Towards automatic analysis of election verifiability properties [⋆] [⋆⋆]

Ben Smyth[1,2], Mark Ryan[1], Steve Kremer[3], and Mounira Kourjieh[1,4]

[1] School of Computer Science, University of Birmingham, UK
[2] École Normale Supérieure & CNRS & INRIA, France
[3] LSV, ENS Cachan & CNRS & INRIA, France
[4] Université de Toulouse, France

research@bensmyth.com, M.D.Ryan@cs.bham.ac.uk,
kremer@lsv.ens-cachan.fr, kourjieh@irit.fr

Abstract. We present a symbolic definition that captures some cases of election verifiability for electronic voting protocols. Our definition is given in terms of reachability assertions in the applied pi calculus and is amenable to automated reasoning using the software tool ProVerif. The definition distinguishes three aspects of verifiability, which we call individual, universal, and eligibility verifiability. We demonstrate the applicability of our formalism by analysing the protocols due to Fujioka, Okamoto & Ohta and a variant of the one by Juels, Catalano & Jakobsson (implemented as Civitas by Clarkson, Chong & Myers).

Key words: Electronic voting protocols, election verifiability, applied pi calculus, ProVerif, automated reasoning.

1 Introduction

Electronic voting systems are being introduced, or trialled, in several countries to provide more efficient voting procedures with an increased level of security. However, current deployment has shown that the security benefits are very hard to realise [10, 20, 9, 24]. Those systems rely on the trustworthiness of the servers and software that is used to collect, tally, and count the votes, and on the individuals that manage those servers. In practice, it is very hard to establish the required level of trust.

The concept of *election verifiability* that has emerged in the academic literature [16, 19, 22] aims to address this problem. It significantly reduces the necessity to trust electronic systems, by allowing voters and election observers to verify

[⋆] A preliminary version of this work was presented at the WISSec'09 workshop.

[⋆⋆] This work has been partly supported by the EPSRC projects *UbiVal* (EP/D076625/2), *Trustworthy Voting Systems* (EP/G02684X/1) and *Verifying Interoperability Requirements in Pervasive Systems* (EP/F033540/1); the *ANR SeSur AVOTÉ* project; the *Direction Générale pour l'Armement* (DGA); and the University of Birmingham's *Blue BEAR* cluster.

independently that votes have been recorded, tallied and counted correctly. To emphasise a voter's ability to verify the results of the entire election process, it is sometimes called *end-to-end* verifiability [21].

We present a preliminary definition of election verifiability in a formal and general setting, and we analyse voting protocols from the literature. We work in the applied pi calculus [2], and we use the ProVerif software tool [8] to automate verification. (The calculus and the tool have already been successful in analysing other properties of electronic voting systems [14, 4].) Our approach puts significant emphasis on the automatic analysis of the verifiability property, using ProVerif. To that end, we introduce a number of encodings that make ProVerif work more efficiently, which are of independent interest.

Election verifiability allows voters and observers to verify that the election outcome corresponds to the votes legitimately cast. We distinguish three aspects of verifiability:

Individual verifiability: a voter can check that her own ballot is included in the bulletin board.
Universal verifiability: anyone can check that the election outcome corresponds to the ballots.
Eligibility verifiability: anyone can check that each vote in the election outcome was cast by a registered voter and there is at most one vote per voter.

(Note that some authors use the term "universal verifiability" to refer to the conjunction of what we call "universal verifiability" and "eligibility verifiability." This distinction is made for compatibility with protocols which do not offer eligibility verifiability.) These three aspects of verifiability are related to the following *correctness properties* [4], defined with respect to honest protocol executions:

Inalterability: no one can change a voter's vote.
Declared result: the election outcome is the correct sum of the votes cast.
Eligibility: only registered voters can vote and at most once.

Election verifiability properties are different from correctness properties, since they assert that voters and observers can check that the correctness properties hold, even when administrators deviate from the protocol (that is, do not perform an honest execution).

We define election verifiability as a family of three tests, corresponding to the three aspects identified above. The individual verifiability test is performed by each voter. The other two tests can be conducted by any observer.

1.1 Contribution

We present a definition of election verifiability which captures the three desired aspects: individual verifiability, universal verifiability and eligibility verifiability. The definition is a sufficient condition for election verifiability, but it is not be necessary; that is, there are some protocols which offer verifiability but are not captured by our definition.

We formalise our definition as reachability assertions in the applied pi calculus. This makes the definition amenable to automated reasoning. In order to make ProVerif work better with our definitions, we introduce a number of encodings and optimisations.

We demonstrate the applicability of our definition by analysing three protocols. The first protocol is a simple illustrative protocol which is trivially verifiable because voters digitally sign their ballot. (Note that this protocol does not achieve other properties such as privacy). We then analyse the protocol by Fujioka *et al.* [16], and a variant of the one by Juels *et al.* [19]; the latter has been implemented as Civitas [13, 12].

1.2 Related work

Juels, Catalano & Jakobson [18, 19] present the first definition of universal verifiability in the provable security model. Their definition assumes voting protocols produce signature proofs of knowledge demonstrating the correctness of tallying. Automated analysis is not discussed.

Universal verifiability was also studied by Chevallier-Mames et al. [11] with the aim of showing an incompatibility result: protocols satisfying their definition are incompatible with vote-privacy (also called ballot secrecy), and hence coercion-resistance. To see this, note that they require functions f and f' such that for any bulletin board BB and list of eligible voters L the function $f(BB, L)$ returns the list of actual voters and $f'(BB, L)$ returns the election outcome (see Definition 1 of [11]). From these functions one could consider any single bulletin board entry b and compute $f(\{b\}, L)$, $f'(\{b\}, L)$ to reveal a voter and her vote. Our definitions do not reflect such an incompatibility with vote-privacy and permit a large class of electronic voting protocols claiming to provide verifiability to be considered.

Baskar, Ramanujan & Suresh [7] and subsequently Talbi et al. [23] have formalised individual and universal verifiability with respect to the FOO [16] electronic voting protocol. Their definitions are tightly coupled to that particular protocol and cannot easily be generalised. Moreover, their definitions characterise individual executions as verifiable or not; whereas such properties should be considered with respect to every execution (that is, the entire protocol).

1.3 Outline

Section 2 recalls the applied pi calculus. In Section 3 we introduce a generic definition of verifiability which is independent of any particular formal framework. In Section 4 our definition is formalised as reachability assertions in the context of the applied pi calculus. The three case studies are analysed in Section 5. Finally, we conclude and give directions for future work (Section 6).

2 Applied pi calculus

The applied pi calculus [2] is a language for modelling concurrent systems and their interactions. It is an extension of the pi calculus which was explicitly designed for modelling cryptographic protocols. For this purpose, the applied pi calculus allows terms to be constructed over a signature rather than just names. This term algebra can be used to model cryptographic primitives.

