revealing his secret data. Yet the attacker should not be able to determine if he actually sent his real data, or a fake receipt. We call this case Receipt-Freeness, denoted RF.
   c) In the strongest case, the voter pretends to be completely under the control of the attacker, i.e. he reveals his secret data and follows the intruder's instructions. Yet the attacker should be unable to determine if he complied with his instructions or if he only pretended to do so. We call this case Coercion-Resistance, denoted CR.

   It is easy to see that Coercion-Resistance is stronger than Receipt-Freeness, which is stronger than Vote-Privacy ($CR > RF > VP$).
2) *Vote-Independence/Corrupted Voter:* The attacker may control another legitimate voter (neither the targeted nor the counterbalancing voter). In that case he could be able to compromise privacy by trying to relate the corrupted voter's vote to the targeted voter's vote (e.g. by copying it) or using the corrupted voter's secret data, such as his credentials or keys. In our definitions, we distinguish two case for Eve (the attacker):
   a) Eve is an Outsider (denoted O): The attacker is an external observer.
   b) Eve is an Insider (denoted I): The attacker has a legitimate voter under his control.

   Intuitively, Insider is the stronger setting ($I > O$).
3) *Security against forced-abstention-attacks:* A protocol can ensure that a voter can still vote as intended, although a coercer wants him to abstain. Note that in contrast to the literature [10], [16], we define this property independently from of Coercion-Resistance, as we also want to apply it in the case of Vote-Privacy. Our model expresses this by requiring the observational equivalence to hold:
   a) in any case, i.e. even if the voter is forced to abstain. We call this case security against Forced-Abstention-Attacks, denoted FA.
   b) if the targeted voter does not abstain from voting (i.e. always participates). We call this case Participation Only, denoted PO.

   In this dimension security against Forced-Abstention-Attacks is a stronger property than Participation Only ($FA > PO$).
4) *Knowledge about the behavior of the counterbalancing voter*: To model different knowledge about the behavior of the counterbalancing voter, we consider two cases:
   a) The observational equivalence holds for any behavior of this voter, i.e. any σ_{f_i}. This models an attacker that knows if the counterbalancing voter is going to post fake ballots, or a situation where there is no "noise" (=fake ballots) on the bulletin board. We call this case Any Behavior, denoted AB.
   b) The observational equivalence holds for at least one behavior of this voter, which may additionally change, i.e. one σ_{f_i} and one $\sigma_{f'_i}$. In this case, the attacker is unsure about the number of fake ballots, i.e. there is enough noise. We call this case Exists Behavior, denoted EB.

   Any Behavior is stronger than EB ($AB > EB$).

The strongest possible property is thus $CR^{I,FA,AB}$, the weakest $VP^{O,PO,EB}$. If we leave out the parameter, we take the weakest setting as a default, i.e. VP denotes $VP^{O,PO,EB}$.

A. Definitions in the applied pi calculus

Our definition is parametrized using the following parameters (as explained above):

- $Privacy = \{CR, RF, VP\}$ ("Coercion-Resistance", "Receipt-Freeness" or "Vote-Privacy").
- $Eve = \{I, O\}$ ("Insider" or "Outsider").
- $Abs = \{FA, PO\}$ ("Security against Forced-Abstention-Attacks"or "Participation Only").
- $Behavior = \{AB, EB\}$ ("Any Behavior"or "Exists Behavior").

