Analysing Privacy Analyses

Giampaolo Bella[1], Denis Butin[2], and Hugo Jonker[3,4]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Italy
[2] TU Darmstadt, Germany
[3] Open University of the Netherlands, Heerlen, The Netherlands
[4] Radboud University, Nijmegen, The Netherlands
giamp@dmi.unict.it, dbutin@cdc.informatik.tu-darmstadt.de, hugo.jonker@ou.nl

Abstract

The debate on people's right to privacy and on its meaning is ongoing worldwide, for example in Europe with the newly adopted General Data Protection Regulation. By contrast, works in the area of formal e-voting privacy analysis, which aim at assessing the privacy preservation of a target e-voting system by means of mathematical rigour, appear to have reached a well-known plateau. This plateau is called *indistinguishability.* However, also other works look at privacy from a formal standpoint, though on different grounds. Notable ones are *unlinkability* and *minimal information disclosure.* This paper provides a contrastive argument about the three mentioned approaches by discussing the intuition behind each of them and by assessing their respective pros and cons with the ultimate aim of revamping the privacy debate also at the level of formal analysis.

1 Introduction

The newly adopted EU General Data Protection Regulation 2016/679 enshrines privacy as a fundamental human right, and the international debate that was revolving around its definition has now shifted to the consequences of the new regulation for concrete data processing scenarios. In particular, Article 35 promotes "safeguards, security measures and mechanisms to ensure the protection of personal data", and e-voting was explicitly included in the Article 29 Working Group's 2016–2018 work programme [2].

We believe that such a debate on mechanisms to ensure privacy, in particular in the context of e-voting, must be accompanied by an appropriate debate also at the level of the formal methods that can be used. This paper tackles the three best-known approaches to formal privacy analysis — indistinguishability, unlinkability and minimal information disclosure — puts them into perspective, introduces informal scenarios to comprehend them and provides a contrastive debate to evaluate them.

2 Privacy as Indistinguishability

Kremer and Ryan [14] introduced the modelling of voter privacy and related properties in the applied pi calculus [1], a security protocol modelling language. The applied pi calculus models security protocols as concurrent, interacting processes. The initial idea of seeing voter privacy as observational equivalence (but not specifically in the applied pi calculus) was introduced a decade earlier [21]. In this more recent approach, cryptographic primitives are supported by the applied pi calculus through the definition of new equational theories, which are equivalence relations over terms. For instance, the equational theory used to model the fundamental principle of symmetric cryptographic is written as $\mathrm{dec}(\mathrm{enc}(x, k), k) = x$. Considering two voters, their formalisation reflects whether an outsider can distinguish between two scenarios which are identical, but for the fact that the voters' votes are swapped. Cryptography is seen as a perfect black box, and the Dolev-Yao model is used, i.e. the attacker is omnipotent except for the fact that she cannot decrypt without the adequate key.

2.1 Intuition

The notion behind the formalism of indistinguishability is that an even stronger property than the confidentiality of a certain voter's vote is proven: an outsider may not even detect that votes were cast differently at all. The gist of this approach is a representation of voter privacy as observational equivalence between processes. A voter casting a vote is represented as a process. A security protocol verification software, ProVerif [5], is used to partially mechanise proof techniques for observational equivalence.

Delaune, Kremer and Ryan [9] studied voter privacy, receipt-freeness and coercion-resistance using observational equivalence for all three of these privacy-type properties. Their formalisation can be summed up as follows. *Voter privacy* holds if an outsider cannot obtain information allowing him to distinguish between two situations in which voters' votes are swapped. *Receipt-freeness* holds if an outsider cannot distinguish between the following two situations involving, again, two voters: (1) the targeted voter votes as instructed, and the voter votes differently; (2) the targeted voter does not vote as instructed, but the other voter votes as per the instruction (even though it was not addressed to him). *Coercion-resistance* is defined similarly, but the outsider communicates with the targeted voter during the vote, and can instruct the voter to send crafted messages.

