Enhancing Video Surveillance with Usage Control and Privacy-Preserving Solutions

Enrico Carniani[1], Gianpiero Costantino[1], Francesco Marino[2], Fabio Martinelli[1]*, and Paolo Mori[1]
[1]*Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche via G. Moruzzi 1, Pisa, Italy*
{enrico.carniani, gianpiero.costantino, fabio.martinelli, paolo.mori}@iit.cnr.it
[2]*Scuola Superiore Sant'Anna, NoES-TeCIP, Pisa, Italy*
fr.marino@sssup.it

Abstract

Nowadays, the use of video surveillance systems for protecting critical environment is becoming more and more popular. On the one hand, video surveillance systems should be able to detect the presence of unauthorized people in the monitored environments to provide effective physical security. On the other hand, video surveillance systems should be also able to preserve the privacy of authorized people. Moreover, some scenarios require that the right of a person to stay in a monitored room depends on dynamic factors that change over time. To consider these aspects, our paper proposes a video surveillance framework based on the Usage Control model. This framework enforces security policies which continuously control that the people who enter the monitored environment have the permission to stay there. Then, we present two solutions to make our video surveillance framework privacy-preserving. The first solution considers that when the security policy is violated, the video surveillance system records the video stream captured by the video cameras installed in the monitored environment. The second solution uses the Secure-Two party Computation technique to identify the people in the monitored environment in a privacy-preserving way. In this paper, we present the architecture of the proposed framework, we provide an example of Usage Control policy in a real scenario and we describe the main details of our two implementations.

Keywords: Video Surveillance, Privacy, Usage Control, Secure Two-party Computation.

1 Introduction

Video surveillance systems are currently widely adopted both in business and public environments due to the need of security in critical environments. To this purpose, video surveillance systems with advanced features, such as humans tracking, have been proposed in the scientific literature, and they are already or they will be soon available on the market at affordable cost. Some companies adopt video surveillance systems to enhance the physical security of their premises by detecting unauthorized accesses. Video surveillance may be considered invasive for the privacy of the people, but it is needed as measure to guarantee people's security, for example to prevent terrorist attacks. Thus, effective video surveillance systems, besides simply detecting the presence of unauthorized people, should be able to preserve the privacy of authorized people who access the monitored environment. In addition, face recognition is heavily used in online photo albums and social networking platforms that have become popular for sharing photos with family and friends. These platforms support automatic detection and tagging of faces when images are uploaded.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 7:4 (December 2016), pp. 20-40
*Corresponding author: Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche via G. Moruzzi 1, Pisa, Italy, Tel: +39-050-3153425

This paper describes a video surveillance framework based on the Usage Control model, and it is an extension of a published short paper[1]. This new version provides more details on our video surveillance framework, focusing more on the prototype implementation, and a totally new contribution to enhance the privacy preserving capability of our framework using the Secure Two-party Computation (2PC) cryptographic technique.

Our framework aims at detecting people located in the monitored environment, identifying them, and continuously evaluating the Usage Control policy to check whether they have the right to stay in the monitored places. The Usage Control policy takes into account attributes, paired with such people and with the monitored environment, which change over time. Hence, the policy could be satisfied when a person enters the monitored environment, but a subsequent update of one or more attributes could cause a policy violation. So, as soon as the policy evaluation determines that one person is no longer authorized to stay in the monitored environment, an alert is raised, and the video stream is recorded.

The main contribution of this paper is the design and implementation of a working framework adopting the Usage Control model in the video surveillance scenario: in this way, the security policy is continuously evaluated while people are in the monitored environment, instead of being evaluated only to decide whether they are allowed to enter it. As reference scenario, we suppose that a company wants to enforce a security policy in which visitors must be always accompanied by an employee when they are in some rooms of the company. This is a Usage Control policy that can be naturally expressed and enforced by the proposed framework, since it requires that a given condition, i.e., the presence of an employee in the room, is continuously verified while the visitor stays in that room. In addition, this paper focuses on how the Usage Control model can be exploited to satisfy the security needs of the video surveillance scenario while guaranteeing users' privacy. To this aim, we propose two privacy-preserving solutions: the first is a soft solution that avoids to record the video stream when users are authorized by the Usage Control system to stay in the monitored room. Instead, the second solution enhances our usage-control system to privately compare faces using the Secure Two-party Computation (2PC) technique. Our solution can be used to extend the capability of the face recognition component by giving the possibility to authorize employees without knowing their real identity.

The structure of this paper is the following: §2 describes some related work, §3 describes some background concerning the Usage Control model and the technique we adopted for face recognition. Our Usage Control approach is presented in §4, instead §5 details the architecture of our framework and gives some details of the prototype we implemented. §6 extends our video surveillance part presenting a privacy-preserving solution to identifies people. Finally, §7 draws the conclusions.