Definition 9 ($Privacy^{Eve,Abs,Behavior}$): A protocol fulfills $Privacy^{Eve,Abs,Behavior}$ if for any voting process $\mathcal{S}$ there exists a process V' and for any substitution σ_{f_A} and σ_{f_C}, and any context A such that $A = \nu \tilde{ch}.(\_|A'^{chout}))$ where $\tilde{ch}$ are all unbound channels and names in A' and in the "hole",

- if $Behavior$ is EB: $\exists$ substitutions σ_{f_B}, $\sigma_{f'_B}$ and $\sigma_{f'_A}$,
- if $Behavior$ is AB: $\forall\ \sigma_{f_B} = \sigma_{f'_B}\ \exists\ \sigma_{f'_A}$,

such that for all votes σ_{v_A} and σ_{v_B} where $V\sigma_{v_B}$ does not make a voter abstain[2], one of the following holds depending on the privacy setting:

- if Privacy is Vote-Privacy (VP):
  $$A\left[\mathcal{S}\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|\mathcal{V}_C\right]\right] \approx_l A\left[\mathcal{S}\left[V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}|V\sigma_{id_B}\sigma_{f'_B}\sigma_{v_A}|\mathcal{V}_C\right]\right]$$
- if Privacy is Receipt-Freeness (RF):
  - $V'^{\backslash out(chc,\cdot)} \approx_l V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}$
  - $A\left[\mathcal{S}\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}^{chc}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|\mathcal{V}_C\right]\right] \approx_l A\left[\mathcal{S}\left[V'|V\sigma_{id_B}\sigma_{f'_B}\sigma_{v_A}|\mathcal{V}_C\right]\right]$
- if Privacy is Coercion-Resistance (CR):
  For any context $C = \nu c_1.\nu c_2.(\_|P')$ with $\tilde{n} \cap fn(C) = \emptyset$ and $\mathcal{S}\left[C\left[V\sigma_{id_A}^{c_1,c_2}\right]|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|\mathcal{V}_C\right] \approx_l \mathcal{S}\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}^{chc}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|\mathcal{V}_C\right]$
  we have
  - $C\left[V'\right]^{\backslash out(chc,\cdot)} \approx_l V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}$
  - $A\left[\mathcal{S}\left[C\left[V\sigma_{id_A}^{c_1,c_2}\right]|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|\mathcal{V}_C\right]\right] \approx_l A\left[\mathcal{S}\left[C\left[V'\right]|V\sigma_{id_B}\sigma_{f'_B}\sigma_{v_A}|\mathcal{V}_C\right]\right]$

where

- If Eve is:
  - Insider(I): $\mathcal{S} := S$ and $\mathcal{V}_C := V\sigma_{id_C}^{c_1,c_2}$
  - Outsider (O): $\mathcal{S} := S'$ and $\mathcal{V}_C := 0$
- If Abs is:
  - Participation Only (PO): $V\sigma_{id_A}$ does not abstain, i.e. $V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A} \not\approx_l 0$.
  - Security against Forced-Abstention-Attacks (FA), he may abstain.

The context A represents the attacker. We chose to make the attacker's behavior explicit as some protocols (such as the one by Juels et al. [10]) require σ_{f_B} and $\sigma_{f'_B}$ to be chosen as a function of the attacker (see our technical report [22] for details, note however that V' is chosen only as a function of the protocol). To ensure that A does not only forward the channels (which would leave the attacker to the outside again and thus contradict our intention of choosing the processes as a function of A), we require A to bind all free names and channels inside. He will only forward the knowledge he is able to obtain during the execution of the protocol.

B. Examples and relation to existing notions

The following examples illustrate how the parameters change the definition and give intuitions.

Example 2 ($VP^{O,PO,AB}$): A protocol fulfills $VP^{O,PO,AB}$ if for any voting process S' and any substitutions σ_{f_A}, σ_{f_B} and σ_{f_C}, and for any context A such that $A = \nu\tilde{ch}.(\_|A'^{chout}))$ where $\tilde{ch}$ are all unbound channels and names in A' and in the "hole", there exist a substitution $\sigma_{f'_A}$ such that for all votes σ_{v_A} and σ_{v_B} where σ_{v_B} and σ_{v_A} does not make a voter abstain we have $A\left[S'\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|0\right]\right] \approx_l A\left[S'\left[V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_A}|0\right]\right]$.
Note that – as labeled bisimilarity is closed under the application of contexts – it is sufficient to prove $S'\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}\right] \approx_l S'\left[V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_A}\right]$ If there is only one correct behavior of V (i.e. no fakes), we can rewrite this as

$$S'\left[V\sigma_{id_A}\sigma_{v_A}|V\sigma_{id_B}\sigma_{v_B}\right] \approx_l S'\left[V\sigma_{id_A}\sigma_{v_B}|V\sigma_{id_B}\sigma_{v_A}\right] \quad (1)$$

