Specification, Design, and Verification of an Accountability-Aware Surveillance Protocol

Thibaud Antignac
antignac@chalmers.se

Mukelabai Mukelabai
muka@student.chalmers.se

Gerardo Schneider
gerardo@cse.gu.se

Department of Computer Science and Engineering
Chalmers | University of Gothenburg
Gothenburg, Sweden.

ABSTRACT

Though controversial, surveillance activities are more and more performed for security reasons. However, such activities are extremely privacy-intrusive. This is seen as a necessary side-effect to ensure the success of such operations. In this paper, we propose an accountability-aware protocol designed for surveillance purposes. It relies on a strong incentive for a surveillance organisation to register its activity to a data protection authority. We first elicit a list of accountability requirements, we provide an architecture showing the interaction of the different involved parties, and we propose an accountability-aware protocol which is formally specified in the applied pi calculus. We use the ProVerif tool to automatically verify that the protocol respects confidentiality, integrity and authentication properties.

Keywords

formal verification; protocol; privacy; accountability

1. INTRODUCTION

In the face of security threats, surveillance systems are more and more in use under various forms such as airport security controls, CCTV cameras, internet-based forms of surveillance, etc. These surveillance tasks can be carried out by private companies, police services, or intelligence agencies. However, concerns have arisen about the privacy intrusiveness of such systems, as it threatens the civil liberties of the citizens that this surveillance is meant to protect [5].

Among the different principles widely acknowledged to constitute the pillars of privacy lie the notion of *accountability*. The need for surveillance organisations to be accountable is becoming of increasing public interest, so policy debate worldwide is gaining prominence in sectors such as academia, freedom activism and politics [14]. Just to name a few, the EU Article 29 Working Party on data protection declared in 2010 the need for surveillance organisations to adopt an accountability principle [2]. Also, the US President instituted a board to review and give recommendations on the operations of the NSA following the leaks by Edward Snowden in 2013 [5]. Finally, the privacy advocate Senator Faulkner of Australia called in 2014 for “strong and rigorous oversight” over surveillance organisations in order to ensure their “strong and effective accountability” [8].

Accountability can be split in four parts: i) transparency, ii) responsibility, iii) assurance, and iv) remediation [16]. *Transparency* is the cornerstone of accountability as it is necessary to enable the other parts [15]. A system providing accountability ensures individuals and organisations can be held accountable for inappropriate uses of information. Achieving transparency through accountability of surveillance organisations would require a balance between two apparently conflicting goals: meeting security objectives of a surveillance organisation on one hand and guaranteeing the privacy of the citizens concerned on the other hand [5]. Indeed, surveillance operations are often secretively carried out on citizens without their knowledge, the justification being that this is intrinsic to the nature of these operations and that the purpose of surveillance would be endangered if transparency was brought into the system.

With accountability as the championed remedy to privacy protection, the challenge still lies in how to render it acceptable to the surveillance organisation without compromising its main mission. That said, this prevents the application of the so-called *Individual Participation Principle* which guarantees, among other things, that the citizen has the right “to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him” [13]. Under current legislations and techniques, a citizen has very weak guarantees of gaining access to information about surveillance collections and processes of which he is the data subject. This is still the case a long time after the data has been collected, once such a disclosure would no longer defeat the purpose of the collection.

In this paper, we consider the context of a Surveillance Organisation that gathers information about Citizens, which in turn may be used against the very same Citizens in court cases. In this eventuality, the Court issues an order for the Surveillance Organisation to disclose some information which may relate to the Citizen under investigation to serve

SAC'17, April 3-7, 2017, Marrakesh, Morocco
Copyright 2017 ACM 978-1-4503-4486-9/17/04…$15.00
http://dx.doi.org/10.1145/3019612.3019826

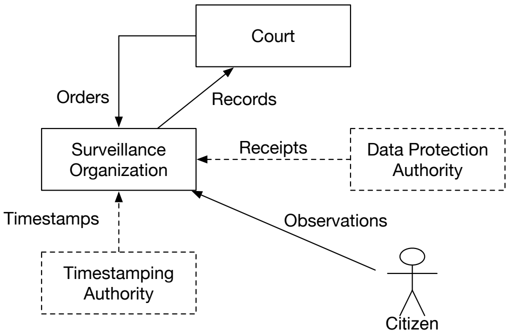

Figure 1: Architecture of AASP (stakeholders and domain).

as evidence. The Surveillance Organisation then replies to this request by disclosing information matching the court order if possible.

The main contribution of this work is the proposal of AASP, an *accountability-aware surveillance protocol*. More concretely:

- An architecture taking into account the context and the requirements of the stakeholders for AASP (Sect. 2);
- The AASP protocol itself (Sect. 3);
- A formal verification that AASP satisfies certain security requirements (with a link to the source code used for this purpose, Sect. 4).

