

Formal Accountability for Biometric Surveillance: A Case Study

Vinh-Thong Ta¹, Denis Butin², and Daniel Le Métayer³

¹ University of Central Lancashire, UK
vtta@uclan.ac.uk

² TU Darmstadt, Germany
dbutin@cdc.informatik.tu-darmstadt.de

³ Inria, Université de Lyon, France
daniel.le-metayer@inria.fr

Abstract. Surveillance, especially using biometric systems, threatens the privacy of individuals. Accountability is an established approach to supporting privacy in general, but it must follow a rigorous process and involve close scrutiny of actual data handling practice to be effective. In this paper, we consider a specific, real-world biometric surveillance system, based on camcorders and bodyprint identification. We show how formalisation can be used to achieve the required level of rigour and exemplify how our formal approach to accountability — in the sense of verifiable compliance with personal data handling policies — supports the privacy of individuals monitored by the system. The formal accountability framework is general enough to be reusable in other settings.

1 Surveillance, Biometrics and Accountability

Surveillance systems using biometrics enjoy growing use, in particular for identification purposes [10, 16, 18]. A powerful feature is the possibility of identifying agents based on automatically detectable visual cues. Some features (e.g. height, hair, clothing) can be acquired without subject cooperation [9]. Even when the gathered information is insufficient to uniquely identify individuals straight away, the capture and processing of their images raises major privacy concern. As put by Campisi, *biometrics are associated with surveillance not simply for legitimate reasons (...) but also with disproportionate, imprecise and invisible use* [7]. Scope creep is therefore a worry. Advanced processing features and the possibility of communication between data controllers (DC) and third parties reinforce these concerns. Mitigating measures are required to keep potential abuse in check.

An established approach to sustain privacy is a focus on *accountability* of DC, in the sense described by the Article 29 Working Group [3]. Accountability is then defined as the duty for DC to not only put in place measures guaranteeing the privacy of data subjects (DS), but also for these measures to be verifiable. Ideally, this verification should be carried out by independent third parties or by agents acting on the behalf of DS (or, wherever practicable, by the DS themselves). This focus on transparency empowers DS and increases pressure on DC to deploy strong

privacy-sustaining measures, as opposed to mere declarations of intention. Three types of accountability are distinguished by Colin Bennett [8]: accountability of policy, of procedures and of practice. The strongest variant, on which we focus here, is accountability of practice which holds that DC ought to demonstrate that their actual data handling complies with their obligations. To be effective, accountability of practice should be based on verifiable, technical information about personal data processing, for instance in the form of auditable event logs [5]. For logs to be easily mappable to privacy policies, a correspondence between low-level system events and high-level data processing can be produced [6].

Our work is inspired by the European project PARIS (PrivAcY pReserving Infrastructure for Surveillance) [1]. Privacy-preserving surveillance infrastructures require accountability models to enforce rigour, clarify process definition and avoid ambiguities. Our motivation is the specification of truly protective accountability measures. Indeed, to be more than mere smoke screens used by data controllers to avoid stronger regulations, accountability must provide concrete evidence about personal data processing and make it possible to have compliance checked by third party auditors.

Related work

Most of the relevant existing work focuses on the security modelling and verification of biometric systems. Lloyd applied Unified Modeling Language (UML) and Java Modeling Language (JML) approaches to the development and security analysis of a biometric authentication system [12]. Salaiwarakul proposed a security verification method for biometric authentication protocols based on the ProVerif protocol analyser [14]. Kanak proposed a formal framework called Biometric Privacy-Security-Trust Model (BioPSTM) to describe the trade-off between privacy and security and their relationship with trust in biometric authentication systems [11].

Formal approaches for reasoning about accountability and privacy system properties are rarely investigated. Accountability has been mentioned in the context of biometrics before, but with a focus on the accountability of system users. For instance, Prabhakar considered the scenario of fingerprint-based information system access control, yielding accountability for system transactions while preserving user anonymity (no names are linked to the fingerprints) [13].

We previously introduced a formal approach to accountability for privacy, independently of the context of biometric surveillance [6]. Our focus there was the correctness of links between system events and operations on categories of personal data in a generic setting. In particular, a generic formal privacy policy language was proposed that defines, for each type of personal data, authorised purposes, deletion delays, request completion delays, admissible contexts and data forwarding policies. Trace compliance properties were defined with respect to data handling events and elements of these privacy policies. We then formalised correctness properties relating personal data handling events and system events. The genericity of this previous work contrasts with the present case study, which

aims at illustrating its application (with some modifications) to a concrete surveillance scenario.

Contributions

We investigate the applicability of a formal approach to accountability to a biometrics surveillance system. The case study under consideration analyses a real-world system deployment by the PARIS project consortium member Visual Tools [15]. The formal approach applied here is based on [6], where the technical aspects of the framework are described in more detail. Modifications to the policy language required for this case study are proposed. To the best of our knowledge, this paper is the first work which examines this specific setting in a real-world system. In contrast to previous work [13], we examine accountability from the system owner’s (DC’s) perspective⁴.

We aim to demonstrate the practical application of a formal framework for accountability to the example of an actual bodyprint-based surveillance system. The case study is performed in sufficient detail to provide a hands-on illustration of accountability of practice for a realistic scenario. The underlying formal framework remains general enough to be applied to different use cases.

Outline

The case study, the involved entities and categories of personal data are presented in Sect. 2. A privacy policy language to model accountability properties is defined next (Sect. 3). Abstract personal data handling events are then introduced to relate processing to privacy policy constraints (Sect. 4) and the compliance of a sequence of events is studied in Sect. 5. We conclude with a discussion (Sect. 6).