This coincides with the definition of Vote-Privacy given by Delaune et al. [7]: Two situations where two voters swap votes are bisimilar. We also note that Receipt-Freeness in the DKR-model corresponds to $RF^{O,PO,AB}$ in our model, and Coercion-Resistance in the DKR-model corresponds to $CR^{O,PO,AB}$ in our model.

Example 3 (Application): Consider our running example of a simple voting protocol. We show that it ensures $VP^{O,PO,AB}$ as defined above. We suppose the secret keys to be secret and the administrator to be honest. In that case, Proverif is able to prove (1), which shows that the simple voting protocol ensures $VP^{O,PO,AB}$.[3]. It is easy to see that this protocol does not guarantee Vote-Privacy for an inside attacker ($VP^{I,PO,AB}$), as he can simply access the votes on the bulletin board and copy them. He can identify which vote was posted by which voter using the signatures. The protocol is not receipt-free ($RF^{Eve,Abs,Behavior}$) either as the randomness used for encrypting the vote can be used as a receipt. Since the bulletin board reveals which voters participated, it is not resistant against forced-abstention attacks.

Example 4 ($VP^{O,FA,AB}$): A protocol fulfills $VP^{O,FA,AB}$ if for any voting process S' and any substitutions σ_{f_A}, σ_{f_B} and σ_{f_C}, and for any context A such that $A = \nu\tilde{ch}.(\_|A'^{chout}))$ where $\tilde{ch}$ are all unbound channels and names in A' and in the "hole", there exists a substitution $\sigma_{f'_A}$ such that for all votes σ_{v_A} and σ_{v_B} where σ_{v_B} does not make a voter abstain we have $A\left[S'\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|0\right]\right] \approx_l A\left[S'\left[V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_A}|0\right]\right]$. In this case, σ_{v_A} can make a voter abstain. As σ_{v_B} may not specify abstention, we have an observational equivalence between a situation where the targeted voter abstains, and a situation where he votes and the counterbalancing voter abstains. This captures security against forced-abstention-attacks.

Example 5 ($RF^{I,PO,AB}$): A protocol fulfills $RF^{I,PO,AB}$ if for any voting process S there exists a process V', and for any substitutions σ_{f_A}, σ_{f_B} and σ_{f_C}, and for any context A such that $A = \nu\tilde{ch}.(\_|A'^{chout}))$ where $\tilde{ch}$ are all unbound channels and names in A' and in the "hole", there exists a substitution $\sigma_{f'_A}$ such that for all votes σ_{v_A} and σ_{v_B} where σ_{v_B} and σ_{v_A} do not make a voter abstain we have $V'^{\backslash out(chc,\cdot)} \approx_l V\sigma_{id_A}\sigma_{f'_A}\sigma_{v_B}$ and $A\left[S\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}^{chc}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|V\sigma_{id_C}^{c_1,c_2}\right]\right] \approx_l A\left[S\left[V'|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_A}|V\sigma_{id_C}^{c_1,c_2}\right]\right]$.
Note that again it is sufficient to prove $S\left[V\sigma_{id_A}\sigma_{f_A}\sigma_{v_A}^{chc}|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_B}|V\sigma_{id_C}^{c_1,c_2}\right] \approx_l S\left[V'|V\sigma_{id_B}\sigma_{f_B}\sigma_{v_A}|V\sigma_{id_C}^{c_1,c_2}\right]$ as labeled bisimilarity is closed under the application of contexts. If there is only one correct behavior of $V\sigma_A$, this coincides with the definition of Vote-Independence with Passive Collaboration in the DKR-model [6]: If a protocol is

[2] This condition ensures that in the case PO no voter can abstain.