2. AASP ARCHITECTURE

In this section, we introduce the architecture of AASP, specifying the different stakeholders (Sect. 2.1) and the domain model (Sect. 2.2) before expressing the requirements (Sect. 2.3).

2.1 Stakeholder Roles

We introduce five different stakeholder roles: three for the main surveillance goal and two for the accountability goal. Thus, several stakeholders can act under the same role though some roles are exclusive.

The three stakeholders mentioned in the introduction naturally arise for our application as their absence would make surveillance operations infeasible or meaningless. They correspond to the boxes (and actor) made from continuous lines in Fig. 1. These three stakeholders are mutually exclusive:

Surveillance Organisations store surveillance records about Citizens. Each of these surveillance records is called an Observation and may be later requested for by Courts.

Citizens represent natural persons being data subjects of Observations of Surveillance Organisations.

Courts are legal entities which have the sole right of making "public" the Observations. This happens when a Court issues a court Order to Surveillance Organisations for Observations on particular Citizens. Each such order may refer to particular metadata such as personally identifiable information (PII, e.g., biometric codes, social security numbers, and names) or locations for instance.

To bring accountability to a classical surveillance architecture, we propose to add two new stakeholders (they could be merged into one depending on the context but we split them for modularity purpose here), who will act as trusted third parties and appear as dashed boxes in Fig. 1, so they have to be independent of the other three.

Data Protection Authorities satisfy the purpose of bringing transparency in the surveillance operations. This is done by making Surveillance Organisations registering their Observations to such authorities.

Timestamping Authorities have as only purpose to timestamp the Observations sent by Surveillance Organisations.

All these stakeholders will have dedicated roles to meet the requirements of the architecture. Before detailing these roles, we draw the domain model to fix a rigorous terminology of the concepts at hand.

2.2 Domain Model

The only entity mentioned so far are the Observations, which are to be modified and augmented during their lifecycle. We introduce in what follows all the different kind of data in AASP, starting with those in classical surveillance schemes, appearing as continuous arrows in Fig. 1, with the direction of the arrow indicating the data flow):

Observations are identifiable surveillance data handled by Surveillance Organisations. Such Observation have Metadata and may also be linked to Facts. Indeed, Facts exist on their own, without having to be linked to Observations, while Metadata only exist to serve the purpose of constructing Observations.

Orders are queries made by Courts to Surveillance Organisations in order to get Records whose Metadata information match specific properties.

Bringing accountability into general schemes require to add new kinds of data (dashed arrows in Fig. 1):

Timestamps ideally relate to the time at which Observations are made. In practice, they reflect at which time Observations have been timestamped by Timestamping Authorities, which may differ.

Receipts are pieces of data delivered by Data Protection Authorities to certify or acknowledge a matter of fact.

Records are Observations associated with both Timestamps and Receipts. This is the only kind of data that should reach Courts when Orders to do so are made.

The description of the domain model allows to express the requirements as detailed in the following section.

2.3 Requirements

AASP requirements are shown in Table 1. They are split into *functional*, *accountability*, and *security* requirements.

Functional requirements #1 and #2 feature the main functionalities of the system, namely to make Surveillance Organisations collect Observations that are later disclosed to Courts upon Orders. We add requirements #3 and #4 for accountability purpose. Requirement #3 mandates that Surveillance Organisations register their Observations to Data Protection Authorities, while requirement #4 requires that Data Protection Authorities reply to Requests from Citizens about their subjection to surveillance activities by Surveillance Organisations.

We only focus here on the security requirements #3 (#4 has been addressed but not reported here for sake of space). Each requirement is split into four different sub-kinds: *confidentiality*, *integrity*, *authentication*, and *non-repudiation.*

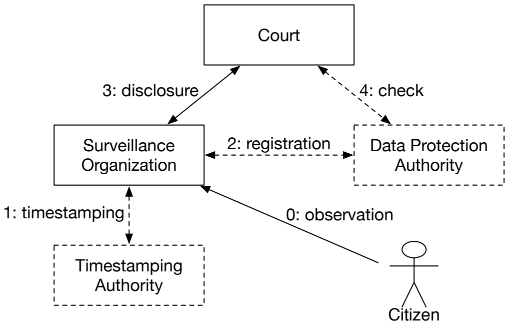

Figure 2: Architecture of AASP (numbers represent the order of steps).