The last two of these properties feature a fundamental difference with voter privacy, because they involve the concept of inability of proof (by the voter, and for the benefit of the attacker, regarding the cast vote). The remainder of our discussion of the indistinguishability approach focuses solely on a discussion of the voter privacy property.

2.2 Machinery

Focusing on the prime example of voter privacy (the weakest property of the tree listed above), some formalism must be introduced so it can be expressed in the applied pi calculus:

- $\{^r/_s\}$ represents an active substitution, replacing the variable s with the term r.
- Processes are described using *names*, which represent atomic data such as a named communication channel. We omit their operational semantics here for space reasons.
- S is an evaluation context, defined as a special voting process with holes instead of two voter processes.
- $fv(A)$ is the set of free variables of A. $fn(A)$ is the set of free names of A. $bn(A)$ is the set of bound names of A.
- The relation $\rightarrow$ is called *internal reduction* and is a relation on extended processes satisfying a number of rules we omit here. The relation $\rightarrow^\alpha$, built on top of $\rightarrow$, allows labelled operations, e.g. α can be the name of an input, the output of a channel name or a basic variable.
- The relation $\approx_l$ is called *labelled bisimilarity.* It is defined as the largest symmetric relation $\mathcal{R}$ on closed extended processes such that if $A\ \mathcal{R}\ B$, then (i) $A \approx_l B$ and (ii) if $A \rightarrow A'$, then $B \rightarrow^* B'$ and $A'\ \mathcal{R}\ B'$ for some B' and (iii) if $A \rightarrow^\alpha A'$ and $fv(\alpha) \subseteq \mathrm{dom}(A)$ and $bn(\alpha)\ \cap fn(B) = \varnothing$, then $B \rightarrow^* \rightarrow^\alpha \rightarrow^* B'$ and $A'\ \mathcal{R}\ B'$ for some B'.

The notion of *frames*, based on extended processes, accounts for the knowledge that processes leak to the environment. Processes may behave differently while still yielding identical frames.

2.3 Findings

With the machinery we just described in place, voter privacy is then defined to hold if the following is true for all possible votes a and b:

Theorem 1 (Voter privacy as labelled bisimilarity in the indistinguishability framework)**.**

$$S[V_A\{{}^{a}/_{v}\} \mid V_B\{{}^{b}/_{v}\}] \approx_l S[V_A\{{}^{b}/_{v}\} \mid V_B\{{}^{a}/_{v}\}]$$

In other words, two processes differing only by swapped votes a and b between two (honest) voters V_A and V_B are observationally equivalent. As seen here, in practice, indistinguishability frameworks use labelled bisimilarity rather than observational equivalence. Since these two relations are known to coincide, the choice of labelled bisimilarity is purely technical — proofs become simpler.

In the framework by Delaune, Kremer and Ryan [9], election officials can be specified as corrupt or honest. Some protocols can be shown to enforce voter privacy in this formalism even with corrupt officials, but not all. As a consequence, statements about voter privacy for protocols are only meaningful if assumptions about the honesty of election officials are made explicit. In the aforementioned paper [9], three e-voting protocols are analysed within the proposed framework: the FOO protocol [11], a later protocol by Okamoto using homomorphic encryption [19] and a simplified version of a vote & go protocol with mixnet randomisation by Lee et al. [16].

The intuitive implication chain from coercion-resistance to voter privacy via receipt-freeness is formally shown to hold.

3 Privacy as Unlinkability

Unlinkability as an approach towards a formal encoding of privacy was abstractly discussed by Langer et al. [15] in the context of electronic voting and was given a semantics precisely in terms of indistinguishability. A practical specification in a formal language was advanced by Butin et al. [8] in particular to handle voter privacy. The latter is reviewed and discussed below to an unprecedented level of detail.