2.1 Syntax

The calculus assumes an infinite set of names $a, b, c, k, m, n, s, t, r, \ldots$, an infinite set of variables $v, x, y, z, \ldots$ and a finite signature Σ, that is, a finite set of function symbols each with an associated arity. A function symbol of arity 0 is a constant. We use metavariables u, w to range over both names and variables. Terms F, L, M, N, T, U, V are built by applying function symbols to names, variables and other terms. Tuples $u_1, \ldots, u_l$ and $M_1, \ldots, M_l$ are occasionally abbreviated $\tilde{u}$ and $\tilde{M}$. We write $\{^{M_1}/_{x_1}, \ldots, ^{M_l}/_{x_l}\}$ for substitutions that replace $x_1, \ldots, x_l$ with $M_1, \ldots, M_l$. The applied pi calculus relies on a simple type system. Terms can be of sort $\mathsf{Channel}$ for channel names or Base for the payload sent out on these channels. Function symbols can only be applied to, and return, terms of sort Base. A term is ground when it does not contain variables. The grammar for processes is shown in Figure 1 where u is either a name or variable of channel sort. Plain processes are standard. As usual we abbreviate conditionals as if $M = N$ then P, when Q is the null process; and similarly we may omit the process P in message input and output when P is 0. Extended processes introduce *active substitutions* which generalise the classical let construct: the process $\nu\, x.(\{^{M}/_{x}\} \mid P)$ corresponds exactly to the process let $x = M$ in P. As usual names and variables have scopes which are delimited by restrictions and by inputs. All substitutions are assumed to be cycle-free.

$P, Q, R ::=$ processes		$A, B, C ::=$ extended processes	
0	null process	P	plain process
$P \mid Q$	parallel	$A \mid B$	parallel composition
$!P$	replication	$\nu\, n.A$	name restriction
$\nu\, n.P$	name restriction	$\nu\, x.A$	variable restriction
$u(x).P$	message input	$\{^{M}/_{x}\}$	active substitution
$\overline{u}\langle M\rangle.P$	message output		
if $M = N$ then P else Q	conditional		

Fig. 1. Applied pi calculus grammar

The sets of free and bound names, respectively variables, in process A are denoted by $\mathrm{fn}(A)$, $\mathrm{bn}(A)$, $\mathrm{fv}(A)$, $\mathrm{bv}(A)$. We also write $\mathrm{fn}(M)$, $\mathrm{fv}(M)$ for the names, respectively variables, in term M. An extended process A is *closed* if it has no free variables. A *context* $C[\_]$ is an extended process with a hole. We obtain $C[A]$ as the result of filling $C[\_]$'s hole with A. An *evaluation context* is a context whose hole is not under a replication, a conditional, an input, or an output.

The signature Σ is equipped with an equational theory E, that is a finite set of equations of the form $M = N$. We define $=_E$ as the smallest equivalence relation on terms, that contains E and is closed under application of function symbols, substitution of terms for variables and bijective renaming of names.

We introduce *linear* processes as a subset of plain processes generated by the grammar

$$\begin{aligned} P ::= {} & 0 \mid \nu\, n.P \mid c(x).P \mid \overline{c}\langle M\rangle.P \\ & \mid \text{if } M = N \text{ then } P \text{ else } 0 \quad (P \neq 0) \\ & \mid \text{if } M = N \text{ then } 0 \text{ else } P \quad (P \neq 0) \end{aligned}$$

Linear processes can be sequentially composed in a natural way. Let P be a linear processes and Q a plain process. We define the plain process $P \circ Q$ to be Q if $P = 0$ and otherwise by replacing the unique occurrence of ".0" in P by ".Q". (Note that ".0" does not occur in "else 0".) Moreover, we note that if P and Q are both linear processes then $P \circ Q$ is also a linear process.

2.2 Semantics

We now define the operational semantics of the applied pi calculus by the means of two relations: structural equivalence and internal reductions. *Structural equivalence* ($\equiv$) is the smallest equivalence relation closed under α-conversion of both bound names and variables and application of evaluation contexts such that:

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

$$\textsc{New-Par} \quad A \mid \nu u.B \ \equiv \ \nu u.(A \mid B) \quad \text{if } u \notin \text{fn}(A) \cup \text{fv}(A)$$

Internal reduction ($\rightarrow$) is the smallest relation closed under structural equivalence, application of evaluation contexts and such that

$$\begin{array}{lll}
\text{COMM} & \overline{c}\langle x\rangle.P \mid c(x).Q \rightarrow P \mid Q & \\
\text{THEN} & \text{if } M = M \text{ then } P \text{ else } Q \rightarrow P & \\
\text{ELSE} & \text{if } M = N \text{ then } P \text{ else } Q \rightarrow Q & \text{if } M, N \text{ ground and } M \neq_E N
\end{array}$$

2.3 Notational conventions

By convention we assume a signature contains: the binary function pair; the unary functions fst, snd; and the constant ε. Moreover, the associated equational theory includes the equations $\mathsf{fst}(\mathsf{pair}(x, y)) = x$ and $\mathsf{snd}(\mathsf{pair}(x, y)) = y$.

Arbitrary length tuples can be constructed as $\mathsf{pair}(x_1, \mathsf{pair}(x_2, \ldots, \mathsf{pair}(x_n, \varepsilon)))$; which, for convenience, we abbreviate $(x_1, \ldots, x_n)$. We also write M_j for $\mathsf{fst}(\mathsf{snd}(\mathsf{snd}(\ldots \mathsf{snd}(M))))$ where $j \geq 1$ and there are $j-1$ occurrences of snd. We also write sometimes $c(x_1, \ldots, x_n)$ (or $c(\tilde{x})$) for the sequence of inputs $c(x_1). \ldots .c(x_n)$.

To aid readability, we also allow boolean combinations of $M = N$ in conditionals in the "if-then-else" process, and in the output of terms. If ϕ is such a combination, then the output $\overline{c}\langle\phi\rangle.P$ is an abbreviation for "if ϕ then $\overline{c}\langle\mathsf{true}\rangle.P$". Boolean combinations of conditionals in "if" statements are handled as follows. First, the boolean combination is written using only the boolean connectives $\wedge$ and $\neg$. Then $\wedge$ is encoded using nested "if" processes; and negation ($\neg$) is encoded by swapping "then" and "else" branches.

2.4 Events and reachability assertions

For the purpose of protocol analysis processes are annotated with *events* which mark important actions performed by the protocol which do not otherwise affect behaviour. We adopt the formalism presented by Abadi, Blanchet & Fournet [1] to capture events. Events are modelled as outputs $\overline{\mathtt{f}}\langle M\rangle$ where $\mathtt{f} \in \mathcal{F}$ is an *"event channel"*: a name in a particular set $\mathcal{F}$ disjoint from the set of ordinary channels a, b, c. Message input on event channels must use *"event variables"* e, e'.

We assume that protocols should be executed in the presence of a so-called Dolev-Yao adversary [15]. The adversary is permitted to input $\mathtt{f}(e)$ on event channels but is forbidden from using the bound event variable e in any other manner. The former condition prevents processes blocking, whereas the latter ensures the adversary's knowledge cannot be extended by the occurrence of events.

Definition 1 (Adversary). *An* adversary *is a closed process such that, any event channel* $\mathtt{f} \in \mathcal{F}$ *and event variable* e*, only occur in inputs of the form* $\mathtt{f}(e)$.

A reachability assertion is specified as an event $\overline{\mathtt{f}}\langle\tilde{X}\rangle$ where $\tilde{X}$ is a tuple of variables and constants. A process satisfies reachability if there exists an adversary who is able to expose the event (Definition 2). When such an adversary does not exist, we say the process satisfies the unreachability assertion $\overline{\mathtt{f}}\langle\tilde{X}\rangle$.