2 A Real-World Biometric Surveillance System

A PARIS consortium member has deployed a biometric system to protect equipment stored in their headquarters located in Madrid⁵. This biometric system is based on video analysis to detect unauthorised access to the office at non-working hours, when only the employees are allowed to be present. The system monitors the main transit areas of the office with camcorders, providing depth and spatial information that is analysed to detect individuals accessing the office. To this end, the camcorders are depth cameras with a video processing unit (VPU).

The biometric surveillance system is composed of two main phases: *enrolment* and *matching*. During enrolment, a group of employees are recorded and registered as authorised. Their presence in the restricted areas is permitted. Re-identification of authorised individuals as well as detection and report of unauthorised individuals takes place during matching.

⁴ That is, the system owner must provide accounts to recorded individuals with respect to the handling of their personal data.

⁵ Details about this use case can be found in a project deliverable [15].

The proposed biometric system uses *bodyprints* [2] for re-identification. A bodyprint is a vector of physical characteristics, such as the height and width of a person and clothing colours, which are sufficiently distinctive to make it possible to identify and differentiate individuals, even with similar clothes. Authorised individuals wear uniforms with a particular colour, which facilitates their identification.

Bodyprint extraction

Bodyprint extraction is a two-step process. First, a person is detected by the camcorders, and his/her movement is tracked by different video frames. Second, the bodyprints of the person are created, based on the tracked frames. Bodyprints are not linked with any other personal data. Moreover, it is hard to reconstruct full images from bodyprints.

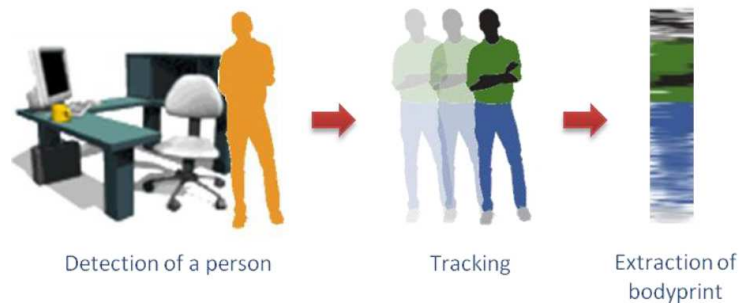


Fig. 1. Bodyprint extraction process (taken from [15])

Enrolment

During enrolment, bodyprints of authorised individuals are extracted and stored in the system. The process of enrolment is performed in three steps:

1. A camcorder records a video of an authorised person, and as a result, a video sequence containing images (frames) of the person is obtained;
2. Bodyprints are extracted from the video frames, and a specific user interface facilitates the selection of the most adequate ones for matching;
3. The selected bodyprints are stored in the Authorised People Database (APDB) located in the re-identification server (RIS).

Enrolment is offline, and managed manually by the System Administrator (SA). The SA gives instruction for authorised individuals before recording them, selects adequate bodyprints and stores them in the APDB.

Matching

The purpose of matching is to detect and report unauthorised access to the office at non-working hours. The process is monitored by the System Operator (SO), and consists of the following steps:

1. Each camcorder continuously captures images of the scene and automatically extracts the bodyprints of the recorded individuals;
2. The camcorders periodically send new bodyprints to the RIS;
3. The RIS compares the new bodyprints with the ones stored in the APDB. The results of the comparison are copied to the alert database;
4. The SO checks the stored alerts through a specific user interface of the alert management server. In case of intrusion, the SO is responsible for reporting the incident to the local authorities.

While enrolment is managed manually by the SA, matching is fully automated, involving an intervention of the SO only in the last step. Each piece of video and non-video data in the system is given a unique identifier (ID) for reference and searching purposes (§5.1.2 in [15]). In terms of personal data minimisation, no ID or basic information (civil identity, name, birthday, etc.) related to the DS is stored in the system; only the IDs of the videos and bodyprints.

In this system, the data retention period follows the Instruction 1/2006 of the Spanish Data Protection Agency [17]. On this basis, images and bodyprints can at most be stored for one month. For enrolment the videos are stored in a VPU until a corresponding set of (temporary) bodyprints has been extracted. This period in any case will not exceed one month. In the matching phase, the videos are deleted right after the bodyprints have been extracted, which is carried out automatically within a few minutes⁶. In each set of extracted bodyprints, one long-term bodyprint is selected and stored in the APDB until the system is retired. The remaining temporary bodyprints are automatically deleted within a few minutes after the long-term bodyprint has been selected. Finally, the bodyprints subject to an alarm during the matching phase are kept in the alert database until the results of the recognition are verified by the SO, with a maximum limit of one day⁷. The other bodyprints are deleted within a few minutes after they have been checked against the APDB.

Eventually, we review the roles and privileges of users in the system. The role of the SAs is to manage the whole system, hence they are granted all privileges, namely, having access to all information stored in the system. For instance, a SA can manage (add, edit, delete data) the APDB, has access to the RIS, and also access to the VPU. The SO is responsible for managing the intrusion alarms; hence, he has access to the RIS and the alert database inside it, and to the APDB as well. The SO receives alarms on a computer or mobile device, and then

⁶ Although in §5.1.2 of [15] (alarm management part), *optionally*, the system could also directly send an image of the intruder with the alarm. We consider only the basic setting to simplify the formalism.

⁷ The worst case delay authorised by Spanish law remains one month. One day is a much more realistic time frame in this case, since the verification is time-sensitive.

logs into the RIS to check the bodyprints subject to the alarm. In case of false alarms these bodyprints are deleted from the RIS alert database by the SO. We assume that accountability auditors are independent third parties such as Data Protection Authority officers. They can require access to certain information stored in the system to verify an intrusion or the compliance of the system with the regulations. To this end, they can be granted reading access to the bodyprints in the alert database, or to the bodyprints in the APDB.