Confidentiality requirement #3-A restricts the access of the different stakeholders roles to Observations such that only Surveillance Organisations can access them. Integrity requirements #3-B forbid Surveillance Organisations to simulate a registration, #3-C to modify Observations after registration, and #3-D to register Observations after they have been requested by Courts. The authentication requirements, #3-E/F, ensure that the different stakeholders actually communicate with the parties they believe they communicate with through securing identification. Finally, non-repudiation requirement #3-G obliges Surveillance Organisations to independently prove they conformed to what was expected.

3. AASP PROTOCOL

We present now the AASP protocol, with focus on the fulfillment of requirements #1, #2, and #3. To improve the quality of the design, we followed the guidelines to build cryptographic protocols given in [1]. In the following, we assume that all the communications between stakeholders are made through public channels (implying that the messages in transit should be protected). This weak assumption gives space for a strong attacker model (as shown in Section 4.2).

The protocol is composed of three main steps: timestamping of Observations (Sect. 3.2), registration of Observations (Sect. 3.3), and disclosure of Observations (Sect. 3.4) as shown in Fig. 2 (the numbers represent the normal order of operations). Surveillance Organisations first send Observations to Timestamping Authorities to get Timestamps associated to these Observations. These Observations along with their Timestamps are then registered to Data Protection Authorities which give back Receipts. All these communications are encrypted to avoid to be eavesdropped by an attacker. All these messages are also signed by the corresponding parties to meet the requirements as will be explained in next section. Once Courts send Orders, Surveillance Organisations sends the corresponding Records as Evidences. Courts can thus verify whether or not (unforgeable) signatures of Timestamping Authorities and Data Protection Authorities appear and detect if Observations have not been registered properly. This provides a strong incentive to Surveillance Organisations to register their Observations as they cannot provide unregistered Observations in Courts without being detected. This allows Data Protection Authorities to reply to requests made by Citizens about their subjection to surveillance activities. All these steps are preceded by a mutual authentication between the stakeholders (Sect. 3.1) (not represented in Fig. 2).

3.1 Authentications

As already mentioned, each session begins with a mutual authentication to securely identify the parties. We chose the classical Needham-Schroeder protocol [10] for this purpose. We do not present it in depth here as it is not part of the contribution, and another authentication scheme could be chosen depending on the architecture and its constraints. Specifically, the Needham-Schroeder protocol relies on a mutual challenge where each agent should prove it has been able to decrypt a nonce (ie., a fresh random number) to the other agent as can be seen in the first box labelled "Authentication" in Fig. 3. The sequence diagram depicted here should be read top-down[1]. The element k_X^{pub} represents the public key of X while k_X^{pr} represents its private key. Nonces generated by X are denoted n_X or nN_x with N a natural number. The identities of the stakeholders are denoted by an abbreviation detailed in the caption. Finally, $\{m\}_{k_X^{\text{pub}}}$ denotes that message m is encrypted with the public key of X, $\text{sig}(m, k_X^{\text{pr}})$ is the signature of message m with the private key of X, and h is a (one-way) hash function. This way

[1] Sequence diagrams only present desired behaviours. The verification presented in Section 4 relies on an attacker model able to perform other (undesired and malicious) behaviours.

Table 1: Functional, accountability, and security requirements.

#	*Kind*	*Sub-kind*	*Requirement*
1	**Functional**		**Surveillance Organisations collect Observations from Citizens**
2	**Functional**		**Surveillance Organisations disclose Observations to Courts upon Orders**
3	**Accountability**		**Surveillance Organisations register Observations to Data Protection Authorities**
3-A	Security	Confidentiality	Only Surveillance Organisations access Observations
3-B	Security	Integrity	Surveillance Organisations cannot falsely pretend to have registered Observations
3-C	Security	Integrity	Surveillance Organisations cannot modify registered Observations
3-D	Security	Integrity	Surveillance Organisations must register Observations before receiving Orders
3-E	Security	Authentication	Surveillance Organisations register Observations to authentic Data Protection Authorities
3-F	Security	Authentication	Data Protection Authorities register Observations from authentic Surveillance Organisations
3-G	Security	Non-repudiation	Surveillance Organisations can prove registration of Observations to Data Protection Authorities
4	**Accountability**		**Data Protection Authorities reply to Requests from Citizens**