Definition 2 (Reachability). *The closed process* P *satisfies the* reachability *assertion* $\overline{\mathtt{f}}\langle\tilde{X}\rangle$ *where* $\tilde{X}$ *is a tuple of variables and constants if there exists an adversary* Q *such that* $P \mid Q \rightarrow^* C[\overline{\mathtt{f}}\langle\tilde{X}\rangle.P']$ *for some evaluation context* C *and process* P'.

3 Election verifiability

Election verifiability can be formalised with respect to tests Φ^{IV}, Φ^{UV}, Φ^{EV} corresponding to the three aspects of our formalisation. A protocol is said to be election-verifiable if three such tests exist that satisfy some conditions that we detail below. Each of the tests Φ^{IV}, Φ^{UV}, Φ^{EV} is a predicate which after substitutions from the bulletin board and elsewhere evaluates to true or false. The designers of electronic voting protocols need not explicitly specify these cryptographic tests since our definition considers the existence of tests (perhaps devised after design) which satisfy our conditions. This extends the applicability of our methodology whilst also permitting the scrutiny of tests specified by protocol designers.

3.1 Overview

Individual verifiability. The test Φ^{IV} takes parameters v (a vote), $\tilde{x}$ (a voter's knowledge), y (a voter's public credential) and z (a bulletin board entry). For Φ^{IV} to be a suitable test it must allow a voter to identify her bulletin board entry. Formally we require for all votes s, if the voter with public credential D votes for candidate s, then there exists an execution of the protocol which produces $\tilde{M}$ such that some bulletin board entry B satisfies:

$$\Phi^{IV}\{^{s}/_{v}, ^{\tilde{M}}/_{\tilde{x}}, ^{D}/_{y}, ^{B}/_{z}\} \tag{1}$$

Moreover, the bulletin board entry should determine the vote; that is, for all bulletin board entries B, public credentials D, D' votes s, s' and tuples $\tilde{M}, \tilde{M}'$ we have:

$$\Phi^{IV}\{^{s}/_{v}, ^{\tilde{M}}/_{\tilde{x}}, ^{D}/_{y}, ^{B}/_{z}\} \wedge \Phi^{IV}\{^{s'}/_{v}, ^{\tilde{M}'}/_{\tilde{x}}, ^{D'}/_{y}, ^{B}/_{z}\} \Rightarrow (s = s') \tag{2}$$

This ensures the test will hold for at most one vote. Additionally, individual verifiability requires voters to accept distinct bulletin board entries:

$$\textit{Bulletin board entries are distinct} \tag{3}$$

This condition requires protocol executions to introduce some freshness (e.g. randomness).

Universal verifiability. This property is encapsulated by the test Φ^{UV} which takes parameters v (a vote) and z (a bulletin board entry). Given Φ^{IV}, the test Φ^{UV} is suitable if every bulletin board entry which is accepted by a voter is also accepted by an observer; and the entry is counted by the observer in the correct way. The property requires that for all executions of the protocol producing $\tilde{M}$ with respect to the voter's vote s, if there exists a bulletin board entry B and public credential D such that the voter accepts the bulletin board entry as hers, then the observer also accepts the entry:

$$\Phi^{IV}\{^{s}/_{v}, ^{\tilde{M}}/_{\tilde{x}}, ^{D}/_{y}, ^{B}/_{z}\} \Rightarrow \Phi^{UV}\{^{s}/_{v}, ^{B}/_{z}\} \tag{4}$$

Moreover, the observer counts the vote correctly. That is, for all bulletin board entries B and votes s, s' if the test succeeds for s and s' then they must be votes for the same candidate:

$$\Phi^{UV}\{^{s}/_{v}, ^{B}/_{z}\} \wedge \Phi^{UV}\{^{s'}/_{v}, ^{B}/_{z}\} \Rightarrow (s = s') \tag{5}$$

This ensures that an observer may only count a vote in one way.

We remark that the implication in property (4) is only one way because the adversary is able to construct ballots which would be accepted by an observer; however, such a ballot should not be accepted by a voter. This malicious behaviour can be detected by eligibility verifiability. We also note that this formalization is stronger than what is actually required: here a voter needs to be able to identify the bulletin board entry corresponding to her vote. This is not the case in all protocols, e.g. in the protocol by Juels *et al.* described in Section 5. However, this formalization has the advantage of being amenable to automated verification.

Eligibility verifiability. The property is encoded by the test Φ^{EV} which takes parameters y (a voter's public credential) and z (a bulletin board entry). Given Φ^{IV}, the test Φ^{EV} is considered suitable if it ensures: 1) an observer can attribute a bulletin board entry to a public credential if and only if the corresponding voter would accept that entry as hers; and 2) a bulletin board entry can be attributed to at most one voter. Formally the property requires for all bulletin board entries B and executions of the protocol producing $\tilde{M}$, with respect to the voter's vote s and public voter credential D, the voter accepts B as hers iff the bulletin board entry can be attributed to her public credential:

$$\Phi^{IV}\{^{s}/_{v}, ^{\tilde{M}}/_{\tilde{x}}, ^{D}/_{y}, ^{B}/_{z}\} \Leftrightarrow \Phi^{EV}\{^{D}/_{y}, ^{B}/_{z}\} \tag{6}$$

This ensures that if Φ^{IV} succeeds for a voter then she is assured that her vote is considered eligible by an observer; and if Φ^{EV} succeeds for a public credential then the corresponding voter must have constructed that bulletin board entry. The condition relies upon a relationship between the voter's knowledge $\tilde{M}$ and the voter's public credential D . As for universal verifiability this condition may be too strong as some verifiable protocols do not allow a voter to link a bulletin board entry to her credentials.

The second condition requires that the test must uniquely determine who cast a bulletin board entry (hence we can ensure that at most one ballot per registered credential may appear on the bulletin board); that is, for all bulletin board entries B and public voter credentials D, D' if the test succeeds for D and D' then the credentials are equivalent:

$$\Phi^{EV}\{^{D}/_{y}, ^{B}/_{z}\} \wedge \Phi^{EV}\{^{D'}/_{y}, ^{B}/_{z}\} \Rightarrow (D = D') \tag{7}$$

This property enables re-vote elimination. The concept of re-voting is particularly useful since it is used by some protocols to provide coercion resistance. In such protocols re-vote elimination is performed with respect to a publicly defined policy to ensure voters vote at most once. Finally, eligibility verifiability also requires that voters must have unique public credentials:

$$\textit{Public voter credentials are distinct} \tag{8}$$

3.2 Verifying an election

Voters and observers check the outcome of the election by performing the tests Φ^{IV}, Φ^{UV} and Φ^{EV} on the bulletin board and data derived from elsewhere (for example, learnt by the voter during an execution of the protocol). For individual verifiability, each voter should be in possession of her vote s, public credential D and $\tilde{M}$ representing the knowledge learnt during an execution of the protocol, such that there exists $j \in [1, |\tilde{B}|]$ satisfying the test $\Phi^{IV}\{{}^{s}/_{v}, {}^{\tilde{M}}/_{\tilde{x}}, {}^{D}/_{y}, {}^{B_j}/_{z}\}$. For universal verifiability, the bulletin board must be such that the observer can map the ballots to the votes appearing in the election outcome. That is, there exists a bijective function $f : \{1, \ldots, |\tilde{B}|\} \rightarrow \{1, \ldots, |\tilde{B}|\}$, such that for all $j \in [1, |\tilde{B}|]$, the test $\Phi^{UV}\{{}^{s_j}/_{v}, {}^{B_{f(j)}}/_{z}\}$ holds. Similarly, for eligibility verifiability to hold an observer must be able to map each public credential to a bulletin board entry. That is, there exists a bijective function $g : \{1, \ldots, |\tilde{B}|\} \rightarrow \{1, \ldots, |\tilde{B}|\}$ such that for all $j \in [1, |\tilde{B}|]$ the test $\Phi^{EV}\{{}^{D_j}/_{y}, {}^{B_{g(j)}}/_{z}\}$ holds.