3 Defining Privacy Policies

Based on the language introduced in [6], we propose a modified privacy policy language to model accountability properties for this case study. Due to space limitations, we only consider the videos and bodyprints of individuals during enrolment and matching. The remaining data, such as the ID of the SA and SO, can be modelled similarly. We aim to show that a formal approach to system design with accountability in mind is feasible and to illustrate the resulting benefits, such as increased clarity.

Privacy policies

Definition 1 (Privacy Policy). *Privacy policies are defined as tuples:*

$$\mathcal{P} = Purposes \times Time \times Time \times Context \times AccPol \times AccPol \times AccPol$$

We distinguish privacy policies for each type of personal data and phase. Specifically, \mathcal{P}_E^{vid} is the policy defined for the video frames of the authorised individuals captured during enrolment. \mathcal{P}_M^{vid} relates to the video frames of the DS recorded during matching. \mathcal{P}_E^{tmp} is defined for the set of temporary bodyprints calculated from the video frames of a given employee recorded during enrolment. In the second step of enrolment, first a set of temporary bodyprints is extracted from the corresponding video, then the most adequate one is stored in the APDB, while the rest will be deleted. \mathcal{P}_M^{tmp} is similar to \mathcal{P}_E^{tmp} but deals with the temporary bodyprints during matching. \mathcal{P}_E^{apdb} is the policy defined on the selected bodyprints stored in the APBD during enrolment. \mathcal{P}_M^{alert} defines the policy for bodyprints stored in the alert database located in the RIS. In case of alarm, the SO will access this database to verify the bodyprint of the intruder.

Specifically, for $\pi_{ev} \in \mathcal{P}_E^{vid}$, $\pi_{mv} \in \mathcal{P}_M^{vid}$, $\pi_{et} \in \mathcal{P}_E^{tmp}$, $\pi_{mt} \in \mathcal{P}_M^{tmp}$, $\pi_{ea} \in \mathcal{P}_E^{apdb}$, $\pi_{alert} \in \mathcal{P}_M^{alert}$, let

$$\pi_* = (ap, d, gd, cx, acc^{sa}, acc^{so}, acc^{au})$$

where $*$ $\in \{ev, mv, et, mt, ea, alert\}$.

Particularly, $ap \in Purposes$ is the set of authorised purposes for data use. The retention delay d is explained in the next subsection. gd is a global (worst-case) delay after which all personal data must be deleted. Unlike the retention delays specified below, global deletion delay is defined to prevent data being kept longer

in the system than needed under any circumstances. $cx \in Context$ is the set of contexts in which the data can be used. $Context$ is the set of constants, for instance, time or location. Finally, acc^{sa} , acc^{so} and acc^{au} specify the access policies for the SA, the SO and auditor, respectively.

Possible values for an access policy are \downarrow_{auth} , meaning that access to the data is allowed after a successful authentication⁸, and \uparrow , denoting that access to the data is forbidden.

Retention delays

The retention delay d has a different meaning depending on the type of privacy policy under consideration:

- For π_{ev} : the delay for the DC to delete the video frames stored in a camcorder after a suitable bodyprint has been extracted and stored in the APDB.
- For π_{mv} : the delay for the DC to delete the video frames of the DS during matching after a bodyprint has been extracted from them for the matching purpose.
- For π_{et} : the delay for the DC to delete the temporary bodyprints during enrolment after the selected (adequate) bodyprint has been added to the APDB database by a SA.
- For π_{mt} : the delay after which the DC must delete the extracted bodyprint during matching, after the comparison of this bodyprint with the stored bodyprints (in the APDB) has been performed.
- For π_{ea} : the delay after which the long-term bodyprint stored in the APDB must be deleted by the DC, after the corresponding DS was disenrolled or the system has been retired.
- For π_{alert} : the deletion delay for the temporary bodyprints stored in the alert database, launched after all the alerts have been examined by the SO.

Concrete privacy policy parameters

Three examples of concrete policies for this case study can be found in Fig. 2.

$$\pi_{ev} = (\{\text{"Enrol"}, \text{"Extract"}\}, 1 \text{ min}, 1 \text{ month}, \{\text{DC Building}\}, \downarrow_{auth}, \uparrow, \downarrow_{auth})$$

$$\pi_{mv} = (\{\text{"Match"}, \text{"Extract"}\}, 1 \text{ min}, 1 \text{ month}, \{\text{DC Building}, 21:00/07:00\}, \uparrow, \uparrow, \uparrow)$$

$$\pi_{et} = (\{\text{"Enrol"}, \text{"Choose"}\}, 1 \text{ min}, 1 \text{ month}, \{\text{DC Building}\}, \downarrow_{auth}, \uparrow, \downarrow_{auth})$$

Fig. 2. Some of the concrete privacy policies used by the system

⁸ In case the authentication is performed by the local authorities, it may involve manual aspects.

We now discuss the parameters for π_{ev} , π_{mv} and π_{et} in Fig. 2. As specified in the use case description [15], the deletion delay for videos and bodyprints follows the Instruction 1/2006 of the Spanish Data Protection Agency, and should be maximum one month. Hence, we set the global deletion delay to 1 month, and for demonstration purposes, the retention delays are set to 1 minute for all privacy policies. The videos are used in the DC building and the matching phase at non-working hours (the period between 9 PM and 7 AM).

In π_{ev} the purposes of the enrolled video can be *enrolment* and *bodyprint extraction* (denoted respectively by “*Enrol*” and “*Extract*”), and the procedure takes place in the building of the data controller. Finally, only the SA and the auditor (after successful authentication) are allowed to access the enrolled videos. By contrast, the purposes of the video in π_{mv} are *matching* and *bodyprint extraction*, which have to be done in the DC building between 9 PM and 7 AM. This video is automatically deleted within a short time, hence, no access possibility is available. Eventually, in π_{et} the purposes of the extracted bodyprints can be either *enrolment* or the *choice* of an adequate bodyprint. The SA and the auditor have access rights to the bodyprints. In practice, the meaning of constants such as *Enrol*, *Extract*, *Match*, *Choose* and *DC Building* has to be defined precisely (in natural language) in documents that must be available to the auditors. Indeed, as discussed in Section 6, accountability audits cannot be entirely mechanized and the application of log analysis tools should be complemented by manual verifications, in particular with respect to notions such as purpose and context which may be subject to interpretation or may require further information.