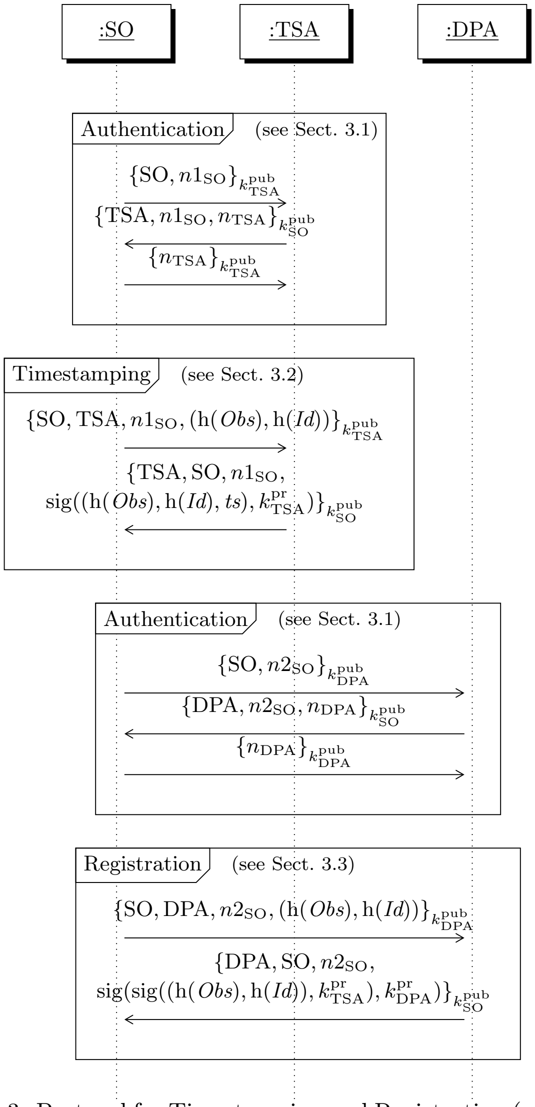
:SO
:TSA
:DPA
Authentication (see Sect. 3.1)
$\{\mathrm{SO}, n1_{\mathrm{SO}}\}_{k^{\mathrm{pub}}_{\mathrm{TSA}}}$
$\{\mathrm{TSA}, n1_{\mathrm{SO}}, n_{\mathrm{TSA}}\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$
$\{n_{\mathrm{TSA}}\}_{k^{\mathrm{pub}}_{\mathrm{TSA}}}$
Timestamping (see Sect. 3.2)
$\{\mathrm{SO}, \mathrm{TSA}, n1_{\mathrm{SO}}, (\mathrm{h}(\mathit{Obs}), \mathrm{h}(\mathit{Id}))\}_{k^{\mathrm{pub}}_{\mathrm{TSA}}}$
$\{\mathrm{TSA}, \mathrm{SO}, n1_{\mathrm{SO}}, \mathrm{sig}((\mathrm{h}(\mathit{Obs}), \mathrm{h}(\mathit{Id}), \mathit{ts}), k^{\mathrm{pr}}_{\mathrm{TSA}})\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$
Authentication (see Sect. 3.1)
$\{\mathrm{SO}, n2_{\mathrm{SO}}\}_{k^{\mathrm{pub}}_{\mathrm{DPA}}}$
$\{\mathrm{DPA}, n2_{\mathrm{SO}}, n_{\mathrm{DPA}}\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$
$\{n_{\mathrm{DPA}}\}_{k^{\mathrm{pub}}_{\mathrm{DPA}}}$
Registration (see Sect. 3.3)
$\{\mathrm{SO}, \mathrm{DPA}, n2_{\mathrm{SO}}, (\mathrm{h}(\mathit{Obs}), \mathrm{h}(\mathit{Id}))\}_{k^{\mathrm{pub}}_{\mathrm{DPA}}}$
$\{\mathrm{DPA}, \mathrm{SO}, n2_{\mathrm{SO}}, \mathrm{sig}(\mathrm{sig}((\mathrm{h}(\mathit{Obs}), \mathrm{h}(\mathit{Id})), k^{\mathrm{pr}}_{\mathrm{TSA}}), k^{\mathrm{pr}}_{\mathrm{DPA}})\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$

Figure 3: Protocol for Timestamping and Registration (with SO a Surveillance Organisation, TSA a Timestamping Authority and DPA a Data Protection Authority).

to describe protocols is adapted from [1] though we use a formalism closer to sequence diagrams than to succession of messages to make them more visually intuitive.

3.2 Timestamping

At this point, we suppose Observations have already been made (the collection of Observations is outside of our scope). The Timestamping Authority role is in charge of providing a timestamp which will be associated with an Observation. The protocol is initiated by Surveillance Organisations as shown in the second top box in Fig. 3.