Compatibility with privacy. Some electronic voting protocols utilise mixnets to obtain privacy. For simplicity we omit formalising the security of mixnets and hence omit modelling the mix. This clearly violates privacy properties. However, since mixnets are verifiable, privacy and election verifiability properties may coexist in practice.

4 Election verifiability in the applied pi calculus

A voting protocol is captured by a voter process V and a process K modelling administrators whom are required to be honest for the purpose of election verifiability. Dishonest administrators need not be explicitly modelled since they are part of the adversarial environment. The process K is assumed to publish public voter credentials; and is commonly responsible for the distribution of voter keys. Channels $\tilde{a}$ are assumed to be private and appear in both V, K. In addition, we consider a context A which performs setup duties; for example, the instantiation of keys for honest administrators. Dishonest administrator keys are modelled as free names. Definition 3 formalises a voting process specification accordingly. The definition allows us to analyse election verifiability with respect to an unbounded number of voters and arbitrarily many candidates.

Definition 3 (Voting process specification). *A voting process specification is a tuple $\langle A, V, K[\bar{c}\langle D\rangle], \tilde{a}\rangle$ where V is a linear process, A and K are contexts such that V, A, K do not contain any occurrence of event channels and event variables. The term D models public voter credentials. The variable $v \in fv(V)$ refers to the value of the vote, $v \notin (bv(A) \cup bv(V))$, $c \notin (\tilde{a} \cup bn(A[V \mid K]))$ and $(fv(V)\backslash\{v\} \cup fv(K)) \subseteq bv(A)$.*

We also suppose that the votes are generated by a special vote generation process G which "chooses" the vote for a given voter and sends it to the voter on a channel name b. A typical vote generation process would be

$$G \mathrel{\hat{=}} !\nu\, s.((!\bar{b}\langle s\rangle) \mid \bar{c}\langle s\rangle)$$

This process models the generation of all possible votes. Intuitively, the channel b allows the voter to select her vote v and the nested replication allows several voters to choose the same vote s, while other voters may vote differently. Each vote s is also made available to the environment by publishing it on the channel c. Unless specified differently we suppose in the remaining that G is defined as above. Given a voting process specification $\langle A, V, K[\overline{c}\langle D\rangle], \tilde{a}\rangle$ and a vote generation process G we can build the process modelling the voting protocol VP as follows:

$$\mathsf{VP} \mathrel{\widehat{=}} \nu\, b.(A[!\nu\, \tilde{a}.((b(v).V) \mid K[\overline{c}\langle D\rangle])] \mid G)$$

where $b \notin (\tilde{a} \cup \mathrm{fn}(A[V \mid K]) \cup \mathrm{bn}(A))$. Observe that since the key generation process is under replication it is possible to construct unique credentials for each voter.

The formalisation of election verifiability (Definition 5) can naturally be expressed as reachability assertions [25, 8] associated with the properties 1-8 of §3.1 which relate to tests $\Phi^{IV}, \Phi^{UV}, \Phi^{EV}$. First the tests must be incorporated into an augmented voting process (Definition 4).

Definition 4 (Augmented voting process). *Given a voting process specification $\langle A, V, K[\overline{c}\langle D\rangle], \tilde{a}\rangle$, a vote generation process G and tests $\Phi^{IV}, \Phi^{UV}, \Phi^{EV}$ the* augmented voting process *is defined as* $\mathsf{VP}^{+} = \nu\, b.(A[!\nu\, \tilde{a}, b'.(\widehat{V} \mid \widehat{K})] \mid G \mid P) \mid Q \mid R$ *where*

$$\begin{array}{ll}
\widehat{V} & = b(v).V \circ c(z).b'(y).(\overline{\mathtt{pass}}\langle(\Phi^{IV}, z)\rangle \mid \overline{\mathtt{fail}}\langle\psi\rangle) \\
\widehat{K} & = K[\overline{b'}\langle D\rangle \mid \overline{\mathtt{cred}}\langle D\rangle \mid \overline{c}\langle D\rangle] \\
P & = b(v').b(v'').c(\tilde{x}').c(\tilde{x}'').c(y').c(y'').c(z').\overline{\mathtt{fail}}\langle\phi' \vee \phi'' \vee \phi'''\rangle \\
Q & = \mathtt{pass}(e).\mathtt{pass}(e').\overline{\mathtt{fail}}\langle e_1 \wedge e'_1 \wedge (e_2 = e'_2)\rangle \\
R & = \mathtt{cred}(e).\mathtt{cred}(e').\overline{\mathtt{fail}}\langle e = e'\rangle
\end{array}$$

$$\begin{array}{ll}
\psi & = (\Phi^{IV} \wedge \neg\, \Phi^{UV}) \vee (\Phi^{IV} \wedge \neg\, \Phi^{EV}) \vee (\neg\, \Phi^{IV} \wedge \Phi^{EV}) \\
\phi' & = \Phi^{IV}\{^{v'}/_{v}, ^{\tilde{x}'}/_{\tilde{x}}, ^{y'}/_{y}, ^{z'}/_{z}\} \wedge \Phi^{IV}\{^{v''}/_{v}, ^{\tilde{x}''}/_{\tilde{x}}, ^{y''}/_{y}, ^{z'}/_{z}\} \wedge \neg(v' = v'') \\
\phi'' & = \Phi^{UV}\{^{v'}/_{v}, ^{z'}/_{z}\} \wedge \Phi^{UV}\{^{v''}/_{v}, ^{z'}/_{z}\} \wedge \neg(v' = v'') \\
\phi''' & = \Phi^{EV}\{^{y'}/_{y}, ^{z'}/_{z}\} \wedge \Phi^{EV}\{^{y''}/_{y}, ^{z'}/_{z}\} \wedge \neg(y' =_E y'')
\end{array}$$

such that $\mathtt{fail}, \mathtt{pass}, \mathtt{cred}$ *are event channels,* $\tilde{x} = (bv(V) \cap (fv(Riv)\backslash\{z\}))$*, and* b, b' *are fresh, that is,* $b, b' \notin (\tilde{a} \cup fn(A[V \mid K]) \cup bn(A[V \mid K]))$*.*