4 Reasoning about Personal Data Handling Events

To reason about personal data handling events with respect to privacy policies, abstract events will be defined first. Abstract states are specified later to express the combined effect of sequences of abstract events. All building blocks to define trace compliance properties (§5) will then be presented. In our context, the ultimate reason for defining trace compliance properties is to define the accountability requirements of the surveillance system.

Abstract Events

To reason about accountability compliance properties, we define the abstract events corresponding to the case study. Each abstract event captures a specific action, or a high-level event occurring during system execution. These events abstract away from system internals such as writing and reading from memory addresses, and are specified based on the format of privacy policies. The key requirement for the set of events is its completeness: it should include all operations that can have an impact on the compliance of the system with respect to any privacy policy. We assume that each recorded video is given a unique identifier *idv* from the set of video identifiers *IDV*, and that each bodyprint extracted from this video is given a unique ID related to the video-ID.

Abstract events (Fig. 3) are tuples starting with an event name, followed by a timestamp capturing the time of the event, parameters of the event, and the privacy policy corresponding to the personal data created by the event (if any⁹). Note that unlike the other parameters, the policies in the events are constants. All parameters in the following list are mandatory.

- E_1 : (*RecordEnrol*, t , cam , enr -video-type, $video$, $idev$, π_{ev})
- E_2 : (*RecordMatch*, t , cam , mat -video-type, $video$, $idov$, π_{mv})
- E_3 : (*ExtrEnrol*, t , $idev$, tmp -bd-set-type, tmp -bd-set, π_{et})
- E_4 : (*ExtrMatch*, t , $idov$, tmp -bd-type, tmp -bd, π_{mt})
- E_5 : (*ChooseEnrol*, t , $idev$, bd -type, bd , π_{ea})
- E_6 : (*AlertMatch*, t , $idov$, $alert$ -bd-type, $alert$ -bd, π_{alert})
- E_7 : (*CompEndMatch*, t , $idov$, tmp -bd-type, tmp -bd)
- E_8 : (*Use*, t , idv , θ , v , $purposes$, $reasons$)
- E_9 : (*Delete*, t , idv , θ , v)
- E_{10} : (*Authenticate*, t , or , idv , θ , v)
- E_{11} : (*AccessReq*, t , or , idv , θ , v)
- E_{12} : (*Access*, t , or , idv , θ , v)
- E_{13} : (*SOAlertVerEnd*, t , $alert$ -bd-type, $alert$ -bd)
- E_{14} : (*SysRetired*, t)

Fig. 3. List of abstract events

Events E_1 – E_2 capture the moment when the camera cam (in the DC’s building) records the video $video$ of type enr -video-type (respectively mat -video-type) with a policy π_{ev} (respectively π_{mv}) for enrolment (respectively matching). The recorded video is given a unique ID $idev$ ($idov$), where the tags ev and ov refer to the enrolment and matching phases, respectively. Similarly, corresponding events E_3 – E_4 for bodyprint extraction during enrolment and matching exist. During enrolment and matching, the set of bodyprints tmp -bd-set (respectively tmp -bd) of type tmp -bd-set-type (respectively tmp -bd-type) is extracted from the video with the IDs $idev$ and $idov$, respectively.

⁹ In this case study, no event leads to the creation of several pieces of personal data. However, tuples could easily be extended to include one policy per created data.

E_5 expresses that during enrolment, the bodyprint bd corresponding to the video identified by $idev$ is selected and stored in the APDB for matching purposes. E_6 occurs during matching, when the bodyprint $alert-bd$ of type $alert-bd-type$ (corresponding to the video $idov$) with the policy π_{alert} has been subject to an alert, and is stored to be verified by the SO. E_7 captures the event when the comparison of the temporary bodyprint $tmp-bd$, extracted from the video $idov$, against the APDB has finished.

$(enr-video-type, video)$ — videos recorded during enrolment
$(mat-video-type, video)$ — videos recorded during matching
$(tmp-bd-set-type, tmp-bd-set)$ — sets of temporary bodyprints extracted from the recorded videos during enrolment
$(tmp-bd-type, tmp-bd)$ — bodyprints extracted from the recorded videos during matching
$(bd-type, bd)$ — (long-term) bodyprints selected from the temporary sets
$(alert-bd-type, alert-bd)$ — bodyprints subject to the alerts, extracted during matching

Fig. 4. Data types and values

We define pairs of data types and values (θ, v) , on which events are defined, in Fig. 4. From now on, let ∇ be the set of these pairs. E_8 represents the events for the data use and $(\theta, v) \in \nabla$. This event defines the use of the data (θ, v) with the ID idv . In our case, *purposes* is the set {"Enrol", "Match", "Extract", "Choose", "Store", "Verification"}, while *reasons* is {"Alert", "AccessRequest"}.

E_9 is a delete event where $(\theta, v) \in \nabla$. It captures the deletion of the data of value v and type θ at time t during enrolment or matching. E_{10} defines an authentication event performed by an originating agent or at time t to access the data (θ, v) with the ID idv .