In the first message, the Surveillance Organisation sends a pair of the hash of a secret observation *Obs* and a hash of the related citizen identity *Id*. The hash of *Id* will later be used by the Data Protection Authority to identify and answer to citizen requests to meet requirement #4 (not modelled here as justified in the previous section). The hash of *Obs* shall be used by the Court to verify disclosed Records from the Surveillance Organisation once an Order has been emitted, as will be showed below. In addition to the pair of the hashed observation and identity, the Surveillance Organisation also sends the nonce $n1_{\mathrm{SO}}$ which has been generated during the authentication scheme. This is for the purpose of ensuring that each session with the Timestamping Authority is unique to avoid replay attacks. The identities of both the Surveillance Organisation and the Timestamping Authority are also added to the message as advised in [1]. The former will allow the Timestamping Authority to check that the message it is dealing with is meant to come from the Surveillance Organisation it previously authenticated with while the latter will allow the Timestamping Authority to check that the message it has received is truly meant for it.

In the second message, the Surveillance Organisation receives Observations timestamped with *ts*. At this point, the Surveillance Organisation should verify the authenticity of the signature coming from the Timestamping Authoritybefore continuing to run the protocol. If this is the case, the Surveillance Organisation can then proceed to the registration as explained in the following.

3.3 Registration

The registration phase is shown in the last box in Fig. 3 First, the Surveillance Organisation sends to the Data Protection Authority the timestamped observation it got from the Timestamping Authority. As was the case for the interaction between the Surveillance Organisation and the Timestamping Authority, the Surveillance Organisation sends these timestamped hashed observations with a nonce and the identities of the stakeholders taking part in the session (which should be the Surveillance Organisation and the Data Protection Authority). This serves the same purpose as before, i.e. to provide guarantees for the session. The Surveillance Organisation then receives a Receipt from the Data Protection Authority in the second message.

Before releasing the Receipt, it is important that the Data Protection Authority checks whether the timestamped observation has been previously signed by the Timestamping Authority. Indeed, one important aspect is the check by the Data Protection Authority that the timestamp is not too old. The registration to the Data Protection Authority has to be made quickly after the timestamp has been emitted by the Timestamping Authority. If this was not the case, it would make it possible for the Surveillance Organisation to register an observation *after* an order has been received from the court, while still associated to a timestamp emitted *before* this order. Another way to follow the same goal would be to send again the record to the Timestamping Authority after it is signed by the Data Protection Authority, to enable the court to compare the two timestamps at a later stage during the protocol described in the following. This choice implies consequences about the agent which has the responsibility to perform this check and the trust relationships between the shareholders.

3.4 Disclosure

Contrary to the two previous cases where the initiator was the Surveillance Organisation, this part of the protocol is initiated by the Court as showed in Fig. 4. The court sends an Order concerning a Citizen with identity *Id* in the first

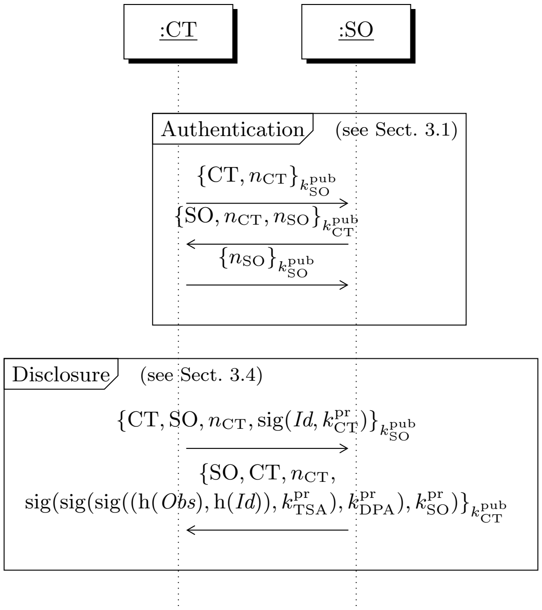
:CT
:SO
Authentication
(see Sect. 3.1)
$\{\mathrm{CT}, n_{\mathrm{CT}}\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$
$\{\mathrm{SO}, n_{\mathrm{CT}}, n_{\mathrm{SO}}\}_{k^{\mathrm{pub}}_{\mathrm{CT}}}$
$\{n_{\mathrm{SO}}\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$
Disclosure
(see Sect. 3.4)
$\{\mathrm{CT}, \mathrm{SO}, n_{\mathrm{CT}}, \mathrm{sig}(\mathit{Id}, k^{\mathrm{pr}}_{\mathrm{CT}})\}_{k^{\mathrm{pub}}_{\mathrm{SO}}}$
$\{\mathrm{SO}, \mathrm{CT}, n_{\mathrm{CT}},$
$\mathrm{sig}(\mathrm{sig}(\mathrm{sig}((\mathrm{h}(\mathit{Obs}), \mathrm{h}(\mathit{Id})), k^{\mathrm{pr}}_{\mathrm{TSA}}), k^{\mathrm{pr}}_{\mathrm{DPA}}), k^{\mathrm{pr}}_{\mathrm{SO}})\}_{k^{\mathrm{pub}}_{\mathrm{CT}}}$