The augmented voting process extends V to bind the voter's intended vote v, assigns the voter's public credential to y and introduces a claimed bulletin board entry z. As in the non-augmented process, the process $!\nu\, s.((!\overline{b}\langle s\rangle) \mid \overline{c}\langle s\rangle)$ produces candidates for whom the voters are allowed to vote. The number of candidates and for whom each voter casts her vote is controlled by the adversarial environment. The events capture the desired reachability assertions. That is, reachability of $\overline{\mathtt{pass}}\langle(\Phi^{IV}, z)\rangle$ captures propositional property 1 of §3.1; unreachability of $\overline{\mathtt{fail}}\langle\psi\rangle$ models propositional properties 4 and 6 of §3.1; and unreachability of $\overline{\mathtt{fail}}\langle\phi' \vee \phi'' \vee \phi'''\rangle$ denotes properties 2, 5 and 7 of §3.1. The universal quantifiers of the propositional properties 2,4-7 are captured by allowing the adversary to input the required parameters. Process Q exploits communication on event channel `pass` to detect the scenario in which two voters accept the same bulletin board entry, hence capturing property 3 of §3.1, by the unreachability of the event $\overline{\texttt{fail}}\langle e_1 \wedge e_1' \wedge (e_2 = e_2')\rangle$ where e_1 and e_1' capture whether two distinct voters accept the bulletin boards e_2 and e_2'. Similarly, communication on the event channel `cred` is used to detect the situation in which two voters are assigned the same public credential, thus modelling property 8 of §3.1 by the unreachability of $\overline{\texttt{fail}}\langle e = e'\rangle$.

Definition 5 (Election verifiability). *A voting process specification* $\langle A, V, K[\overline{c}\langle D\rangle], \tilde{a}\rangle$ *satisfies* election verifiability *if there exists tests* $\Phi^{IV}, \Phi^{UV}, \Phi^{EV}$ *such that the augmented voting process* VP^+ *and tests satisfy the following conditions:*

1. VP^+ *satisfies the unreachability assertion:* $\overline{\texttt{fail}}\langle \mathsf{true}\rangle$.
2. VP^+ *satisfies the reachability assertion:* $\overline{\texttt{pass}}\langle(\mathsf{true}, x)\rangle$.
3. *The tests* $\Phi^{IV}, \Phi^{UV}, \Phi^{EV}$ *satisfy the following constraints:*
   - $fv(\Phi^{IV}) \subseteq bv(V) \cup \{v, y, z\}$
   - $fv(\Phi^{UV}) \subseteq \{v, z\}$
   - $fv(\Phi^{EV}) \subseteq \{y, z\}$
   - $(fn(\Phi^{IV}) \cup fn(\Phi^{UV}) \cup fn(\Phi^{EV})) \cap bn(\mathsf{VP}^+) = \emptyset$

We remark that the restriction $\mathrm{fn}(\Phi^{IV}) \cap \mathrm{bn}(\mathsf{VP}^+) = \emptyset$ does not lose generality, since restricted names "$\nu\, n$" can be referenced by variables as follows: "$\nu\, n.$let $x\_n = n$ in".

Many protocols in literature do not provide eligibility verifiability. We therefore define a *weakly augmented voting process* and *weak election verifiability* to capture only individual and universal verifiability. A *weakly augmented voting process* is defined as an augmented voting process but with $R = 0$, $\psi = (\Phi^{IV} \wedge \neg\, \Phi^{UV})$ and replacing $\overline{\texttt{fail}}\langle \phi' \vee \phi'' \vee \phi'''\rangle$ with $\overline{\texttt{fail}}\langle \phi' \vee \phi''\rangle$. The definition of *weak election verifiability* is obtained by omitting conditions on Φ^{EV} from Definition 5.

5 Case studies

We demonstrate the applicability of our methodology by analysing electronic voting protocols from literature. The ProVerif tool [8] has been used for automation and our input scripts are available online[5]. ProVerif's ability to reason with reachability assertions is sound (when no trace is found the protocol is guaranteed to satisfy the unreachability assertion) but not complete (false reachability traces may be found). As a consequence reachability traces output by ProVerif for Condition 2 of Definition 5 must be checked by hand. In this paper all such traces correspond to valid reachable states.

[5] `http://www.bensmyth.com/publications/10arspa/`

5.1 Postal ballot protocol

Description. Consider an electronic variant of a *"postal ballot"* (or *"mail-in ballot"*) protocol whereby a voter receives her private signing key sk_V from a keying authority, constructs her signed ballot and sends it to the bulletin board. The keying authority is also responsible for publishing the voter's public verification key $\mathsf{pk}(sk_V)$ which will serve as her public credential. The protocol does not satisfy all of the desirable electronic voting properties; but it should certainly provide election verifiability.

Formalisation in applied pi. The corresponding voting process specification is given by $\langle A, V, K[\overline{c}\langle D\rangle], (a)\rangle$ where

$$\begin{array}{ll} A \mathrel{\widehat{=}} \_ & K[\overline{c}\langle D\rangle] \mathrel{\widehat{=}} \nu\, sk_V.(\overline{a}\langle sk_V\rangle \mid \overline{c}\langle D\rangle) \\ V \mathrel{\widehat{=}} a(x).\overline{c}\langle(v, \mathsf{sign}(v,x))\rangle & D \mathrel{\widehat{=}} \mathsf{pk}(sk_V) \end{array}$$

We model digital signatures (without message recovery) by the equation

$$\mathsf{checksign}(\mathsf{sign}(x,y), x, \mathsf{pk}(y)) = \mathsf{true}$$

The resulting (non-augmented) voting process is then defined as

$$\begin{array}{rl} \mathsf{VP}_{\mathsf{postal}} \mathrel{\widehat{=}} \nu b.\big(& !\nu a.\ (b(v).a(x).\overline{c}\langle(v, \mathsf{sign}(v,x))\rangle \\ & \qquad\qquad \mid \nu\, sk_V.(\overline{a}\langle sk_V\rangle \mid \overline{c}\langle \mathsf{pk}(sk_V)\rangle)) \\ & \mid\ !\nu s.(!\overline{b}\langle s\rangle \mid \overline{c}\langle s\rangle)\big) \end{array}$$

Analysis. The augmented voting process VP^+ can be derived with respect to tests:

$$\begin{array}{l} \Phi^{IV} \mathrel{\widehat{=}} z =_E (sign(v,x), v, pk(x)) \\ \Phi^{UV} \mathrel{\widehat{=}} \mathsf{checksign}(z_1, z_2, z_3) =_E \mathsf{true} \wedge v =_E z_2 \\ \Phi^{EV} \mathrel{\widehat{=}} \mathsf{checksign}(z_1, z_2, z_3) =_E \mathsf{true} \wedge y =_E z_3 \end{array}$$

ProVerif is able to automatically verify the protocol satisfies election verifiability.