[3] The code used is available on our website: http://www-verimag.imag.fr/~dreier/papers/sfcs-code.zip.

Protocol	**Privacy Notion**	**Comments**
Juels et al. [10]	$CR^{I,FA,EB}$	Requires fakes to achieve CR
Bingo Voting [11]	$CR^{I,PO,AB}$	Trusted voting machine
- variant	$CR^{I,FA,AB}$	Secure against forced abstention
Lee et al. [17]	$CR^{O,PO,AB}$	Vulnerable to vote-copying
Okamoto [9]	$RF^{I,PO,AB}$	Based on trap-door commitments
- variant	$RF^{I,FA,AB}$	Private channel to administrator
Fujioka et al. [8]	$VP^{I,PO,AB}$	Based on blind signatures
- variant	$VP^{I,PO,AB}$	Permits multiple votes
Simp. Voting Prot.	$VP^{O,PO,AB}$	Vulnerable to vote-copying

TABLE I
RESULTS OF THE CASE STUDIES

receipt-free, there exists a counter-strategy (V') that allows the targeted voter to fake the receipt and vote differently. Analogously, Vote-Independence in the DKR-model corresponds to $VP^{I,PO,AB}$, and Vote-Independence with passive Collaboration corresponds to $CR^{I,PO,AB}$ in our model.

IV. Hierarchy and Case Studies

As already described in Section III, we have a hierarchy of notions in each dimension.

Proposition 1: For $Privacy \in \{VP, RF, CR\}$, $Abs \in \{FA, PO\}$ and $Behavior \in \{AB, EB\}$ we have:

1) If a protocol respects $Privacy^{I,Abs,Behavior}$, then it also respects $Privacy^{O,Abs,Behavior}$.
2) If a protocol respects $Privacy^{Eve,FA,Behavior}$, it also respects $Privacy^{Eve,PO,Behavior}$.
3) If a protocol respects $Privacy^{Eve,Abs,AB}$, it also respects $Privacy^{Eve,Abs,EB}$.
4) Coercion-Resistance is stronger than Receipt-Freeness, which is stronger than Vote-Privacy:
    - If a protocol respects $CR^{Eve,Abs,Behavior}$, it also respects $RF^{Eve,Abs,Behavior}$.
    - If a protocol respects $RF^{Eve,Abs,Behavior}$, it also respects $VP^{Eve,Abs,Behavior}$.

This was partly shown before in the DKR-model [7], the extension to our model is straightforward. All the formal proofs are given in our technical report [22].

We applied our family of notions on several case studies, chosen to show that each of our dimensions corresponds to a different property of real-world protocols. Due to space limitations we cannot discuss the protocols in details here (see our technical report [22]). The results are summed up in and Table I. The proofs of coercion-resistance and receipt-freeness were done by hand, yet we were able to verify automatically a variant of the protocol by Fujioka et al. [8] which allows for multiple votes per voter. Additionally we were able to automate the originally manual proof of "Vote-Independence" for the same protocol using a slightly modified model compared to the original paper [6].

V. Conclusion

We proposed a modular family of formal privacy notions in the applied pi calculus which allows to assess the level of privacy provided by a voting protocol. We applied the family of notions in several case studies, including a case with multiple votes per voter (a variant of FOO [8]) and forced abstention attacks (see Table I). In particular we were able to show that the different dimensions of our definitions correspond to different properties of real-world protocols, and that in many cases the verification can be done automatically using existing tools.

Future Work. In this paper we employ a possibilistic approach: We call a protocol secure if there is a way for the targeted voter to escape coercion. As we do not consider probabilities, the adversary may still have a certain probability of detecting that the coerced voter tried to resist coercion. This lies beyond the scope of this paper, yet a computational translation of our definitions should be able to address it.