2 Related Work

The following works make use of visual or cryptographic techniques to protect users' privacy. The authors of [1] propose the adoption of a PrivacyCams to encode video stream directly by cameras. Sensitive details are hided in the stream and only authorized people can access the raw data. More specifically, they suggest the adoption of level of authorizations to enable users to unveil sensible details depending on their authorization grant. Fidelao et al. [2] propose a privacy architecture for sensors, and in particular for video surveillance cameras. They suggest the use of a privacy buffer to detect and tag private data coming from sensors. Tagging of data is done by means of privacy filters that label the data as private or not private. Korshunov at al. in [3] borrow the warping algorithm from the animation and artistic fields to protect privacy of users in the video stream. The warping algorithm shifts Pixels into slightly different locations making the original face very difficult to be identified. Sohn et al. [4] propose something close

[1]This paper is an extended version of paper published in the 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing PDP 2016, Heraklion, Crete, Greece, 17-19 February 2016.

to Korshunov. In fact, they use the JPEG Extended Range (JPEG ER) technique to scramble users' face on surveillance video content. Their results demonstrate how JPEG ER provides privacy-sensitive face regions giving a good level of protection in a stream of 30fps with a resolution of 4CIF, i.e., $704x576$ pixel. Instead, Saini et al. [5] propose an interesting study on privacy leakage of a streaming coming from multiple cameras. The model they present establishes the privacy loss due to three main features: *what* (i.e., activities), *when* (i.e., time), and *where* (i.e., location). Moreover, the same authors propose a framework that implements anonymous surveillance by decoupling the knowledge on the context from the recorded video and requiring that the video are monitored remotely [6]. Serpanos et al. in [7] face security and privacy issues in distributed smart cameras. In particular, they point out privacy and security properties that video-surveillance systems should provide starting from the architecture set up. Winkler et al. in [8] presented on a survey on video-surveillance in which the authors discussed the state of the art in security and privacy protection, and show how these works cover security aspects into data-centric, node-centric, network-centric, and user-centric security.

The following works specifically use Secure Two-party Computation techniques to perform face recognition operations in a privacy preserving fashion. In [**?**] and [9] the authors use secure two-party computation and homomorphic encryption schemes to make the well-known face recognition algorithm Eigenfaces privacy preserving. In particular they use the Paillier additively homomorphic scheme [10] to make the server compute the encrypted values of the euclidean distances between the sample face image and all the face images stored in a database without disclosure of information between the client and the server. In [**?**] the minimum between the encrypted distances and the relative face image is found through the Damgård, Geisler and Krøigaard [11] [12] cryptosystem, while in [9] the authors use a garbled circuit. In [13] face images are represented as 900 bit vectors encoding spatial and appearance information, and the matching algorithm consists in finding the face image(s) in the database at Hamming distance from the sample face image below of a certain threshold. In this scheme client and server compute the encrypted value of the Hamming distance between each face image in the database and the sample image using additive homomorphic encryption and then perform oblivious transfer to figure out whether such distance is below of the predetermined threshold. Homomorphic encryption is also used in [14] to achieve privacy preserving image retrieval in a cloud computing environment and face recognition using SIFT.

The following works make use of policies to enable actions in specific conditions. Wickramasuriya et al. [15] suggest a solution that hides camera objects to keep users' privacy by specifying policies to access regions of the video stream. They also localise and authenticate users by means of RFID tag that, however, can be seen as a weak point since they may undergo identity transfers. Birnstill et al. in [16] make use of usage-control system to define policies that are differently evoked on two operational modes and regulate the actions to execute the video stream. In particular, the default mode occurs in a normal/quite situations, and here an operator is not able to see the video stream coming from the cameras. While, on the alarm mode the camera shows the video stream to the operator.

Differently from the above papers that make use of policies, our work propose a video-surveillance framework based on the Usage Control model with two solutions to preserve the privacy of the identified people. As far as we know, works as that one of Birnstill et al. in [16] use policies that are enforced to trigger actions to enable or disable the video recording. So, the alarm mode is not triggered according to an attribute state evaluated by the policy, but it is triggered by other external components. Then, the policies only manages the access to the video stream. Even in the work of Wickramasuriya et al. policies regulate the access to different region of the video stream, and they use RFID as authentication system. Instead, we propose a video surveillance framework that entrusts the management of users' identity to a face recognition algorithm in a privacy-preserving fashion, and uses Usage Control policies which take into account attributes of the users, of the monitored rooms and of the environment that change over time. In this way, besides preserving the privacy of the users authorized to enter the monitored rooms,

our framework is also able to detect when a user who was previously authorized to enter a room should leave that room because of a policy violation cause by a subsequent attribute update.

3 Background

3.1 Usage Control Model

The Usage Control (UCON) model improves the traditional access control models with *mutable attributes* and new decision factors besides *authorizations*, i.e., *obligations* and *conditions*. This section summarizes the main concepts of the UCON model; a detailed description can be found in [17, 18, 19, 20].

Mutable attributes represent features of subjects and resources (objects) that can be modified due to the access decision process or to the usage of resources. This could impact the rights of other accesses that are in progress [21]. For instance, the number of people in a room, which changes every time a person enters or exits the room, is a mutable attribute paired with the room, which is the resource that is accessed. *Immutable attributes*, instead, do not change their value frequently, and their modification is only done through administrative actions. For instance, the role of a person in a company is an immutable attribute and can be updated after a career advancement (e.g., from employee to department head). To face the mutability of attributes during the usage of a resource, the Usage Control model evaluates the policy before *(pre-evaluation)* and during the usage of that resource (*ongoing-evaluation*). The policy re-evaluation during the resources usage reduces the risk of their misuse when a given permission is no more valid.