5.2 Protocol due to Fujioka, Okamoto & Ohta

Description. The FOO protocol [16] involves voters, a registrar and a tallier. The protocol relies on a commitment scheme and blind signatures which we model by the following equational theory.

$$\begin{array}{c} \mathsf{checksign}(\mathsf{sign}(x,y), x, \mathsf{pk}(y)) = \mathsf{true} \\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x,y), z), y) = \mathsf{sign}(x,z) \\ \mathsf{unblind}(\mathsf{blind}(x,y), y) = x \\ \mathsf{open}(\mathsf{commit}(x,y), y) = x \end{array}$$

The voter first computes her ballot as a commitment to her vote $m' = \mathsf{commit}(v, r)$ and sends the signed blinded ballot $\mathsf{sign}(\mathsf{blind}(m', r'), sk_V)$ to the registrar. The registrar checks the signature belongs to an eligible voter and returns $\mathsf{sign}(\mathsf{blind}(m', r'), sk_R)$ the blind signed ballot. The voter verifies that this input (variable bsb in the process V below) corresponds to the registrar's signature and unblinds the message to recover her ballot signed by the registrar $m = \mathsf{sign}(m', sk_R)$. The voter then posts her signed ballot to the bulletin board. Once all votes have been cast the tallier verifies all the entries and appends an identifier l to each valid record[6]. The voter then checks the bulletin board for her entry, the triple (l, m', m), (modelled in V below by the input in variable bbe) and appends the commitment factor r. Finally, using r the tallier opens all of the ballots and announces the declared outcome. The protocol claims to provide individual and universal verifiability but does not consider eligibility verifiability.

Formalisation in applied pi. The voting process specification is $\langle \_, V, K[\overline{c}\langle \mathsf{pk}(x)\rangle], (a)\rangle$ where $K = c(x).(\overline{a}\langle x\rangle \mid \_)$ and V is defined as follows:

$$\begin{aligned}
V = {} & \nu\, r.\text{let } x\_r = r \text{ in} \\
& \nu\, r'.\text{let } x\_r' = r' \text{ in} \\
& a(x). \\
& \text{let } m' = \mathsf{commit}(v, r) \text{ in} \\
& \overline{c}\langle(\mathsf{pk}(x), \mathsf{blind}(m', r'), \mathsf{sign}(\mathsf{blind}(m', r'), x))\rangle. \\
& c(bsb). \\
& \text{if } \mathsf{checksign}(bsb, \mathsf{blind}(m', r'), \mathsf{pk}(sk_R)) = \mathsf{true} \text{ then} \\
& \text{let } m = \mathsf{unblind}(bsb, r') \text{ in} \\
& \overline{c}\langle(m', m)\rangle. \\
& c(bbe). \\
& \text{if } m' = bbe_2 \wedge m = bbe_3 \text{ then} \\
& \overline{c}\langle(bbe_1, r)\rangle
\end{aligned}$$

Analysis. Let tests Φ^{IV}, Φ^{UV} be defined as follows:

$$\begin{aligned}
\Phi^{IV} & \mathrel{\widehat{=}} z =_E \langle bbe_1, \mathsf{commit}(v, x\_r), \mathsf{unblind}(bsb, x\_r'), x\_r, v\rangle \\
& \quad \wedge \mathsf{checksign}(z_3, z_2, \mathsf{pk}(sk_R)) =_E \mathsf{true} \\
\Phi^{UV} & \mathrel{\widehat{=}} z_2 =_E \mathsf{commit}(z_5, z_4) \wedge \mathsf{checksign}(z_3, z_2, \mathsf{pk}(sk_R)) =_E \mathsf{true} \wedge z_5 =_E v
\end{aligned}$$

ProVerif enables automatic verification of election verifiability with respect to weak election verifiability.

5.3 Protocol due to Juels, Catalano & Jakobsson and Clarkson, Chong & Myers

Description. The protocol due to Juels, Catalano & Jakobsson [19], which has been implemented by Clarkson, Chong & Myers as Civitas [13, 12], involves voters, registrars and talliers.

The registrars announce the candidate list $\tilde{s} = (s_1, \ldots, s_l)$ and provide an *anonymous credential* k to each legitimate voter. For each such credential an encrypted version $\mathsf{penc}(k, r'', \mathsf{pk}(sk_T))$ is published on the bulletin board.

[6] The value l is used for practical purposes only; it does not affect security.

Each voter selects her vote $s \in \tilde{s}$ and computes the ciphertexts $M = \mathsf{penc}(s, r, \mathsf{pk}(sk_T))$ and $M' = \mathsf{penc}(k, r', \mathsf{pk}(sk_T))$. The first ciphertext contains her vote and the second contains her credential. In addition, the voter constructs a signature proof of knowledge, that is, a non-interactive zero-knowledge proof of knowledge, demonstrating the correct construction of her ciphertexts and that she has chosen a valid candidate; that is, $s \in \tilde{s}$. The voter posts her ciphertexts and signature proof of knowledge to the bulletin board.

After some predefined deadline the outcome is computed as follows. First, the talliers discard any entries for which signature proofs of knowledge do not hold and eliminate re-votes by performing pairwise plaintext equivalence tests[7] on all the ciphertexts containing voting credentials posted by the voter. Re-vote elimination is performed in a verifiable manner with respect to some publicly defined policy, e.g. keeping the last vote. Then, the talliers perform a verifiable re-encryption mix on the votes and credentials, keeping the link between each such pair. The aim of this mix is that the voter herself cannot trace her vote anymore which allows the protocol to achieve coercion-resistance. After the mix all invalid credentials are discarded using plaintext equivalence tests between the entries posted to the bulletin board by the mix and those published by the registrar. Finally, the talliers perform a verifiable decryption and publish the result.

Formalisation in applied pi. We model a simplified version of the protocol described above which omits the mix. The resulting protocol is not coersion-resistant, but still provides anonymity. This change is made because our definition is too strong to hold on the complete protocol. We discuss future work regarding a more general definition in our conclusion.

The formalisation of signature proofs of knowledge in the applied pi calculus that we adopt is due to Backes *et al.* [5, 6]. A signature proof of knowledge is a term $\mathsf{spk}_{i,j}(\tilde{U}, \tilde{V}, F)$ where $\tilde{U} = (U_1, \ldots, U_i)$ denotes the witness (or private component), $\tilde{V} = (V_1, \ldots, V_j)$ defines the public parameters and F is a formula over those terms. More precisely F is a term without names or variables, but includes distinguished constants α_k, β_l where $k, l \in \mathbb{N}$. The constants α_k, β_l in F denote placeholders for the terms $U_k \in \tilde{U}$, $V_l \in \tilde{V}$ used within a signature of knowledge $\mathsf{spk}_{i,j}(\tilde{U}, \tilde{V}, F)$. For example, the signature proof of knowledge used by voters in the Juels, Catalano & Jakobsson voting protocol [19] demonstrates possession of a vote s, credential k and randomisation factors r, r' such that $M = \mathsf{penc}(s, r, \mathsf{pk}(sk_T))$, $M' = \mathsf{penc}(k, r', \mathsf{pk}(sk_T))$ and $s \in \tilde{s}$; that is, the proof shows the ciphertexts are correctly formed and s is a valid candidate. This can be captured by $\mathsf{spk}_{4,3+l}((s, r, k, r'), (M, M', \mathsf{pk}(sk_T), s_1, \ldots, s_l), \mathcal{F})$ where $\mathcal{F}$ is defined as $\beta_1 = \mathsf{penc}(\alpha_1, \alpha_2, \beta_3) \wedge \beta_2 = \mathsf{penc}(\alpha_3, \alpha_4, \beta_3) \wedge (\alpha_1 = \beta_4 \vee \ldots \vee \alpha_1 = \beta_{4+l})$. A term $\mathsf{spk}_{i,j}(\tilde{U}, \tilde{V}, F)$ represents a valid signature if the term obtained by substituting U_k, V_l for the corresponding α_k, β_l evaluates to true. Verification