Figure 4: Protocol for Disclosure (with CT a Court and SO a Surveillance Organisation).

message. This message also includes the traditional nonces and identities provided with the encrypted message. The court then reads the reply made by the Surveillance Organisation in the second message and checks that:

1. The Observation in the Record is authentically signed by the Timestamping Authority;
2. The hash of the Observation in the Record is the same as the one signed by the Timestamping Authority;
3. The Observation in the Record is authentically signed by the Data Protection Authority;
4. The hash of the Observation in the Record is the same as the hashed Observation signed by the Data Protection Authority;
5. The hash of the identity *Id* which is the object of the Order corresponds to the one previously given by the Surveillance Organisation to the Data Protection Authority (cf. Sect. 3.3);
6. The hash of the identity *Id* which is the object of the Order corresponds to the identity disclosed in the Record by the Surveillance Organisation;
7. The timestamp *ts* has been emitted before the Order has been sent to the Surveillance Organisation.

The rationale for these verifications will be developed in the next section about the verification of AASP.

4. AASP VERIFICATION

We present here the formal verification of AASP. Some of the requirements will be proven from the protocol itself while some other will rely on an automatic analysis made by the protocol verification tool ProVerif.[2] The language to express queries in ProVerif is easy and we will directly present the queries and their meaning when needed. The requirements verified are confidentiality (Sect. 4.2), integrity (Sect. 4.3), and non-repudiation (Sect. 4.4). As we already mentioned, we do not focus on authentication requirements which are not a contribution specific to this paper (though we performed the verification of these requirements as well). Technically, authentication requirements are handled using the same techniques as those for integrity requirements.

Before describing the verification process we give a brief overview of the theory behind ProVerif.

4.1 ProVerif Modelling

ProVerif [4] is a verifier of properties related to cryptographic protocols. It relies on the *applied pi calculus* which is a modelling language belonging to the family of process algebras [6]. It allows to represent concurrent processes and their interactions through channels and reason about secrecy, authentication, and privacy properties.

The distinctive feature of the applied pi calculus compared to other process algebras is a rich term algebra allowing to define many different cryptographic primitives through equational theories. These cryptographic primitives are assumed to be perfect. For instance, $\mathsf{fst}(\langle x, y\rangle) = x$ models projection of the first element of a pair, and the equation $\mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) = x$ models asymmetric encryption (with a public key x and a private key built thanks to the function pk), $\mathsf{getmsg}(\mathsf{sig}(x, y)) = x$ represents signed message extraction, and finally $\mathsf{checksign}\ (\mathsf{sign}(x, y), \mathsf{pk}(y)) = x$, signature verification. The absence of equation corresponding to the hash function ensures it cannot be destructed (and thus symbolically inverted).

Moreover, it is possible to express conditional statements and inputs and outputs of messages over channels. These terms can be assigned to variables, and a restriction mechanism is provided to model scopes.

In the following, we will use the syntax of the ProVerif verifier which is a typed variant of the applied pi calculus instead of the abstract syntax of the calculus as defined in [18]. For example, the process P, defined as

$$\begin{aligned}
P &= Q|R|S \\
Q &= \mathrm{out}(c, m). \\
R &= \mathrm{in}(c, x).(\text{if } x = n \text{ then } \mathrm{out}(d, o) \text{ else } \mathrm{out}(d, p)) \\
S &= \mathrm{in}(d, y)
\end{aligned}$$

models three processes Q, R, and S running concurrently. Here, a message m is sent through the channel c by Q. This message is received by R and assigned to the variable x which is compared to n. Depending on the result of the test, either the message o or the message p is sent through the channel d, which is get by S (assigned to the variable y). We will explain and motivate special constructs in the following. (See [18] for a complete syntax and semantics.)

4.2 Confidentiality

ProVerif provides a framework to verify confidentiality properties under the Dolev-Yao attacker model [7]. In this model, the attacker has complete control of the network, equipped with the capability to perform the application of any function provided in the equational theory. The attacker can thus build an encrypted message provided it has the corresponding message and key, without any limit (these are

[2] The source code of our model is available online (see http://www.cse.chalmers.se/research/databin/files/aasp.pv).

called constructor functions). However, it can only decrypt those encrypted messages for which it also has the corresponding key (these are called destructor functions).

The two queries listed in Fig. 5 perform a request to know whether or not the attacker can gain access to the variables *Obs* and *Id*. These two queries are verified by ProVerif and guarantee that an external attacker cannot have access to these information, fulfilling partially the requirement #3-A.

```
query attacker(Obs);
      attacker(Id);
```

Figure 5: ProVerif queries to verify confidentiality.