E_{11} specifies the access request received by the DC from or in order to access the data v of type θ . In case or is SA , (θ, v) is $(enr-video-type, video)$, $(tmp-bd-set-type, tmp-bd-set)$, or $(bd-type, bd)$, because the SA has access to the VPU and the APDB. Similarly, when or is SO , (θ, v) is the alerted bodyprints $(alert-bd-type, alert-bd)$ stored in the RIS, or the long-term bodyprints $(bd-type, bd)$ in the APDB. If or is *Authority*, then (θ, v) is any data type / value pair that the SA and the SO can access. Note that or can also be an unauthorised agent, in which case, access will be refused.

Event E_{12} is the actual access by or to data (θ, v) , where the parameters are similar to the *AccessReq* case (E_{11}). Finally, in E_{13} the SO has terminated the verification of the alerted bodyprint $alert-bd$, while E_{14} indicates that the surveillance system has retired at time t .

Fig. 5 provides an extract of the data flow graph and the relationships between some events during enrolment and matching.

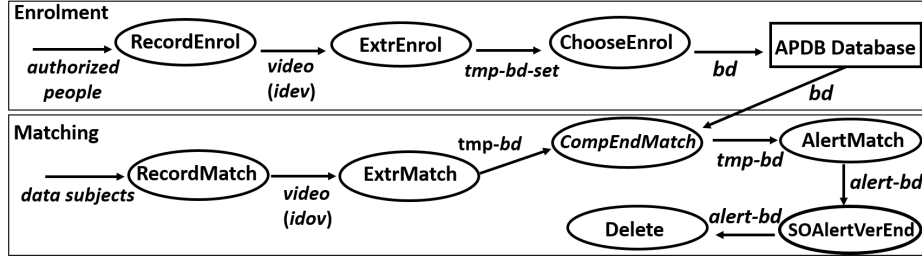


Fig. 5. Data flow graph extract, showing the relationships between some of the events

Traces and Abstract States

Sequences of abstract events are now defined. They constitute a history of personal data handling events, and will be used for compliance checking.

Definition 2 (Trace). A trace σ is a sequence of abstract events.

We provide the notion of abstract states to define compliant traces based on the semantics of events. The main difference between our formalism and the one proposed in [6] is as follows. In this previous work, the system stores information about the DS, such as IDs. Here, only IDs about the videos and bodyprints are stored. Hence, instead of defining abstract states on the pair of data types and DS, we define them on the pair of data types and video IDs. $\mathbb{P}(S)$ denotes the power set of S .

Definition 3 (Abstract state). The abstract state of a system associated with data types and video IDs $(Type, IDV)$ is a function $(Type, IDV) \rightarrow Time \times Cam \times \{Value\} \times Policy \times \mathbb{P}(Entity, \mathbb{N}) \times \mathbb{P}(Entity, \mathbb{N}) \times \mathbb{P}(Entity, \mathbb{N})$.

We distinguish abstract states with regard to video and non-video data types such as temporary bodyprints, stored bodyprints, alerted bodyprints. A state includes the time of the current state, the camera that recorded the video with the given ID, the current value of the data (video or non-video), the policy on this data, as well as the sets of SAs, SOs and Authorities who have been granted access to it. The associated value in \mathbb{N} specifies the trace position where the access to this data has been granted.

For instance, in case of *enr-video-type* and *tmp-bd-set-type* we have the states:

$$\begin{aligned} (enr-video-type, idev) &\rightarrow (t, cam, \{video\}, \pi_{ev}, sa, so, aud) \\ (tmp-bd-set-type, idev) &\rightarrow (t, cam, tmp-bd-set, \pi_{et}, sa, so, aud) \end{aligned}$$

The semantics of an abstract event at a given position in a trace is denoted:

$$\mathcal{S}_A : (Event \times \mathbb{N}) \rightarrow AbstractState \rightarrow AbstractState$$

$$\begin{aligned}
& \mathcal{S}_A((RecordEnrol, t, cam, idev, \pi_{ev}), j) \sum = \\
& \sum[(enr-video-type, idev) \rightarrow (t, cam, \{video\}, \pi_{ev}, \emptyset, \emptyset, \emptyset)] \\
& \mathcal{S}_A((ExtrEnrol, t, idev, tmp-bd-set-type, tmp-bd-set, \pi_{et}), j) \sum = \\
& \sum[(tmp-bd-set-type, idev) \rightarrow (t, cam, tmp-bd-set, \pi_{et}, \emptyset, \emptyset, \emptyset)] \\
& \mathcal{S}_A((ChooseEnrol, t, idev, bd-type, bd, \pi_{ea}), j) \sum = \\
& \sum[(bd-type, idev) \rightarrow (t, cam, \{bd\}, \pi_{et}, \emptyset, \emptyset, \emptyset)] \\
& \mathcal{S}_A((AlertMatch, t, idov, alert-bd-type, alert-bd, \pi_{alert}), j) \sum = \\
& \sum[(alert-bd-type, idov) \rightarrow (t, cam, \{alert-bd\}, \pi_{alert}, \emptyset, \emptyset, \emptyset)] \\
& \mathcal{S}_A((Access, t, or, idv, \theta, v), j) \sum = \\
& \sum[(\theta, idv) \rightarrow (t, cam, \{v\}, \pi, sa \cup \{(or, j)\}, so, aud)], \text{ if } or = SA, \text{ else} \\
& \sum[(\theta, idv) \rightarrow (t, cam, \{v\}, \pi, sa, so \cup \{(or, j)\}, aud)], \text{ if } or = SO, \text{ else} \\
& \sum[(\theta, idv) \rightarrow (t, cam, \{v\}, \pi, sa, so, aud \cup \{(or, j)\})], \text{ if } or = Aud, \\
& \text{where } (t, cam, \{v\}, \pi, sa, so, aud) = \sum(ds, \theta) \\
& \mathcal{S}_A((Delete, t, idv, \theta, v), j) \sum = \sum[(\theta, idv) \rightarrow \perp] \quad \text{where } (\theta, v) \in \nabla
\end{aligned}$$

Fig. 6. Abstract states semantics (extract)

Only an extract of the abstract state semantics is shown here for the sake of conciseness; see Fig. 6.