References

[1] Bundesverfassungsgericht (Germany's Federal Constitutional Court), "Use of voting computers in 2005 bundestag election unconstitutional," March 2009.

[2] UK Electoral Commission, "Key issues and conclusions: May 2007 electoral pilot schemes."

[3] Ministerie van Binnenlandse Zaken en Koninkrijksrelaties (Netherland's Ministry of the Interior and Kingdom Relations), "Stemmen met potlood en papier (voting with pencil andpaper)," May 2008.

[4] B. Smyth and V. Cortier, "Attacking and fixing helios: An analysis of ballot secrecy," in *CSF'11*. IEEE, 2011.

[5] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *POPL'01*, 2001, pp. 104–115.

[6] J. Dreier, P. Lafourcade, and Y. Lakhnech, "Vote-independence: A powerful privacy notion for voting protocols," in *4th Workshop on Foundations & Practice of Security (FPS)*, ser. LNCS, vol. 6888. Springer, 2011, p. 164ff.

[7] S. Delaune, S. Kremer, and M. Ryan, "Verifying privacy-type properties of electronic voting protocols," *Journal of Computer Security*, vol. 17, pp. 435–487, 2009.

[8] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Advances in Cryptology – AUSCRYPT '92*, ser. LNCS. Springer, 1992, vol. 718, pp. 244–251.

[9] T. Okamoto, "An electronic voting scheme," in *Proceedings of the IFIP World Conference on IT Tools*, 1996.

[10] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *WPES'05*. ACM, 2005, pp. 61–70.

[11] J.-M. Bohli, J. Müller-Quade, and S. Röhrich, "Bingo voting: Secure and coercion-free voting using a trusted random number generator," in *E-Voting and Identity*, ser. LNCS. Springer, 2007, vol. 4896, pp. 111–124.

[12] P. Klus, B. Smyth, and M. D. Ryan, "Proswapper: Improved equivalence verifier for proverif." http://www.bensmyth.com/proswapper.php, 2010.

[13] B. Blanchet, M. Abadi, and C. Fournet, "Automated verification of selected equivalences for security protocols," *Journal of Logic and Algebraic Programming*, vol. 75, no. 1, pp. 3–51, 2008.

[14] T. Moran and M. Naor, "Receipt-free universally-verifiable voting with everlasting privacy," in *CRYPTO 2006*, ser. LNCS, vol. 4117. Springer, 2006, pp. 373–392.

[15] M. R. Clarkson, S. Chong, and A. C. Myers, "Civitas: Toward a secure voting system," *IEEE Symposium on Security and Privacy*, vol. 0, pp. 354–368, 2008.

[16] M. Backes, C. Hritcu, and M. Maffei, "Automated verification of remote electronic voting protocols in the applied pi-calculus," *CSF*, vol. 0, pp. 195–209, 2008.

[17] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Providing receipt-freeness in mixnet-based voting protocols," in *ICISC*, ser. LNCS. Springer Berlin / Heidelberg, 2004, vol. 2971.

[18] R. Küsters and T. Truderung, "An Epistemic Approach to Coercion-Resistance for Electronic Voting Protocols," in *S&P*. IEEE, 2009, pp. 251–266.

[19] R. Canetti and R. Gennaro, "Incoercible multiparty computation (extended abstract)," in *FOCS*, 1996, pp. 504–513.

[20] D. Unruh and J. Müller-Quade, "Universally composable incoercibility," in *CRYPTO 2010*, ser. LNCS. Springer, 2010, vol. 6223, pp. 411–428.

[21] J.-M. Bohli and A. Pashalidis, *Relations Among Privacy Notions*. Springer, 2009, pp. 362–380.

[22] J. Dreier, P. Lafourcade, and Y. Lakhnech, "A formal taxonomy of privacy in voting protocols," Verimag Research Report, Tech. Rep. TR-2011-10, May 2011, available at http://www-verimag.imag.fr/TR/TR-2011-10.pdf.