3.1 Intuition

The idea of using unlinkability to model voter privacy is inspired by a real-world scenario, which can be demonstrated as follows. An attacker sits in her mobile laboratory in her van, just outside the voting site of an electronic election. She intercepts the data cable between the site and the local hub, hence she captures the entire election traffic and analyses it. She observes when a specific voter enters the site, so she can guess reasonably well what portion of network traffic corresponds to the vote of that voter. If she manages to distil out a vote, namely a candidate identifier, from that traffic portion, then she has violated the privacy of that vote.

It can be imagined that the attacker will collect relevant message components such as names and keys from the traffic using all means at her disposal, such as splitting concatenated messages and decrypting ciphertexts using the keys that she has. If the analyst's aim is to study how well the protocol is designed to protect the votes, then it can be assumed that encryption cannot be broken; however, the intertwining of cryptographic weaknesses and protocol design flaws is currently being studied in other contexts [3, 6]. Arguably, the attacker will also check whether the components obtained from various messages in the traffic have something in common and, if so, will meaningfully relate the messages. We next show how this threat model can be formalised.

3.2 Machinery

Unlinkability can be defined on an inductive protocol model, which can be built using the Inductive Method. This is briefly reviewed here, and only the details of unlinkability are given. The method is to use Higher-Order Logics to specify a security protocol as an inductive definition and then leverage the induction principle to prove properties of the model. The method was introduced by Paulson [20] and fully fleshed out by Bella [4]. Full mechanisation is achieved with the support of the Isabelle/HOL interactive theorem prover [23]. A Dolev-Yao attacker is also specified as the agent called *Spy*, who can intercept, break down and recompose messages at will but not break encryption. Cryptography is therefore assumed to work perfectly, so that a ciphertext can be opened only if the corresponding encryption key is available.

The formal protocol model is the set of all possible *traces* of the given protocol, each trace being a list of protocol events. Two main functions are *knows* and *analz*. The former expresses all messages that an agent can learn from a trace, and the latter extracts all components of a given set of messages by means of decomposition of clear-texts and decryption of ciphertexts using available keys. Although we omit their full definitions for brevity, we observe that the set *analz*(*knows Spy evs*) expresses all message components, such as keys and nonces, that the Spy can derive from a trace *evs*. This set is crucial both in traditional work to express secrecy and below to build the machinery for unlinkability, as we shall see.

Three protocol events are modelled, respectively *Says* for sending a message to specific recipient, *Gets* for receiving a message, and *Notes* for noting down a message for future use. Here is a trace admitted in the model of an imaginary voting protocol:

$$\begin{array}{l}[\ \ \mathit{Says\ Spy\ O}\ (\mathit{Crypt}\ (\mathit{priSK}\ V)\ (\mathit{Agent\ vote})),\ \ \mathit{Gets\ O}\ (\mathit{Crypt}\ (\mathit{priSK}\ V)\ (\mathit{Agent\ vote})),\\ \ \ \mathit{Says}\ V'\ O\ (\mathit{Crypt}\ (\mathit{priSK}\ V)\ (\mathit{Agent\ vote}')),\ \ \mathit{Says}\ V\ O\ (\mathit{Crypt}\ (\mathit{priSK}\ V)\ (\mathit{Agent\ vote}))\ ]\end{array}$$

Traces are built in reverse order, so the first event here sees voter V cast vote *vote* in a signed form with the election official *O*; then voter V' casts vote *vote*′; then the official receives the first vote; finally the Spy replays the first vote casting message. Note that the correspondence between receiving a message and sending it can be interleaved with other events, that a sent message is not necessarily received, and that the Spy may intercept all traffic and reuse it.