The semantics of the video recording event *RecordEnrol* is captured by an update of the state for the pair $(enr-video-type, idev)$ with the tuple of the recording time, the camcorder, and (the contents of) the video itself. At this time no access is allowed to the video yet, hence, three empty sets are included. Similarly, the semantics of the bodyprints extraction event *ExtrEnrol* is defined by a tuple including the set of temporary bodyprints *tmp-bd-set* extracted from the video *idev*. At the moment of bodyprints extraction, no access right has been granted yet. The event *AlertMatch* updates the state with the time *t*, the camcorder recorded the video *idov* of the alerted bodyprint, the value and the policy of the bodyprint. The semantics of the event *Access* is based on the value of *or*, who is granted access to the data (θ, v) . As a result, *or* is added to the corresponding set of SAs, SOs and Authorities, respectively. Finally, the event *Delete* captures the deletion of the data (θ, v) , updating the state with the undefined state \perp .

Having defined abstract events and abstract state semantics, we can now define the final state of a trace. This notion captures the combined effect of all personal data handling events up to the end of a trace. The final state of a trace $\sigma = [e_1, \dots, e_n]$ is defined as $F_A(\sigma, 1) \sum_0$ with $\forall \theta, \forall idv, \sum_0(\theta, idv) = \perp$ and

$$\begin{aligned}
& F_A([\], n) \sum = \sum \\
& F_A([e_1, \dots, e_m], n) \sum = F_A([e_2, \dots, e_m], n+1) \left(\mathcal{S}_A(e_1, n) \sum \right)
\end{aligned}$$

We set $State_A(\sigma, i) = F_A(\sigma_{|i}, 1) \sum_0$, with $\sigma_{|i} = \sigma_1 \dots \sigma_i$ the prefix of length *i* of σ (i.e. the partial trace up to index *i*).

Final states will in turn be used to specify trace compliance next (§5).

5 Compliance of Event Traces

We now define the compliance of event traces. This notion captures the accountable operation of the biometric surveillance system. Trace compliance rules A_1 – A_{12} are stated $\forall i \in \mathbb{N}, \forall idv, \forall \theta$. We first describe the rules in natural language, before formalising them in Fig. 7. These rules are not an attempt at exhaustiveness with regards to privacy compliance modelling. Rather, we aim to convey the importance of clarity and precision for privacy compliance rules.

- A_1 : No data v of type θ appears in an abstract state after the expiration of the global deletion delay.
- A_2 : Data v of type θ is used only for purposes defined in its policy.
- A_3 : During enrolment, if the policy forbids all access to data v of type θ , then there is none.
- A_4 : During enrolment, every access to the personal data by the SA must be preceded by the corresponding successful authentication.
- A_5 : Every access to the personal data by the SO must be preceded by the corresponding successful authentication (matching).
- A_6 : Every access to personal data by the auditor must be preceded by the corresponding access request.
- A_7 : During enrolment, the deletion of a video must occur within the duration $\pi_{ev}.d$ after a corresponding set of (temporary) bodyprints has been extracted.
- A_8 : Deletion of a video must occur within the duration $\pi_{mv}.d$ after a bodyprint has been automatically extracted from it for matching.
- A_9 : Deletion of a set of temporary bodyprints must occur within the duration $\pi_{et}.d$ after an adequate bodyprint has been chosen by the SA for storage in the APDB.
- A_{10} : Deletion of an automatically extracted bodyprint must occur within the duration $\pi_{mt}.d$ after the comparison of this bodyprint with the stored bodyprints (in the APDB) has ended.
- A_{11} : Deletion of a long-term bodyprint in the APDB must occur within the duration $\pi_{ea}.d$ after the system has retired.
- A_{12} : Deletion of the temporary bodyprints stored in the alert database must occur within the duration $\pi_{alert}.d$ after all alerts have been examined by the SO.

Let $EvTime$ be a function such that $EvTime(\sigma_i) = t_i$ and $\sigma_i = (X, t_i, \dots)$, $t_i \in Time$. Trace compliance rules are formally defined in Fig. 7. A_1 specifies that if in the current state of (θ, idv) the time is t , then there cannot be any event with a timestamp later than $t + \pi.gd$ since all data must have been deleted after this time. In A_2 if the state of data (θ, idv) at the $(i - 1)$ th event includes the policy π , then its use in the next event must comply with the defined purposes. A_3 specifies that if or accesses the data (θ, v) at the i th event, then this can only happen when previously the policy does not forbid access for or . Following this line of argument, the remaining rules can be interpreted in a similar way.