Authorization predicates are needed to determine if a subject has the permission to access a specific object. To perform this check, the decision making phase uses subject/object attributes, and the actions on a object requested by a subject. The UCON model defines two categories of authorizations: pre-Authorizations (*preA*), here the decision phase is done when the subject requests to access the object, and ongoing-Authorizations (*onA*), here the decision phase is done while the access is in progress.

Obligation predicates state if specific requirements are fulfilled to access objects. In particular, Pre-oBligation (*preB*) predicates verify the requirements before the access, while ongoing-oBligations (*onB*) continuously check that the requirements are fulfilled.

Conditions are requirements that do not depend on subjects or objects. They evaluate environmental or system status, such as current time.

The UCON model has been successfully adopted in several scenarios, such as Web, Grid, or Cloud to protect the usage of several kind of resources. Sandhu et al [22] propose their model in collaborative computing systems, such as the GRID environment. Their model is based on a centralized repository for attribute management. Immutable attributes are managed in push mode (i.e., the attributes value is submitted to the authorization service by the user himself), instead the mutable attributes are managed in pull mode (i.e., the attributes value are collected by the authorization service just before their use).

The authors of [23] propose an Usage Control enforcement mechanism for applications. They propose a prototype to control data in a social network, and their mechanism allows the data owner to avoid that data would be printed, saved, copied&pasted, etc., from unauthorized users.

In [24], the authors propose the adoption of the Usage Control model to enhance the security of IaaS Cloud services. The proposed authorization system regulates the usage of the Virtual Machines provided by the IaaS service, and it goes beyond traditional authorization systems because it interrupts (suspends) the usage of running Virtual Machines when the corresponding rights do not hold any more.

The work in [25] tackles the problem of Usage Control of multiple copies of a data object when they are stored in distributed systems. Moreover, the Usage Control policies specifies the parties that will be

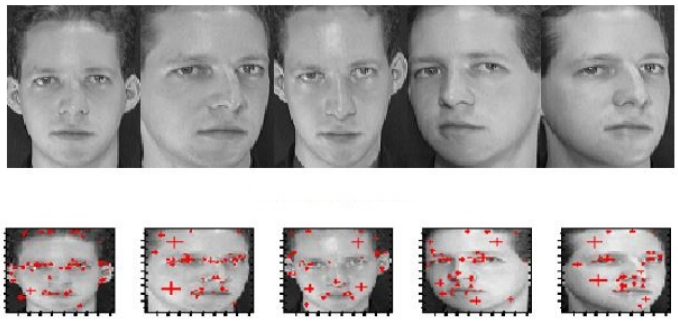


Figure 1: Locations of interest points with SIFT

involved in the decision process. In fact, a policy could be evaluated on one site, enforced on another, and the attributes needed for the policy evaluation might be stored in (many) different locations.

3.2 Face Recognition

Face recognition is one of the most common techniques used by humans for visual interaction. Face recognition is used in contexts like video surveillance, recognition of individuals in airports and border crossings, authentication and access systems. However, in these scenarios is often difficult to recognize faces, since faces can be affected by:

- aging;
- different facial expressions;
- changes in lighting and background;
- changes in position with respect to the camera;
- presence of glasses;
- partial occlusions of the face.

Despite these problems, in the last thirty years researchers have been active in designing and developing recognition systems as reliable and robust as possible to tackle the above issues. Face recognition methods are classified as: i) feature-based and ii) holistic.

3.2.1 Feature-based

Here, images are processed to have distinctive facial features such as the eyes, mouth, nose, and others. In addition, geometric relationships are calculated among those facial points to reduce the input facial image to a vector of geometric features. Finally, statistical pattern recognition techniques are applied to match faces.

Featured-based techniques are quite robust to position changes of the input image since features point are extracted a priori of the matching analysis of the image. However, the major disadvantage of these approaches is the difficulty to automatically detect feature.

SIFT. Scale-Invariant Feature Transform (*SIFT*) is an algorithm to detect and describe local features in images (Fig. 1). These are used to obtain reliable matching between different views of the same object and for this reason they are invariant to scale and orientation. SIFT follows the following four step to extract features:

Figure 2: Image transformation using Eigenfaces.

1. it localizes potential interest points in the image by detecting the maxima and minima of a set of Difference of Gaussian (DoG) filters applied at different scales all over the image;
2. it discards points with low contrast;
3. it assigns an orientation to each key point based on local image features;
4. it computes a local feature descriptor based on the local image gradient, and transforms the orientation of the key point to provide orientation invariance. Every feature is a vector of 128 dimension distinctively identifying the neighborhood around the key point.