The unlinkability approach rests on the notion of an *association*, namely a link, which is simply a set of elements of type *msg*; therefore, an association has type *msg set*. A fundamental operator is *aanalz*:

```
primrec aanalz :: agent ⇒ event list ⇒ msg set set where
  aanalz_Nil:   aanalz A [] = {}
| aanalz_Cons:
  aanalz A (ev # evs) =
   (if A = Spy then
    (case ev of
      Says A′ B X ⇒
       (if A′ ∈ bad  then aanalz Spy evs
        else if isAnms X
```

```
          then insert  ({Agent B} ∪ (analzplus {X} (analz(knows Spy evs))))
                        (aanalz Spy evs)
          else insert ({Agent B} ∪ {Agent A′ } ∪
                       (analzplus {X} (analz(knows Spy evs)))) (aanalz Spy evs))
    | Gets A′ X ⇒ aanalz Spy evs
    | Notes A′ X ⇒ aanalz Spy evs)
    else aanalz A evs)
```

It takes parameters of type *agent* and *event list*, returning one of type *msg set set*. It therefore captures the set of associations that a given agent can build by observing a trace. The base case of the primitive recursive definition is obvious; the recursive case is less so. It can be seen that only the Spy can build associations, as the symbolic evaluation of the current event *ev* cancels that event out for other agents, and the last line concludes so. The Spy builds associations only if the current event is a *Says* one: this is sound because received messages were also sent, and because notes are intended as off-line records.

The definition specifies further cases. If the sender in the *Says* event is an accomplice of the Spy's, namely it belongs to the set *bad*, then also in this case the evaluation cancels the current event out because the Spy's goal is to violate the privacy of agents who do not collude with her. The next distinction is on whether the current message, which is X in the definition, was sent over an anonymous channel or not, as declared by the predicate *isAnms*, whose definition we omit. If the message is anonymously sent, then the current set of associations *aanalz Spy evs* is extended with a new association $\{Agent\ B\} \cup (analzplus\ \{X\}\ (analz(knows\ Spy\ evs)))$. This association features the recipient B along with all components that the Spy can extract from the current message using her entire knowledge gained from the current trace. The operator *analzplus*, whose definition we omit, takes precisely two parameters: a message to break down as much as possible using keys derived from a set of messages. This is where the set *analz*(*knows Spy evs*) mentioned above comes into play. Therefore, *analzplus* models the brute-forcing of a message in all possible ways except for cryptanalysis.

The only remaining case has the current message as not anonymous. The association that is extracted is the same as the previous case's but also contains the identity of the sender, here A'.

Modelling unlinkability requires another important operator, *asynth*:

```
inductive_set
 asynth :: msg set set ⇒ msg set set
 for as :: msg set set where
  asynth_Build [intro]: ⟦a1 ∈ as; a2 ∈ as; m ∈ a1; m ∈ a2; m ≠ Agent Adm; m ≠ Agent Col⟧
                        ⟹ a1 ∪ a2 ∈ asynth as
```

It can be seen how this operator transforms a set of associations in another one, namely by merging two associations as a new one provided that they have an element in common. Another requirement is that the common element is not an obvious one for the protocol, such as the officials *Adm* and *Col*. This operator models the attacker's ability to compare the associations deriving from different messages and combine them meaningfully, namely when they intersect. A voting protocol might protect voter privacy by distributing the link between a voter and his vote over several messages, so *asynth* empowers the attacker against that.

3.3 Findings

The ultimate aim of the machinery defined above was to build the set *asynth* (*aanalz Spy evs*). It is the set of all possible associations that the Spy can build by observing (the messages circulated on) the trace *evs*. It contains vast and, from a malicious perspective, valuable information. Therefore, given a generic association drawn from that set, the analyst will assess whether that association may contain a voter and his vote. If this is not the case, the informal conclusion will be that the protocol preserves voter privacy.

For example, the main theorem about voter privacy on the FOO protocol [8] is the following:

Theorem 2 (`foo_V_privacy_asynth`).

$$
\begin{aligned}
&[\![\mathit{Says}\ V\ \mathit{Adm}\ \{\!|\mathit{Agent}\ V,\ \mathit{Crypt}\ (\mathit{priSK}\ V)\ (\mathit{Crypt}\ b\ (\mathit{Crypt}\ c\ (\mathit{Nonce}\ \mathit{Nv})))|\!\}\in \mathit{set}\ \mathit{evs};\\
&\ a \in (\mathit{asynth}\ (\mathit{aanalz}\ \mathit{Spy}\ \mathit{evs}));\\
&\ \mathit{Nonce}\ \mathit{Nv} \in a;\ V \notin \mathit{bad};\ V \neq \mathit{Adm};\ V \neq \mathit{Col};\ \mathit{evs} \in \mathit{foo}]\!] \Longrightarrow \mathit{Agent}\ V \notin a
\end{aligned}
$$