$$\begin{aligned}
A_1: & \text{State}_A(\sigma, i - 1)(\theta, idv) = (t, cam, \{v\}, \pi, so, sa, aud) \implies EvTime(\sigma_i) \leq t + \pi.gd \\
A_2: & \sigma_i = (Use, t, idv, \theta, v, purposes, reasons) \wedge \\
& \text{State}_A(\sigma, i - 1)(\theta, idv) = (t, cam, \{v\}, \pi, so, sa, aud) \implies purposes \subseteq \pi.ap \\
A_3: & \sigma_i = (Access, t, or, idv, \theta, v) \wedge \\
& \text{State}_A(\sigma, i - 1)(\theta, idv) = (t, cam, \{v\}, \pi, so, sa, aud) \implies \pi.acc^{or} \neq \uparrow \\
A_4: & \sigma_i = (Access, t, SA, idov, \theta, v) \implies \exists j \mid \exists t' \mid \sigma_j = (Authenticate, t', SA, idov, \theta, v) \wedge \\
& t' < t \\
A_5: & \sigma_i = (Access, t, SO, idov, \theta, v) \implies \exists j \mid \exists t' \mid \sigma_j = (Authenticate, t', SO, idov, \theta, v) \wedge \\
& t' < t \\
A_6: & \sigma_i = (Access, t, Auditor, idov, \theta, v) \implies \\
& \exists j \mid \exists t' \mid \sigma_j = (AccessReq, t', Auditor, idov, \theta, v) \wedge t' < t \\
A_7: & \sigma_i = (ExtrEnrol, t', idov, tmp-bd-set-type, tmp-bd-set, \pi_{et}) \wedge \\
& \text{State}_A(\sigma, i - 2)(enr-video-type, idov) = (t, cam, \{video\}, \pi_{ev}, sa, so, aud) \implies \\
& \exists j \mid \exists t'' \mid \sigma_j = (Delete, t'', idov, enr-video-type, video) \wedge (t' < t'' \leq t' + \pi_{ev}.d) \\
A_8: & \sigma_i = (ExtrMatch, t', idov, bd-type, bd, \pi_{mt}) \wedge \\
& \text{State}_A(\sigma, i - 1)(mat-video-type, idov) = (t, cam, \{video\}, \pi_{mv}, sa, so, aud) \implies \\
& \exists j \mid \exists t'' \mid \sigma_j = (Delete, t'', idov, mat-video-type, video) \wedge (t' < t'' \leq t' + \pi_{mv}.d) \\
A_9: & \sigma_i = (ChooseEnrol, t', idov, bd-type, bd, \pi_{ea}) \wedge \\
& \text{State}_A(\sigma, i - 1)(tmp-bd-set-type, idov) = (t, cam, \{tmp-bd-set\}, \pi_{et}, sa, so, aud) \implies \\
& \exists j \mid \exists t'' \mid \sigma_j = (Delete, t'', idov, tmp-bd-set-type, tmp-bd-set) \wedge (t' < t'' \leq t' + \pi_{et}.d) \\
A_{10}: & \sigma_i = (CompEndMatch, t', idov, tmp-bd-type, tmp-bd) \wedge \\
& \text{State}_A(\sigma, i - 1)(tmp-bd-type, idov) = (t, cam, \{tmp-bd\}, \pi_{mt}, sa, so, aud) \implies \\
& \exists j \mid \exists t'' \mid \sigma_j = (Delete, t'', idov, tmp-bd-type, tmp-bd) \wedge (t' < t'' \leq t' + \pi_{mt}.d) \\
A_{11}: & \sigma_i = (SysRetired, t') \wedge \\
& \text{State}_A(\sigma, i - 1)(bd-type, idov) = (t, cam, \{bd\}, \pi_{ea}, sa, so, aud) \implies \\
& \exists j \mid \exists t'' \mid \sigma_j = (Delete, t'', idov, bd-type, bd) \wedge (t' < t'' \leq t' + \pi_{ea}.d) \\
A_{12}: & \sigma_i = (SOAlertVerEnd, t', alert-bd-type, alert-bd) \wedge \\
& \text{State}_A(\sigma, i - 1)(alert-bd-type, idov) = (t, cam, \{alert-bd\}, \pi_{alert}, sa, so, aud) \implies \\
& \exists j \mid \exists t'' \mid \sigma_j = (Delete, t'', idov, alert-bd-type, alert-bd) \wedge (t' < t'' \leq t' + \pi_{alert}.d)
\end{aligned}$$

Fig. 7. Trace compliance rules

We note that accountability covers a huge number of requirements, hence, exhaustivity is beyond the scope of our paper. A sample set of compliance rules is provided to capture the most relevant aspects of accountability.

Compliance checking

Trace compliance is defined with respect to the previous rules:

Definition 4 (Trace compliance). *A trace σ is compliant if it satisfies all the properties A_1, \dots, A_{12} .*

The *Context* part of privacy policies did not appear in the above compliance checking rules. Generally speaking, context compliance may require manual verification by a human analyst. Even for time constraints (in our case, non-working hours), automated verification may not always be possible. Additional facts may need to be taken into account, or different context elements combined, with the final decision requiring individual appreciation. Since informal aspects can always crop up in compliance verification scenarios, manual verification must be integrated with formal verification in a single, coherent framework.

This formal definition of trace compliance can be used in practice by implementing a *log analyser*, i.e. a software tool taking as input a file containing a record of data handling events and outputting a **Compliant** / **Non-compliant** value. Data handling logs are files containing timestamped records of abstract events. They must be designed with compliance checking in mind to be usable. Such design is not trivial, and a balance must be found between semantic richness and the constraint of personal data minimisation. The issue of log design for compliance checking is explored in more detail in previous work [4].

In practice, logs generated by systems often contain events expressed at a lower level than the one relevant in conjunction with privacy policies. Events may be recorded at the system level and consist in a sequence of operations such as memory address reading, writing and deleting. However, logs at this level can also be used for compliance checking if certain conditions are met. A correspondence between different levels of abstraction is defined in [6], which makes it possible to apply the approach to low level logs.

6 Discussion

To the best of our knowledge, we have presented the first case-study of the application of a formal accountability framework to a biometric surveillance system. Our approach relies on the following building blocks:

1. A view of accountability as the provision, by a DC, of sufficient information to make the compliance of personal data handling verifiable to individuals or to auditors acting on their behalf.
2. The specification of distinct privacy policies for the different phases of operation of the system (enrolment and matching) and for the different categories of personal data involved.
3. The definition of abstract events, corresponding to the handling of personal data by the DC in a format compatible with the previously defined privacy policies.
4. The definition of a semantics of abstract system states for specific data types and values, and a distinction between the enrolment and matching phases.
5. Based on these abstract state semantics, the specification of data handling compliance rules for traces (sequences) of abstract events.