[7] A *plaintext equivalence test* [17] is a cryptographic predicate which allows the comparison of two ciphertexts. The test returns true if the ciphertexts contain the same plaintext.

of such a statement is modelled by the function $\mathsf{ver}_{i,j}$. The equational theory includes the following equations over all $i, j \in \mathbb{N}$, tuples $\tilde{x} = (x_1, \ldots, x_i), \tilde{y} = (y_1, \ldots, y_j)$ and formula F which is a ground term over $\Sigma \cup \{\alpha_k, \beta_l \mid k \leq i,\ l \leq j\}$ without any names:

$$\begin{aligned}
&\mathsf{public}_\mathsf{p}(\mathsf{spk}_{i,j}(\tilde{x}, \tilde{y}, F)) \ = y_p \text{ where } p \in [1, j] \\
&\mathsf{formula}(\mathsf{spk}_{i,j}(\tilde{x}, \tilde{y}, F)) = F
\end{aligned}$$

In addition, we define equations such that $\mathsf{ver}_{i,j}(F, \mathsf{spk}_{i,j}(\tilde{U}, \tilde{V}, F')) =_E \mathsf{true}$ if $F =_E F'$ and $F\{{}^{U_1}/_{\alpha_1}, \ldots, {}^{U_i}/_{\alpha_i}, {}^{V_1}/_{\beta_1}, \ldots, {}^{V_j}/_{\beta_j}\}$ holds where $i = |\tilde{U}|$, $j = |\tilde{V}|$ and F, F' are ground terms over $\Sigma \cup \{\alpha_k, \beta_l \mid k \leq i,\ l \leq j\}$ without names. We omit the details of these equations which are similar to [5, 6].

The protocol uses a variant of the ElGamal encryption scheme [19]. In addition to the standard equation $\mathsf{dec}(\mathsf{penc}(x, y, \mathsf{pk}(z)), z) = x$, this scheme allows the construction of a dedicated decryption key for a particular ciphertext which is modelled by the equation:

$$\mathsf{dec}(\mathsf{penc}(x, y, \mathsf{pk}(z)), \mathsf{commit}(\mathsf{penc}(x, y, \mathsf{pk}(z)), z)) = x$$

Verifiable decryption can then be achieved using the signature proof of knowledge $\mathsf{spk}_{1,3}((\alpha_1), (\beta_1, \beta_2), \mathcal{F}')$ where $\mathcal{F}'$ is given by $\beta_1 = \mathsf{commit}(\beta_2, \alpha_1)$. The proof shows that if $\beta_2 = \mathsf{penc}(M, N, \mathsf{pk}\alpha_1)$ for some terms M, N, then β_1 is a decryption key for β_2. Finally, plaintext equivalence tests are modelled by the equation

$$\begin{aligned}
&\mathsf{pet}(\mathsf{penc}(x, y, pk(z)), \mathsf{penc}(x, y', \mathsf{pk}(z)), \\
&\qquad\qquad \mathsf{petkey}(\mathsf{penc}(x, y, \mathsf{pk}(z)), \mathsf{penc}(x, y', \mathsf{pk}(z)), z)) = \mathsf{true}
\end{aligned}$$

The voting process specification can now be defined as $\langle A, V, K[\overline{c}\langle D\rangle], (a)\rangle$ where public credential $D = \mathsf{penc}(k, r'', \mathsf{pk}(sk_T))$ and A, V, K are specified as follows:

$$\begin{aligned}
A \ &= \nu\, sk_T.(\overline{c}\langle \mathsf{pk}(sk_T)\rangle \mid (!A') \mid \_) \\
V \ &= \nu\, r.\text{let } x\_r = r \text{ in } \nu\, r'.\text{let } x\_r' = r' \text{ in } a(x\_k). \\
&\quad \text{let } x\_s_1 = s_1 \text{ in } \ldots \text{ let } x\_s_l = s_l \text{ in} \\
&\quad \text{let } x\_pkT = \mathsf{pk}(sk_T) \text{ in} \\
&\quad \text{let } M = \mathsf{penc}(v, r, \mathsf{pk}(sk_T)) \text{ in} \\
&\quad \text{let } M' = \mathsf{penc}(x\_k, r', \mathsf{pk}(sk_T)) \text{ in} \\
&\quad \text{let } N = \mathsf{spk}_{4,3+l}((v, r, x\_k, r'), (M, M', \mathsf{pk}(sk_T), s_1, \ldots, s_l), \mathcal{F}) \text{ in} \\
&\quad \overline{c}\langle (M, M', N)\rangle \\
K \ &= \nu\, k.\nu\, r''.(\overline{a}\langle k\rangle \mid \_) \\
A' &= c(y).\text{if } \mathsf{ver}_{4,3+l}(\mathcal{F}, y) = \mathsf{true} \text{ then} \\
&\quad c(z).\text{if } \mathsf{pet}(z, \mathsf{public}_2(y), \mathsf{petkey}(z, \mathsf{public}_2(y), sk_T)) = \mathsf{true} \text{ then} \\
&\quad \overline{c}\langle \mathsf{spk}_{1,2}((sk_T), (\mathsf{commit}(\mathsf{public}_1(y), sk_T), \mathsf{public}_1(y)), \mathcal{F}')\rangle). \\
&\quad \overline{c}\langle \mathsf{petkey}(z, \mathsf{public}_2(y), sk_T)\rangle
\end{aligned}$$

For simplicity we consider a single registrar K and tallier A' who are assumed to be honest. Moreover, we assume the existence of a secure mix protocol and hence do not model or verify the shuffle. The registrar process is standard and is responsible for publishing the public voter credentials. Process A' captures the tallier's responsibility to provide suitable keys for plaintext equivalence tests used for eligibility checking and performing verifiable decryption of honestly constructed ballots.

Analysis. The protocol is dependent on the candidate list and therefore cannot be verified with respect to an arbitrary number of candidates. We must therefore restrict the adversarial environment to consider a fixed number of candidates. We therfore define the vote generation process as $G \mathrel{\widehat{=}} (!\bar{b}\langle s_1 \rangle) \mid \ldots \mid (!\bar{b}\langle s_l \rangle)$ where $s_1, \ldots, s_l$ are free names representing the candidates for whom voters may cast their votes. Let the tests $\Phi^{IV}, \Phi^{UV}, \Phi^{EV}$ be defined as follows:

$$\begin{aligned}
\Phi^{IV} &\mathrel{\widehat{=}} \phi' \wedge z_1 =_E \mathsf{spk}_{4,3+l}((v, x\_r, x\_k, x\_r'), (M, M', x\_pkT, x\_s_1, \ldots, x\_s_l), \mathcal{F})\\
\Phi^{UV} &\mathrel{\widehat{=}} \phi \ \wedge \mathsf{dec}(\mathsf{public}_2(z_2), \mathsf{public}_1(z_2)) =_E v\\
\Phi^{EV} &\mathrel{\widehat{=}} \phi' \wedge \mathsf{ver}_{4,3+l}(\mathcal{F}, z_1) =_E \mathsf{true}
\end{aligned}$$

$$\begin{aligned}
\phi &\mathrel{\widehat{=}} \mathsf{ver}_{1,2}(\mathcal{F}', z_2) =_E \mathsf{true} \wedge \mathsf{public}_1(z_1) =_E \mathsf{public}_2(z_2)\\
\phi' &\mathrel{\widehat{=}} \phi \wedge \mathsf{pet}(y, \mathsf{public}_2(z_1), z_3) =_E \mathsf{true}
\end{aligned}$$

where $M = \mathsf{penc}(v, x\_r, x\_pkT)$ and $M' = \mathsf{penc}(x\_k, x\_r', x\_pkT)$. The augmented voting process can now be derived with respect to the size of the candidate list l. Using ProVerif in association with a PHP script that generates ProVerif scripts for different values of l, the protocol can be successfully verified to satisfy election verifiability with respect to $l \in [1, 100]$.