The theorem premises bind the main elements. They state that the main event whereby a generic voter V casts his vote, here formalised as nonce *Nv* appears in a generic trace *evs* of the protocol model *foo*. They also specify, that the voter is neither of the protocol authorities, the administrator *Adm* and the collector *Col*. Most importantly, they state an association a that the Spy can derive as just discussed, and assert that V's vote belongs to that association. The theorem concludes that the voter does not belong to the association instead. The proof is omitted because out of the scope of this paper.

There is no notion of a vote *belonging* to a specific voter, namely no function that associates a vote to the voter who cast it. Although it would be easy to define, the conclusion of the theorem as it stands is more generic, hence stronger: no vote can be associated to any voter.

4 Privacy as Minimal Information Disclosure

Privacy has also been studied in the area of statistical databases. A trend in this field is to *quantify* privacy, e.g. k-anonymity [22], ℓ-diversity [17], and differential privacy [10]. Similar notions are rare in the domain of formal protocol analysis, though examples exist in the field of voting [12, 13]. The advantage of quantifying privacy is that this can capture privacy loss and thus be used to analyse the *minimal information disclosure* inherent in a system.

4.1 Intuition

Like unlinkability, using minimal information disclosure to model privacy is inspired by privacy in voting. In most countries, elections are divided into districts, and each district has one or more polling stations where voters assigned to that polling station can cast their vote. It is common to announce the result of each polling station individually, to ensure a measure of verifiability. Remark that this reveals information about how a voter could have voted: suppose a candidate received zero votes in polling station A and some votes of polling station B, then the privacy of voters that were to vote in polling station A is less than the voters of polling station B.

A traditional indistinguishability approach would find all voters in A distinguishable from voters in B, but find that (barring further information) voters within one polling station are indistinguishable. However, how much privacy is lost remains unclear. This is what minimal information disclosure aims to capture: how much privacy does a subject have.

4.2 Machinery

Minimal information disclosure models the system under analysis as a labelled transition system induced by a normal process algebra. The underlying notion on which privacy is build is *trace* *indistinguishability* of two traces. Two traces t, t' are considered indistinguishable if there exists a *reinterpretation* ρ under which the intruder-observed part of the traces are equal, formally:

$$t \sim t' \equiv \exists\ \rho\colon \mathit{observation}(t) = \rho(\mathit{observation}(t')) \wedge K_I^t = \rho(K_I^{t'}).$$

Reinterpretations are an intruder's "guess" of what happened in the trace — assigning values to variables such as keys, nonces, encrypted terms, etc. Naturally, these must be consistent, that is, everywhere where two terms t, t' can be distinguished, their reinterpretations $\rho(t), \rho(t')$ must also be distinguishable.

Even if two reinterpretations are consistent, they may be distinguishable — e.g. when they lead to different outcomes. For example, in a voting system every voter chooses a candidate. We can model this as a mapping $\gamma\colon \mathcal{V} \to \mathcal{C}$, where $\mathcal{V}$ represents the set of voters and $\mathcal{C}$ the set of candidates. The specific mapping used determines the result. So, two mappings γ_a, γ_b are indistinguishable for the intruder, notation $\gamma_a \simeq \gamma_b$, if and only if

$$\begin{array}{c}\forall\ t \in \mathit{Traces}(\mathit{Votsys}(\gamma_a))\colon \exists\ t' \in \mathit{Traces}(\mathit{Votsys}(\gamma_b))\colon t \sim t'\ \wedge \\ \forall\ t \in \mathit{Traces}(\mathit{Votsys}(\gamma_b))\colon \exists\ t' \in \mathit{Traces}(\mathit{Votsys}(\gamma_a))\colon t \sim t'.\end{array}$$