To completely meet this requirement, we verify whether or not the Timestamping Authority or the Data Protection Authority can access these observations. It is not possible to directly express this as a query with ProVerif. Indeed, ProVerif only verifies knowledge gained by an external attacker. It would be possible to output all the knowledge of a party to the public channel, but this would model a collusion between a Dolev-Yao attacker and this party, which is not what we actually want. By an analysis of the protocol, we see that all the sensitive information that the Timestamping Authority and the Data Protection Authority receive from the Surveillance Organisation have been previously hashed. Given that the hash cannot be inverted, (which is modelled in ProVerif by the absence of an equation $\mathsf{unhash}(\mathsf{hash}(x)) = x$ in the equational theory), neither the Timestamping Authority nor the Data Protection Authority can access the observations. As a consequence, the requirement #3-A is met (the court may later receive the observations but it is expected and not part of the accountability requirement #3).

The use of such hashes is useful to provide integrity guarantees while ensuring the confidentiality of the data (it is generally called a *commitment scheme* [9]).

4.3 Integrity

The satisfaction of the integrity requirements are proved by two different means relying either on conditions directly expressed in the processes and dynamically executed by the parties, or on a static analysis relying on the addition of events in the ProVerif models to allow the tool to reason about their occurrence.

Requirement #3-B is ensured by the fact that the Court checks the signature of the Data Protection Authority before releasing the Receipt. Because signatures are supposed to be unforgeable, if the Surveillance Organisation is able to provide an observation which has been signed by the Data Protection Authority, it can only be because the Data Protection Authority actually signed it (provided that the Data Protection Authority did not leak its signature).

Requirement #3-C is met since the Court verifies whether the (hashed) observation provided in the Record matches the (hashed) observation signed by the Data Protection Authority. This relies on the second pre-image resistance property of hashes which made it hard to find a different element having the same hash as the current element.

Finally, requirement #3-D is verified by using *correspondence* between events. The order of these events makes it possible to verify that the observations have been registered before the court order is emitted. This is expressed thanks to the query shown in Fig. 6. This query relies on (ghost) events which have been added to the protocol in order to witness behaviours without modifying the execution. The `==>` in this query can be understood as an implication on the occurrence of events. Thus, `ev1 ==> ev2` means that each time `ev1` appears, `ev2` also has to have appeared. We added more events in Fig. 6 to verify a stronger property including other events. The property satisfying the requirement #3-D is directly implied by this (stronger) property.

Authentication is also verified by such correspondence properties between events. Indeed, a mutual authentication is successful if, after execution of the authentication protocol, each party believes it communicated with the other only if this is true. This can thus also be modelled through correspondence assertions similar to the one presented in Fig. 6. All the authentication requirements (#3-E/F) have been verified in the protocol.

4.4 Non-Repudiation

The non-repudiation requirement #3-G ensures that the Data Protection Authority cannot deny having registered observations from the Surveillance Organisation. This is done by the Data Protection Authority communicating to the Surveillance Organisation a Receipt signed by the signature key of the Data Protection Authority. The verification of this property is not made by ProVerif. It relies on the fact that the protocol guarantees that such signed messages are sent by construction for successful executions.

All the requirements from Table 1 have been verified either by performing ProVerif queries or analysing the behaviour of the parties (conditional executions, non-disclosure of secret keys, and disclosure of unforgeable messages).

5. RELATED WORK

In cloud computing and other areas such as e-commerce services for instance, a number of approaches for privacy have been proposed such as [19] for privacy preservation in images from video-surveillance. However, all these solutions mainly focus on information hiding by means of access control mechanisms and encryption techniques [12].

The definition of surveillance architectures have already been explored in the literature. The SALT framework has been introduced as a multidisciplinary approach to preserve privacy in video surveillance systems and "serves as a decision support to assist system designers and other stakeholders in coping with complex privacy requirements in a systematic and methodological way" [11]. It "provides reusable, generic, and synthetic guidelines, reference information, and criteria to be used or modified by experts and other stakeholders on privacy by design and accountability by design" [11]. The approach comprises two steps: (i) Guiding surveillance system owners through a process for legal, socio-contextual, and ethical impact assessments of the envisioned system, including the impact on individuals' privacy, and (ii) Referring the designers to socio-contextual, ethical, and logical considerations during the design phase to reduce the impact of the system on individuals' privacy. To achieve the latter,

```
query x:bitstring,y:pkey; inj-event(rcvOrdersRecords(x,y))
          ==> (inj-event(rcvCourtOrder(x,y)) ==> (inj-event(rcvReceiptFromDPA(x,y))
          ==> (inj-event(rcvCommitment(x,y)) ==> (inj-event(rcvTSFromTSA(x,y))
          ==> inj-event(rcvTSRequestFromSO(x,y)))))).
```

Figure 6: Query to verify integrity.

accountability features and state of the art privacy preserving technologies are provided to the designers. Unlike our work with AASP, to the best of our knowledge the SALT framework has not been formalised.