To recognize a face with those others stored in a database, SIFT extracts the features from the available faces. Then, SIFT extracts the features from the new face and compares them with those ones found before. The face with the largest number of matching points is considered as the closest one. In particular, a feature is considered similar to another one when their distance is less than a specific fraction of the distance to the next nearest feature. This allows SIFT to reduce the number of false matches.

3.2.2 Holistic

Here, faces are identified using global representations rather than on local features of the face. Holistic techniques fall into two groups:

- Statistical: a 2D array represents the image and recognition is performed comparing the input face with all other faces stored in the database. A weak aspect of this approach is the classification in a high dimensionality space. For this reason, other techniques suggest the use of statistical dimensionality reduction methods to get the most meaningful feature dimensions before performing the recognition.
- Artificial Intelligence (A.I.): it utilizes neural networks and machine learning techniques to recognize faces.

It is important to point out that these techniques work under limited circumstances, like equal illumination, scale, pose, and so on, and in addition they are computationally very expensive.

As part of the Holistic techniques, we provide a brief overview of the SIFT and Eigenfaces algorithms since we use them within our video surveillance architecture for face recognition.

Eigenfaces. Eigenfaces is a facial recognition algorithm based on the dimensionality reduction approach of Principal Components Analysis (PCA). The basic idea is to treat each image as a vector in a high-dimensional space. Then, PCA is applied to the set of images to produce a new reduced subspace that captures most of the variability between the input images. The Principal Component Vectors (eigenvectors of the sample covariance matrix) are called the *eigenfaces*. Each input image can be represented as a linear combination of these eigenfaces by projecting the image onto the new eigenfaces space. Then,

identification is performed by matching the faces in this reduced space. The algorithm consists of the following steps:

1. it normalizes the M facial images contained in the considered dataset to line up the eyes and mouths resampled at the same pixel resolution. All images must have the same dimensions H x W;
2. it represents the images as (H x W)-dimensional vectors and computes the average face vector;
3. it subtracts the average vector from each vector;
4. it constructs the matrix A with dimensions (H x W) x M, containing in its rows the vectors obtained in the previous step;
5. it computes the eigenvectors of A^TA, corresponding to the eigenvectors of the covariance matrix AA^T relative to the largest eigenvalues, which are the ones responsible for the most of the variance in the dataset;
6. it selects and normalizes the K most significant eigenvectors, that become the eigenfaces;
7. it projects all the faces in the dataset in this K-dimensional space.

In the recognize step, Eigenfaces projects the face that must be recognized into the K-dimensional Eigenfaces space. Then, the distance between the input face and all faces stored in a database is computed, the face with the minimum distance is selected, and if this distance is less than a fixed threshold, therefore a match is found.

4 Usage Control in Video Surveillance

We based our video surveillance framework on the Usage Control in order to be able to enforce security policies for the entire time that a person is inside a monitored room. When cameras detect a person in the room, this person is identified and the policy is evaluated. At first, the video stream is temporary recorded into a buffer. During the identification phase, faces are detected from the video stream and they are compared with those ones stored in the Face Database to find an identification match. When a person is properly matched, some attributes, which belong to him/her, are retrieved from an Attribute server. The role of the identified person in the company is an example of such attributes. Then, these attributes are used to perform the pre-evaluation of the security policy. If the result of the evaluation is “deny”, the policy states that the person cannot stay in the room, and an internal alarm is trigged to force the person to leave the room. In parallel, the temporary video buffer is permanently stored as well as the current video stream captured by the cameras. Otherwise, if the pre-evaluation result is “permit”, the temporary video buffer is discarded, the current video registration is blocked and the person can stay in the room. However, while the person is inside the room, the ongoing-evaluation of the policy is executed and since some mutable attributes may change, these may cause a policy violation. In fact, as soon as the ongoing-evaluation of the policy results in a violation, the alarm is triggered, the video stream is recorded and the person is forced to leave the room.

The following example shows the advantages resulting from the adoption of a Usage Control based authorization system in video surveillance. We suppose that a company regulates the access to some critical rooms of its department requiring that guests can access them only during the working hours, and they are accompanied by an employee when they are in these rooms. This kind of policy cannot be easily expressed by traditional access control models because they allow to evaluate the policy at access request time only. Instead, the company requires that the authorization system continuously verifies that

Policy:	1
Target:	2
(o.id = ”CED”)	3
Rule-1:	4
target:	5
(a.id = ”enter(s, o)”)	6
pre-authorization:	7
(”employee” $\in$ s.role)	8
pre-update	9
(o.numEmployee++)	10
post-update	11
(o.numEmployee$--$)	12
Rule-2:	13
target:	14
(a.id = ”enter(s, o)”)	15
pre-authorization:	16
(”guest” $\in$ s.role) AND	17
(o.numEmployee $>$ 0) AND	18
pre-condition:	19
(e.workingTime = TRUE)	20
on-authorization:	21
(o.numEmployee $>$ 0)	22
on-condition:	23
(e.workingTime = TRUE)	24

Table 1: Usage control policy example

each person in the room is authorized to stay there. Thus, our framework is the right solution to enforce that policy since it is designed to continuously evaluate ongoing policies.