To determine how much privacy is left to a user of the system, we construct a *choice group*. The choice group consists of all choices for user-actions that are indistinguishable from other choices for the intruder. For the example of a voting system *Votsys*, the choice group of a given mapping γ_a for a given voter v is the set of all candidates $\gamma_b(v)$ a voter could have chosen, which the intruder could not distinguish from the actual mapping γ_a. This is expressed as

$$\mathit{cg}_v(\mathit{VotSys}, \gamma_a) = \{\gamma_b(v)\ \mid \gamma_b \simeq \gamma_a\}.$$

To determine the *minimal information disclosure* for a user, we determine her choice group. The size of the choice group quantifies the privacy of a given user – which we can compare to the number of all possible choices (the size of the set of candidates). By extending the intrudel model to include collaboration with the intruder, we can compare privacy of coerced and non-coerced users. We consider whole-sale (a priori and a posteriori) knowledge sharing and collaborating on information sent over private communication channels. The latter can involve merely sharing what the user send (via an action $\mathit{is}(\mathit{term})$) or even letting the intruder construct such terms (via action sequence $\mathit{is}(\mathit{vars}(\mathit{term})) \cdot \mathit{ir}(\mathit{newterm})$).

Collaborating users are modelled via a process transformation (introducing the above actions) on regular user processes. Privacy of such collaborating users can now be compared to non-collaborating users by determining, for each, the choice group and comparing their sizes.

4.3 Findings

The above machinery provides a quantified model of privacy. Consider, for example, the FOO protocol. Informally, a voter must have *some* minimum amount of privacy if an attacker can interpret his vote in at least two ways. In terms of the framework: the voter's choice group must contain at least two elements. Of course, this is not possible if the result is unanimous or if there's only one voter. Hence, the theorem requires the existence of at least two voters who vote differently:

Theorem 3 (privacy of FOO92)**.** *Suppose* $|\mathcal{V}| > 1$ *and* $|\mathcal{C}| > 1$*. Then, for all* γ_a *for which* $\exists\ va, vb \in \mathcal{V}\colon \gamma_a(va) \neq \gamma_a(vb)$*, we have* $\forall\ v \in \mathcal{V}\colon |\mathit{cg}_v(\mathit{FOO}, \gamma_1)| > 1$.

5 Analysis

Privacy as Indistinguishability Indistinguishability conveniently supports privacy-type properties stronger than mere voter privacy: receipt-freeness and coercion-resistance can also be tackled by this approach, albeit the degree of supported mechanisation may vary.

Generally speaking, a major drawback of the indistinguishability approach using the applied pi calculus is that voter privacy proofs cannot be entirely mechanised, namely the necessary proofs are only partially supported in the ProVerif protocol verification software. ProVerif can generally check static equivalences, but not full observational equivalence. Depending on the cryptographic primitives used in an e-voting protocol, the equational theories necessary to include them in the framework may be beyond the scope of ProVerif mechanisation, to the point that even static equivalence may not always be checked by the software. The Okamoto protocol was a traditional example in which ProVerif could not prove voter privacy [19]; however, this particular limitation has been overcome only recently [7].

Also, the threat model hard-coded in ProVerif overestimates the standard Dolev-Yao threat model, hence it may signal attacks that the human analyst realises not to be so. Therefore, despite the tool itself is automatic, the amount of human intervention required is not negligible.

Privacy as Unlinkability In spite of subjectivity, we believe that one of the major strengths of this approach is that it is intuitive. It also resembles the investigator's approach on a crime scene, of finding a clue such as the size of a footprint and then reducing the set of suspects from everyone to only those wearing that size. And the inductive references modelled by *asynth* represent the additional deductions that the investigator may build.