Such accountability techniques are more and more praised to address privacy issues [17]. Though they do not solve all concerns, they can provide guarantees about how personal data is processed. Some work such as [3] model how policies can be expressed and enforced in cloud services. Our contribution lies at a higher level by proposing an architecture enforcing accountability by design.

6. CONCLUSION

We have presented AASP, an accountability-aware surveillance protocol, which has been formally verified using the ProVerif verifier. The protocol relies on multiple commitment schemes allowing to keep the data confidential while ensuring integrity properties at the same time. The incentive to register the activity is based on a legal obligation for Surveillance Organisations to do this. This solution does not technically guarantee that Surveillance Organisations will register all their observations to Data Protection Authorities. We chose this solution, which is weaker than would be the use of trusted components, to make it acceptable by Surveillance Organisations while at the same time improving the current situation where Surveillance Organisations are not held accountable. It could be argued that to be used in practice, such a technical solution requires a change in the legal framework, thus making it unlikely to be deployed. This is true and this is why we propose this solution before any legal framework relies on it. Indeed, we claim that showing that such a protocol is possible is a prerequisite for the legal framework to evolve. Consequently, we hope the existence of such a (formal) solution as proposed in this paper, proves useful to advocacy lawyers and civil rights activists in their fight for better surveillance practices.

We are currently working on a correct-by-construction implementation derived from the model to give a practical framework. We hope this will reduce even more the gap between accountability and surveillance activities.

Acknowledgements

This research has been supported by the Swedish funding agency SSF under the grant *DataBIn: Data Driven Secure Business Intelligence*.

7. REFERENCES

[1] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Trans. Softw. Eng.*, 22(1):6–15, Jan. 1996.

[2] Article 29 Data Protection Working Party. Opinion 3/2010 on the principle of accountability, 2010.

[3] W. Benghabrit, H. Grall, J.-C. Royer, M. Sellami, M. Azraoui, K. Elkhiyaoui, M. Önen, A. S. De Oliveira, and K. Bernsmed. *From Regulatory Obligations to Enforceable Accountability Policies in the Cloud*, pages 134–150. Springer, Cham, 2015.

[4] B. Blanchet. *Automatic Verification of Security Protocols in the Symbolic Model: The Verifier ProVerif*, pages 54–87. Springer International Publishing, Cham, 2014.

[5] R. A. Clarke, M. J. Morell, G. R. Stone, C. R. Sunstein, and P. P. Swire. Protecting citizens and their privacy, 12 2013.

[6] V. Cortier and S. Kremer. Formal models and techniques for analyzing security protocols: A tutorial. *Found. and Trends in Prog. Lang.*, 1(3):151–267, 2014.

[7] D. Dolev and A. C. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.

[8] J. Faulkner. Surveillance, intelligence and acountability: an australian story, 10 2014.

[9] O. Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[10] G. Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *TACAS'96*, pages 147–166. Springer, 1996.

[11] Z. Ma, D. Butin, F. Jaime, F. Coudert, A. Kung, C. Gayrel, A. Mana, C. Jouvray, N. Trussart, N. Grandjean, et al. Towards a multidisciplinary framework to include privacy in the design of video surveillance systems. In *APF'14*, pages 101–116. Springer, 2014.

[12] T. J. Neela and N. Saravanan. Privacy preserving approaches in cloud: a survey. *Indian Journal of Science and Technology*, 6(5):4531–4535, 2013.

[13] OECD. Guidelines governing the protection of privacy and trans border flows of personal data, 2013.

[14] OECD. Privacy principles, 2013.

[15] F. A. Pasquale. Beyond innovation and competition: The need for qualified transparency in internet intermediaries. *Available at SSRN 1686043*, 2010.

[16] S. Pearson. Toward accountability in the cloud. *IEEE Internet Computing*, 15(4):64, 2011.

[17] S. Pearson and A. Charlesworth. *Accountability as a Way Forward for Privacy Protection in the Cloud*, pages 131–144. Springer, Berlin, Heidelberg, 2009.

[18] M. D. Ryan and B. Smyth. Applied pi calculus. In V. Cortier and S. Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, chapter 6. IOS Press, 2011.

[19] W. Zhang, S. Cheung, and M. Chen. Hiding privacy information in video surveillance system. In *ICIP'05*, volume 3, pages 868–871. IEEE, 2005.