6 Conclusion

This paper presents a preliminary formal definition of election verifiability for electronic voting protocols. The idea of tests for individual, universal and eligibility verifiability (and the associated acceptability conditions) is independent of any particular formalism. We instantiate this idea in terms of reachability assertions in the context of the applied pi calculus. The definition is suitable for automated reasoning using the ProVerif software tool, which we demonstrate by providing the code used for our analysis of the protocol by Fujioka, Okamoto & Ohta [16] and a variant of the one by Juels, Catalano & Jakobsson [19] and Clarkson, Chong & Myers [13, 12].

The definition is work in progress, because it is currently insufficiently general to take account of protocols that use homomorphic encryption (such as Helios 2.0 [3]). Moreover, it is likely that the 'pointwise' nature of the tests Φ^{UV} and Φ^{EV} is too strong for some protocols; more likely, the observer performs a test on the whole bulletin board at once, rather than a separate test on each of its entries. In future work, we intend to generalise our definitions of universal verifiability and eligibility verifiability to work with a greater variety of voting systems. We also intend to study more carefully the relation between verifiability and coercion resistance. Finally, we aim to formalise the security of re-vote elimination policies, present in many realistic systems, which we omitted from our case studies.

Acknowledgements

We are particularly grateful to Michael Clarkson for careful reading of an earlier draft, and for his perceptive questions and comments. In addition, we would like to thank the anonymous WISSec'09 reviewers for helpful remarks.

References

1. M. Abadi, B. Blanchet, and C. Fournet. Just fast keying in the pi calculus. *ACM Transactions on Information and System Security (TISSEC)*, 10(3):1–59, July 2007.
2. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *POPL'01: Proceedings of the 28th ACM Symposium on Principles of Programming Languages*, pages 104–115, New York, USA, 2001. ACM.
3. B. Adida, O. de Marneffe, O. Pereira, and J.-J. Quisquater. Electing a university president using open-audit voting: Analysis of real-world use of Helios. In *Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE)*, 2009.
4. M. Backes, C. Hritcu, and M. Maffei. Automated verification of remote electronic voting protocols in the applied pi-calculus. In *CSF'08: Proceedings of the 21st IEEE Computer Security Foundations Symposium*, pages 195–209, Washington, USA, 2008. IEEE.
5. M. Backes, M. Maffei, and D. Unruh. Zero-Knowledge in the Applied Pi-calculus and Automated Verification of the Direct Anonymous Attestation Protocol. Cryptology ePrint Archive: Report 2007/289, July 2007.
6. M. Backes, M. Maffei, and D. Unruh. Zero-Knowledge in the Applied Pi-calculus and Automated Verification of the Direct Anonymous Attestation Protocol. In *S&P'08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 202–215, Washington, DC, USA, 2008. IEEE Computer Society.
7. A. Baskar, R. Ramanujam, and S. P. Suresh. Knowledge-based modelling of voting protocols. In *TARK'07: Proceedings of the 11th International Conference on Theoretical Aspects of Rationality and Knowledge*, pages 62–71, New York, USA, 2007. ACM.
8. B. Blanchet. Automatic verification of correspondences for security protocols. *Journal of Computer Security*, 2009. To appear.
9. D. Bowen. Secretary of State Debra Bowen Moves to Strengthen Voter Confidence in Election Security Following Top-to-Bottom Review of Voting Systems. California Secretary of State, press release DB07:042 `http://www.sos.ca.gov/elections/voting_systems/ttbr/db07_042_ttbr_system_decisions_release.pdf`, August 2007.
10. Bundesverfassungsgericht (Germany's Federal Constitutional Court). Use of voting computers in 2005 Bundestag election unconstitutional. Press release 19/2009 `http://www.bundesverfassungsgericht.de/en/press/bvg09-019en.html`, March 2009.

11. B. Chevallier-Mames, P.-A. Fouque, D. Pointcheval, J. Stern, and J. Traore. On Some Incompatible Properties of Voting Schemes. In *WOTE'06: Proceedings of the International Association for Voting Systems Sciences Workshop on Trustworthy Elections*, 2006.
12. M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a secure voting system. Technical Report 2007-2081, Cornell University, May 2007. Revised March 2008. `http://hdl.handle.net/1813/7875`.
13. M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a secure voting system. In *S&P'08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 354–368, Washington, DC, USA, 2008. IEEE Computer Society.
14. S. Delaune, S. Kremer, and M. D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 2009. To appear.
15. D. Dolev and A. C. Yao. On the security of public key protocols. *Information Theory*, 29:198–208, 1983.
16. A. Fujioka, T. Okamoto, and K. Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *ASIACRYPT'92: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pages 244–251, London, 1992. Springer.
17. M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts. In *ASIACRYPT '00: Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security*, pages 162–177, London, UK, 2000. Springer.
18. A. Juels, D. Catalano, and M. Jakobsson. Coercion-Resistant Electronic Elections. Cryptology ePrint Archive, Report 2002/165, 2002.
19. A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 61–70, New York, NY, USA, 2005. ACM. See also `http://www.rsa.com/rsalabs/node.asp?id=2860`.
20. Ministerie van Binnenlandse Zaken en Koninkrijksrelaties (Netherland's Ministry of the Interior and Kingdom Relations). Stemmen met potlood en papier (Voting with pencil and paper). Press release `http://www.minbzk.nl/onderwerpen/grondwet-en/verkiezingen/nieuws--en/112441/stemmen-met-potlood`, May 2008.
21. Participants of the Dagstuhl Conference on Frontiers of E-Voting. Dagstuhl accord. `http://www.dagstuhlaccord.org/`, 2007.
22. K. Sako and J. Kilian. Secure voting using partially compatible homomorphisms. In *CRYPTO'94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, pages 411–424, London, UK, 1994. Springer-Verlag.
23. M. Talbi, B. Morin, V. V. T. Tong, A. Bouhoula, and M. Mejri. Specification of Electronic Voting Protocol Properties Using ADM Logic: FOO Case Study. In *ICICS'08: Proceedings of the 10th International Conference on Information and Communications Security Conference*, pages 403–418, London, 2008. Springer.
24. UK Electoral Commission. Key issues and conclusions: May 2007 electoral pilot schemes. `http://www.electoralcommission.org.uk/elections/pilots/May2007`.
25. T. Y. C. Woo and S. S. Lam. A semantic model for authentication protocols. In *S&P'93: Proceedings of the 1993 IEEE Symposium on Security and Privacy*, pages 178–194, Washington, DC, USA, 1993. IEEE Computer Society.