Table 1 shows a representation of this Usage Control policy in a human readable language. The object (denoted by o) is one of the rooms of the company that the Usage Control system must protect, and the subjects (s) is the person who enters the room. Entering the room is the action (a) performed on the object. The environment is represented by e. This policy is a subset of a policy that regulates the access to all the monitored room of the company. In particular, this policy exploits the following attributes: o.id represents the ID of the rooms, a.id represent the ID of the actions controlled by the policy, s.role represents the role of the subject in the company, o.numEmployee is a mutable attribute and represents the number of employees who are in the room in a given moment. The policy takes also into account an attribute of the environment, denoted by e.workingTime, which states whether the current time is within the working hours. The entire policy consists of two rules. *Rule-1* (lines 4-12) regulates the access of an employee to the data center room, and *Rule-2* (lines 13-24) regulates the access of guests in the same room. More specifically, line 3 defines the ID of the object protected by this policy, which is “CED”and it is the room that hosts the data center of the company.

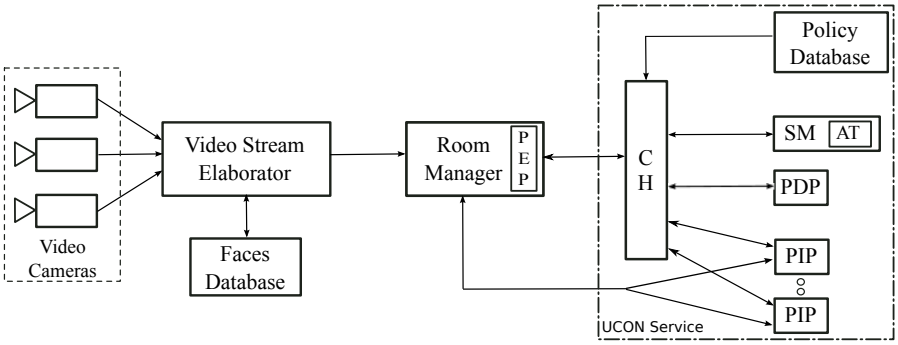


Figure 3: Architecture of the proposed surveillance framework

The target of *Rule-1* is the action of entering a room (line 6). Instead, the pre-authorization expressed in lines 7-8 allows a subject with role "employee" to execute the action. If the pre-authorization is satisfied, the pre-update defined in lines 9-10 is executed and the attribute *numEmployee* of the room is incremented. This value will be decremented by the post-update (lines 11-12) when the subject leaves the room.

Even, the *Rule-2* targets the action of accessing a room (lines 14-15). The pre-authorization (lines 16-18) states that a subject is allowed only if he/she has attribute role "guest", and the value of the attribute *numEmployee* paired with the room is greater then zero. The pre-condition (lines 19-20), instead, states that the current time must be within the working time to permit the access. The on-authorization in lines 21-22, is continuously evaluated and states that the value of the attribute *numEmployee* of the room must be greater then zero also during the access time. Finally, the on-condition (lines 23-24) requires that the current time must be within the working time while the subject is located in the room.

5 Architecture of the Video Surveillance Framework

This section describes the architecture of the proposed Usage Control video surveillance framework, which is shown in Figure 3. The *Video Camera*s (VCs) are installed in the rooms to be monitored, and they capture the video streams for the identification and video surveillance processes. In our reference scenario, we assume that a set of cameras are installed in such a way that they provide a total coverage of the monitored rooms. A set of guidelines for designing effective video surveillance scenario set up, called "adversary and scenario engineering", is presented in [26]. The cameras are connected to the *Video Stream Elaborator* (VSE), which is the component in charge of detecting faces in the video streams it receives. The VSE performs the face detection exploiting the Haar[2] feature-based cascade classifier, which is an effective object detection method based on a machine learning approach where a cascade function is trained with positive and negative images that are useful to detect objects in other images. The VSE applies the classifier to each frame it receives from the cameras. If one or more faces are found, the VSE executes the face recognition step using the SIFT algorithm (§3.2), in order to match the detected faces with those ones stored into the Faces Database.

If one person (or more) has been detected in the room, the VSE sends the related ID, say A, to the *Room Manager* (RM), along with other attributes of A retrieved from the Faces Database, e.g., the role of A in the company and the department he belongs to. In order to know whether A has just entered the room or he was already there, the RM manages a table that stores the IDs of the people currently in the room. If A has just entered the room, the RM triggers the embedded Policy Enforcement Point (PEP) to

[2]OpenCV, http://tiny.cc/t2la4x

interact with the Usage Control (UCON) service to perform the authorization process. If A was already in the room, instead, the RM simply sends the update command to the proper PIP in order to update the value of the attribute that stores the timestamp representing the last time A has been seen in the room. In addition, the RM is also in charge of detecting when a user leaves the room to notify the Usage Control service. The PEP interacts with the UCON service through the *tryaccess, permitaccess, denyaccess, revokeaccess, endaccess* messages, which are derived from the *Usage Control actions* defined in [18]. The PEP sends a *tryaccess* message to the UCON service every time a new person enters the room. This message includes the data collected by the RM (i,e., user ID, user role, room ID, etc). The UCON service evaluates the policy, and sends back a *permitaccess* or *denyaccess* response to the PEP. This enforces the *denyaccess* response by triggering an alarm and starting the video recording. In case of *permitaccess*, instead, the PEP does not perform any action because the person has the right to enter the room. When a person leaves the room the PEP notifies the Usage Control service by sending the *endaccess* message. In addition, the PEP starts the video recording when it receives the *revokeaccess* message from the Usage Control service, which states that the person in the room is no more allowed to stay there. Finally, the PEP is also in charge of enforcing the obligations included in the messages received from the Usage Control service, such as playing some announcements.