A weakness could be that *aanalz* does not extract the sender's identity over anonymous channels. One may argue that the attacker derives the sender's identity from the out-of-band, for example by observing who enters a voting site. This criticism would be easy to address by removing the innermost if-then-else of *aanalz* so as to always extract the sender's identity.

It could also be observed that *asynth* only takes a single step of synthesis between associations that intersect. Its premises mention two associations that are drawn from the original set of associations termed *as* and not from the inductive set *asynth as*. On one hand, this may not be considered a serious weakness until there is evidence that protocols vastly appeal to the technique of distributing crucial associations over several protocol messages; on the other hand, porting the current proofs over the generalised definition is yet to be achieved.

Privacy as Minimal Information Disclosure The main strength of minimal information disclosure is that it enables one to reason about privacy in situations where there is more than one class of subjects. Whether this distinction is made by physical means (e.g. geographically, such as with voting precincts), based on attributes (e.g. age) or through yet different means, the existence of different classes need not imply a complete loss of privacy.

Currently, the main drawback of this approach is the specific focus of the current machinery. Minimal information disclosure was originally conceived for the domain of vote privacy, and the underlying mechanism is tailored to that specific domain. Secondly, the approach inherently requires one to know a priori what privacy-affecting information will be available to the attacker. In voting, this is the result — public information. In other areas, it will not always be clear. For example, a few years ago top secret documents were leaked to journalists. It was not clear whether this act was a breach of trust or a breach of security, i.e. whether it was done by someone with appropriate security clearance or by someone without, until Edward Snowden (holding appropriate clearance) revealed his involvement.

5.1 Comparison

The minimal information disclosure approach to privacy is inherently finer-grained than the other two. Definitive boolean claims about privacy may seem intrinsically simplistic if interpreted in a real-world context. However, both indistinguishability and unlinkability enjoy stronger tool support and hence are more easily applicable to additional case studies.

As for tool support, ProVerif offers full automation to indistinguishability while Isabelle is not fully automatic and requires human directions over the proof argument. However, also automatic tools generally require human effort to ensure termination and meaningful outputs, and ProVerif makes no exception when used intensively for a privacy assessment.

The unlinkability approach has other shortcomings [18]. One is that *aanalz* does not extract the identity of the agent who signs a digital signature; thus, if a voting protocol prescribes the voter to sign his vote and submit the outcome, this pair voter-vote would not appear in the set of all possible associations *asynth*(*aanalz Spy evs*), whereas the voter privacy of this protocol would not be met by means of indistinguishability. A similar conclusion would apply to a protocol that allowed an attacker to learn that two voters casted the same vote.

This does not mean that unlinkability is weaker than indistinguishability. The two shortcomings affect the *implementation* of the unlinkability concept outlined above (§3) and not the concept itself. They can easily be resolved by having *aanalz* also extract the signer's identity and by proving an additional theorem that no pair of voters appear in any meaningful association along with a vote. The unlinkability concept was in fact informally given [8, 15] to abstractly mean that the voter could not be linked to his vote from the malicious observation of a trace; it was *then* prototypically implemented via *aanalz*, *asynth* and corresponding theorems, which in fact can be easily extended or amended. We remark that it is vice versa with the indistinguishability concept: it was precisely coined *after* a given formal definition in terms of labelled bisimilarity (Theorem 1, §2).

By contrast, an informal argument can be advanced to conclude that the two concepts are equivalent. It can be outlined as follows. Unlinkability implies indistinguishability because the former tackles general associations between any voter and any votes. Therefore, proving unlinkability implies that the attacker cannot see any of the associations (that a voter cast a specific vote, and another vote cast another vote) which premise the indistinguishability conclusion. So, that conclusion trivially holds (because its premises are falsified). It is also the case that indistinguishability implies unlinkability because "linkability of the vote to the voter states that a run $[\ldots, v, \ldots, \omega(v), \ldots]$ is distinguishable from a run where the vote was cast by another voter $[\ldots, v', \ldots, \omega(v), \ldots]$" [15].