While formal models for accountability have been described before, this case study shows how such a framework can be tailored to a real-world setting involving biometric identification for surveillance. One specificity of this setting is that similar categories of personal data are handled in different ways depending on the stage of system processing. In this case, video frames and bodyprints are handled distinctly in different databases and at different operation stages.

Generally speaking, this example emphasises the importance of fine-grained distinction between the handling of personal data in different contexts. This need for clarity can be seen as one more argument for the importance of technical privacy policies, as opposed to privacy policies merely expressed in natural language, which are more prone to ambiguity.

Needless to say, strong measures must be taken to ensure the security (especially the confidentiality and integrity) of the log files. Their integrity must be guaranteed at two levels. At the time of their generation, log files must reflect actual system operation. It can be in the interest of DC to maliciously forge false traces to fake accountability while conducting non-compliant data handling operations. Even when this is not the case, implementation errors can break the link between system-level operations on personal data and the high-level narrative presented in the log files. Great care must therefore be taken to ensure this link is preserved. A possible technical approach is partial formal modelling of the components critically involved in log generation. In addition, the log generation process and the system itself must be documented with sufficient precision to make it possible for an auditor to check (manually) that the logs include all relevant events and form an adequate evidence for the operation of the system. Last but not least, after their generation, the integrity of log files must be preserved by preventing tampering.

To further elucidate the real-world applicability of our approach, demonstrating actual compliance checking is desirable. The feasibility of this mainly depends on the availability of exploitable logs. As discussed earlier, semantically useful log design is not obvious since both ambiguity and the presence of unnecessary personal data must be avoided. Logs must also be compatible with the chosen format of machine-readable privacy policies. Once usable logs are available, a log analyser can be implemented — a comparatively easy step in comparison if compliance semantics are well-defined. The implementation work then mainly involves a parsing module and coding compliance semantics. The task is more complicated if prior translation from system events to abstract events is required for the logs, and would need the definition of a correspondence relationship [6].

A case study fully incorporating legal and organisational aspects could be a worthwhile future work to further elucidate the concrete use of accountability.

Acknowledgement

This work was partially funded by the European project PARIS / FP7-SEC-2012-1, the Inria Project Lab CAPPRIS (Collaborative Action on the Protection of Privacy Rights in the Information Society) and the German Research Foundation (DFG).

References

1. PrivAcY pReserving Infrastructure for Surveillance (PARIS) Project, <http://www.paris-project.org>
2. Albiol, A., Albiol, A., Oliver, J., Mossi, J.: Who is who at different cameras: people re-identification using depth cameras. *Computer Vision, IET* 6(5), 378–387 (2012)
3. Article 29 Data Protection Working Party: Opinion 3/2010 on the principle of accountability (2010), http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2010/wp173_en.pdf
4. Butin, D., Chicote, M., Le Métayer, D.: Log Design for Accountability. In: 2013 IEEE Security & Privacy Workshop on Data Usage Management. pp. 1–7. IEEE Computer Society (2013)
5. Butin, D., Chicote, M., Le Métayer, D.: Strong Accountability: Beyond Vague Promises. In: *Reloading Data Protection*, pp. 343–369. Springer (2014)
6. Butin, D., Le Métayer, D.: Log Analysis for Data Protection Accountability. In: 19th International Symposium on Formal Methods (FM 2014), *Lecture Notes in Computer Science*, vol. 8442, pp. 163–178. Springer (2014)
7. Campisi, P.: *Security and Privacy in Biometrics*. Springer (2013)
8. Colin J. Bennett: *Implementing Privacy Codes of Practice*. Canadian Standards Association (1995)
9. Denman, S., Fookes, C., Bialkowski, A., Sridharan, S.: Soft-biometrics: Unconstrained authentication in a surveillance environment. In: *Digital Image Computing: Techniques and Applications (DICTA 2009)*. pp. 196–203. IEEE Computer Society (2009)
10. Jain, A., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 14(1), 4–20 (2004)
11. Kanak, A., Sogukpinar, I.: BioPSTM: a formal model for privacy, security, and trust in template-protecting biometric authentication. *Security and Communication Networks* 7(1), 123–138 (2014)
12. Lloyd, J., Jürjens, J.: Security Analysis of a Biometric Authentication System Using UMLsec and JML. In: 12th International Conference on Model Driven Engineering Languages and Systems (MODELS 2009). *Lecture Notes in Computer Science*, vol. 5795, pp. 77–91. Springer (2009)
13. Prabhakar, S., Pankanti, S., Jain, A.K.: Biometric Recognition: Security and Privacy Concerns. *IEEE Security & Privacy* 1(2), 33–42 (2003)
14. Salaiwarakul, A.: *Verification of Secure Biometric Authentication Protocols*. Ph.D. thesis, University of Birmingham (2010), <http://etheses.bham.ac.uk/1166/>
15. Saornil, M., Rodríguez, F.J., Montenegro, M., Ma, Z.: PARIS Project Deliverable 6.1: Biometrics Use Case Description (2014), http://www.paris-project.org/images/Paris/pdfFiles/PARIS_D6.1_Biometrics_Use_Case_Description_v1.0.pdf
16. Socolinsky, D.: Design and Deployment of Visible-Thermal Biometric Surveillance Systems. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. pp. 1–2 (2007)
17. Spanish Data Protection Agency: Instruction 1/2006 on processing personal data for surveillance purposes through camera or video-camera systems (2006), http://ec.europa.eu/justice/policies/privacy/policy_papers/docs/instrucciones_videovigilancia_en.pdf
18. Wheeler, F.W., Weiss, R., Tu, P.H.: Face Recognition at a Distance System for Surveillance Applications. In: *Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS 10)*. pp. 1–8 (2010)