The Usage Control service consists of several components, namely: the *Context Handler*, the *Policy Information Points*, the *Policy Decision Point*, the *Session Manager*, and the *Access Table*.

The **Context Handler** (CH) is the front-end of the UCON service, and it is invoked by the PEP embedded in the RM. It manages the components of the UCON service to execute the *pre-evaluation* and the *ongoing-evaluation* of the policy. In particular, when the CH receives the *tryaccess* message, it performs the *pre-evaluation* of the policy. To this aim, it retrieves further attributes that could be exploited in the decision process by querying the **Policy Information Point**s (PIPs). These attributes are related to the user who entered the room, to the room itself, and to the environment. The PIPs, in turn, interact with the real providers of such attributes, called Attribute Managers (AMs), according to the specific protocols they provide. The CH can be configured to interact with any number of PIPs, in order to be able to manage all the AMs available in each specific scenario. The CH contacts the Policy Decision Point (PDP) sending the access request message enriched with the values of the attributes just collected. The PDP evaluates this request against the pre-policy (i.e., the policy including pre-authorization, pre-conditions, pre-updates and pre-obligations only), returning the access decision to the CH. If the access decision includes pre-updates of some attributes, the CH executes them by exploiting the *update* interface of the PIPs which manage these attributes. Finally, the CH forwards the access decision to the PEP. If the PDP decision is *allow*, i.e., the person is allowed to enter the room, the CH begins the *ongoing-evaluation* of the policy. Hence, the CH subscribes the attributes required for the decision process exploiting the *subscribe* interface of the PIPs. Consequently, the PIPs provide the current values of the attributes they manage to the CH, and will notify the CH as soon as one (or more) of these attributes has changed its value. Then, the CH sends the corresponding enriched access request to the PDP to evaluate the ongoing-policy (i.e., the policy including on-authorization, on-conditions, on-updates and on-obligations only), which, in principle, is different from the pre-policy. If the response of the PDP is *denyaccess*, the CH sends the *revokeaccess* message to the PEP, which starts the video recording. Otherwise, no action is taken. Due to the attribute subscriptions, a PIP could notify the CH that the value of an attribute is changed. Even in this case, the CH coordinates the components of the Usage Control service to perform the ongoing-policy re-evaluation exploiting the updated values of the mutable attributes. If the evaluation of the ongoing-policy returns *denyaccess*, i.e., the person in the room loses the right to stay there, the CH sends the *revokeaccess* message to the PEP that starts the video recording.

The **Policy Decision Point** (PDP) is a XACML evaluation engine following the format defined by the XACML standard [27]. Hence, given a policy and an access request, the PDP evaluates the policy for that request, and returns the decision: *allow* or *deny*. As previously described, the CH invokes the

PDP each time a pre-policy or an ongoing-policy must be evaluated.

The **Session Manager** (SM) is in charge of keeping trace of the ongoing usage sessions in order to implement the ongoing-evaluation of Usage Control policies as a consequence of an attribute update. In fact, each time an attribute changes its value, the SM is invoked to determine for which of the sessions that are in progress the ongoing-policy must be re-evaluated. In particular, the SM exploits the Access Table (AT) to store data about ongoing sessions. The CH invokes the SM to create a new entry in the AT for each access request (*tryaccess)* that is allowed by the pre-evaluation phase, setting the session status in the entry to "active". The SM is then invoked to modify the session status from "active" to "revoked" when the *revokeaccess* message is sent to the PEP because the ongoing-evaluation of the policy results in a policy violation, i.e., the right of the person to be in the room is no longer valid. Moreover, the SM deletes the entries linked to ended accesses due to an *endaccess* message. Finally, the CH invokes the SM when the values of an attribute is changed. In this case, the SM retrieves from the Access Table the list of the sessions that could be affected by this change, and sends it to the CH, which performs the re-evaluation of the ongoing-policy exploiting the updated attribute values for each of them.

The **Access Table** (AT) keeps meta-data about accesses in progress, i.e., the active usage sessions. The Access Table is implemented through a Database. Each entry refers to an active session, and it stores a set of data such as the session ID, the access request, and the session status (i.e., pending, active, ended, revoked). As previously explained, these entries are created by the SM when new sessions begin, are read by the SM when the value of attributes change, and are deleted when the related accesses are terminated.