6 Conclusions

This paper reviewed the three main approaches to conduct formal privacy analysis in the context of e-voting. Due to space limitations, a full account on the related work was impossible, hence only those works that were barely sufficient to support the arguments were discussed.

Indistinguishability is unequivocally the most widespread and best developed approach, and its tool support is improving. Unlinkability seems to be an equivalent concept, but its currently available implementation, though promising, is weaker than the implementation of the other approach by means of labelled bisimilarity. Orthogonally to both, minimal information disclosure is a promising and highly detailed approach whose tools support is worthy of development.

References

[1] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *POPL 2001*, pages 104–115. ACM, 2001.

[2] Article 29 Data Protection Working Party. Work programme 2016–2018 (WP235), 2016.

[3] M. Backes, B. Pfitzmann, and M. Waidner. Formal Methods and Cryptography. In *FM 2006*, volume 4085 of *LNCS*, pages 612–616. Springer, 2006.

[4] G. Bella. *Formal Correctness of Security Protocols*. Information Security and Cryptography. Springer, 2007.

[5] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *CSFW-14*, pages 82–96. IEEE Computer Society, 2001.

[6] B. Blanchet. A Computationally Sound Mechanized Prover for Security Protocols. *IEEE Trans. Dependable Sec. Comput.*, 5(4):193–207, 2008.

[7] B. Blanchet and B. Smyth. Automated Reasoning for Equivalences in the Applied Pi Calculus with Barriers. In *CSF 2016*, pages 310–324. IEEE Computer Society, 2016.

[8] D. Butin, D. Gray, and G. Bella. Towards Verifying Voter Privacy through Unlinkability. In *ESSoS 2013*, volume 7781 of *LNCS*, pages 91–106. Springer, 2013.

[9] S. Delaune, S. Kremer, and M. Ryan. Verifying Privacy-Type Properties of Electronic Voting Protocols: A Taster. In *Towards Trustworthy Elections, New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 289–309. Springer, 2010.

[10] C. Dwork. Differential Privacy. In *ICALP 2006*, volume 4052 of *LNCS*, pages 1–12. Springer, 2006.

[11] A. Fujioka, T. Okamoto, and K. Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *AUSCRYPT '92*, volume 718 of *LNCS*, pages 244–251. Springer, 1992.

[12] H. Jonker, S. Mauw, and J. Pang. A formal framework for quantifying voter-controlled privacy. *J. Algorithms*, 64(2-3):89–105, 2009.

[13] H. Jonker and J. Pang. Bulletin Boards in Voting Systems: Modelling and Measuring Privacy. In *ARES 2011*, pages 294–300. IEEE Computer Society, 2011.

[14] S. Kremer and M. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In *ESOP 2005*, volume 3444 of *LNCS*, pages 186–200. Springer, 2005.

[15] L. Langer, H. Jonker, and W. Pieters. Anonymity and Verifiability in Voting: Understanding (Un)Linkability. In *ICICS 2010*, volume 6476 of *LNCS*, pages 296–310. Springer, 2010.

[16] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing Receipt-Freeness in Mixnet-Based Voting Protocols. In *ICISC 2003*, volume 2971 of *LNCS*, pages 245–258. Springer, 2003.

[17] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[18] A. Marcedone. Indistinguishability vs Unlinkability: defining voter privacy in an electronic protocol, 2014. Università di Catania, unpublished.

[19] T. Okamoto. An electronic voting scheme. In *IFIP World Conference on IT Tools*, pages 21–30, 1996.

[20] L. C. Paulson. The Inductive Approach to Verifying Cryptographic Protocols. *Journal of Computer Security*, 6(1-2):85–128, 1998.

[21] S. Schneider and A. Sidiropoulos. CSP and Anonymity. In *ESORICS 96*, volume 1146 of *LNCS*, pages 198–218. Springer, 1996.

[22] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[23] M. Wenzel, L. C. Paulson, and T. Nipkow. The Isabelle Framework. In *TPHOLs 2008*, volume 5170 of *LNCS*, pages 33–38. Springer, 2008.