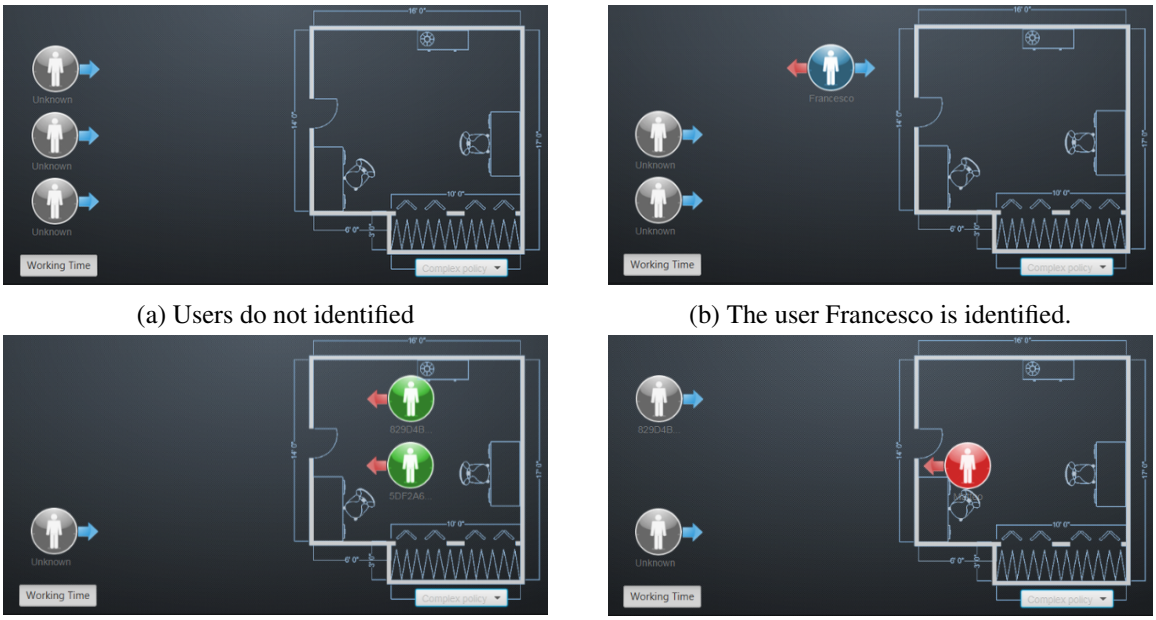
5.1 Architecture Development

This section describes some details related to the implementation of the surveillance framework depicted in Figure 3. In our framework, we used a D-Link DCS-942L IP camera, which is able to capture up to 20 frames per second with a resolution of 640x480 pixels. The camera is accessible using the HTTP protocol, and also it provides a built-in night vision with sound and motion detection capabilities. The Video Stream Elaborator is a software component developed as a Java application, and it uses JavaCV[3] to implement the functions for real-time computer vision, to access the camera video stream and to perform face detection operations. Each face detected by the VSE is compared with the faces available in the *Faces Database* to find a match (if it exists). The Faces Database contains a jpg picture of about 200x200 pixels per each user, along with user attributes (i.e., name, role, etc.). The Faces Database is implemented using MySQL as DataBase Management System[4], and the face recognition operation is executed using SIFT (§3.2.1) included in *Fiji*[5], which is a Java image processing package for image transformation, registration and interpretation.

We adopt SIFT to identify faces due to its robustness to variation in scale and orientation. In our test we observed that SIFT achieves a true positive rate near to 100% when the pictures in the database have good lighting conditions and quality of the captured stream is high without shades. That percentage decreases proportionally with the variation of those conditions. However, it is worth noting that the number of false positive remains negligible even in different conditions.

After the face identification, the VSE shares the *userID* and the *role* attributes with the RM, and the latter interacts with the the UCON service through the embedded PEP. Both the RM and the UCON service are entirely developed in Java. The PEP exploits the remote procedure call mechanism provided by the Apache XML-RPC library (v.3) to interact with the UCON service, in particular with the CH. Communications between PEP and CH use the XML-RPC protocol over HTTP since in our testbed the

[3] JavaCV, https://github.com/bytedeco/javacv
[4] MySQL, https://www.mysql.com
[5] Fiji, http://fiji.sc/Fiji



(a) Users do not identified

(b) The user Francesco is identified.

(c) A guest with an employee inside the room.

(d) The authorization for the guest is revoked.

Figure 4: Identification and authorization states recorded by our Video-Surveillance application.

RM and the UCON service are deployed on the same machine. However, HTTPS is also supported by our prototype, and it should be adopted when the RM and the UCON service are deployed on distinct machines.

The PDP implementation is based on the WSO2 Balana[6] open source software. Balana implements the XACML engine, while the CH invokes the PDP by means of its API that follows the XACML standard.

The SM is the component that manages the database to store the data related to the ongoing sessions. In particular, the SM implements the Access Table exploiting a Data Abstraction Layer based on the Oracle Java Persistence Architecture (JPA) v2.1 [28], which is an Object Relational Mapping (ORM) abstraction. This allows the SM to work with distinct DataBase Management System by properly changing a configuration file. In our reference implementation, the SM exploits the Apache Derby[7] DataBase Management System.

We implemented two PIPs in our prototype. The first PIP manages the number of employees present in each monitored room. The second PIP says whether the current time is within the working hours or not. Their interfaces provide remote methods to perform the *retrieve, subscribe/unsubscribe and update* operations. These PIP communicates with the CH through XML-RPC over HTTP calls, since they are located on the same machine of the CH.

5.2 Architecture Deployment and Evaluation

To validate our video surveillance framework, we protected a room of our department with the prototype we developed. When a person enters the room, the continuous face recognition is performed through the cameras, and the system governed by our Usage Control service enforces the policies shown in Table 1. Policies are enforced exploiting the following attributes:

[6]Balana, http://xacmlinfo.org/category/balana/
[7]Apache Derby, http://db.apache.org/derby/

1. the role of a person;
2. the number of employees inside the room;
3. if the current time is within the working time.

The role attribute is retrieved from the Face Database during the face recognition phase, while the remaining two attributes are retrieved using the following PIPs:

- *PIPNumEmployee*: this PIP is in charge of counting how many employees are currently within the room. The value is kept in a attribute that is incremented and decremented by the policy.
- *PIPWorkingTime*: this PIP is in charge of keeping a boolean value indicating whether the current time is within the working time or not.

To indicate the presence of users outside/inside the room, we developed a dashboard application, which is linked to the Usage Control module and video cameras. This application shows the position of the users inside or outside the room. When a user accesses the monitored room the camera detects his/her face and SIFT executes the face identification, by matching the face detected with those ones already available in the faces database. The configuration of SIFT that we set up in our testbed framework allows the Video Stream Elaborator to recognize faces with a number of false positive close to zero. This result is achieved by setting a high accuracy threshold in SIFT when it matches faces. Thus, when our Video Stream Elaborator component identifies a person, it gives a very high certainty that the user identified is that one registered in the faces database. However, this high level of accuracy requires that the quality of the video stream captured by the camera is very good and clean. So, blurry images are discarded by the Video Stream Elaborator during the face recognition phase.

The dashboard application that we developed is thought to be used by a security operator to monitor the activity in one or more rooms. The dashboard application represents each user with an icon that is automatically moved inside or outside the room by the Room Manager according to the position of the user. The state of a user is represented by our application using different colors for the icon:

- grey: the user is outside the camera view;
- blue: the user face is being processed;
- green: the user is authorized to stay in the room;
- red: the user is not authorized to stay in the room.

If the Video Stream Elaborator identifies more than a single face in the stream, it is able to correctly identify the faces as done in a single face identification setting, and even in this situation, false positive percentage is negligible, provided that the video stream should present high quality of images without shades.

In the empirical tests we conducted in our laboratory, we involved two actors, i.e., an employee and a guest. Fig. 4a shows all users outside the room with no identification, so they are labelled as *unknown*. When a user gets in the room, the Video Stream Elaborator correctly identifies the user *Francesco*, see Fig. 4b. In parallel, the Room Manager sends a *tryAccess* with the related *userID* and *role* to the UCON; the UCON successfully returns a *Permit* allowing the user to stay in the room, see Fig. 4c. In this phase, the Room Manager obscures the identity of the employee with a randomly generated identifier and the room recording state is moved to OFF to protect employee's privacy.

Always in Fig. 4c, the Video Stream Elaborator highlights the presence of another user. He is identified as a guest and he is granted by the Room Manager to stay in the room only because an employee is already inside the room during the working time range, as policy number two states. When the employee leaves the room (Fig. 4d), the policy is not matched anymore and the guest is obliged to leave the room. To inform the guest of this condition, we use a speaker inside the room that notifies the guest about the new authorization state, and the guest must leave the room. For security reasons, the cameras inside the room start recording each action done by the guest.

6 Privacy-preserving face recognition

The use of video-surveillance into departments, offices, banks and so on, is seen as a very sensible aspect since employers cannot leverage on video-surveillance in any situations. Often, employers use cameras for security purposes but at the same time cameras are forbidden if used to monitor employees' activity[8]. In general, the rapid improvement and widespread deployment of video-surveillance technology raises strong concerns regarding the violation of individuals' privacy. In fact, biometric information can be collected and misused to profile and track individuals against their will. For these reasons, the interest in privacy-preserving face recognition systems is increasing and to tackle this aspect, we propose a privacy-preserving solution that aims to authenticate users/employees keeping private their identity during the recognition phase. Thus, we designed our privacy-preserving solution to work in compliance with the architecture defined in Figure 3. However, we force the Video Stream Elaborator to not match in clear the face recognized with the Faces Database, instead it runs a Secure Two-party Computation (2PC) session with a new component placed in the next to the Face Database. In this way, we build a client-server infrastructure in which the client is represented by the new sub-component installed in the VSE, while the server resides in a new component that is placed just before the Face Database and physically available far away from the VSE or outsourced to a third-party, see Fig. 5. We repeat saying that the goal of this two components is to run a 2PC in which an employee is authenticated without knowing his/her real identity. So, the VSE will know, after the 2PC session, the content of a boolean value saying whether the employee was identified or not. If the VSE receives a *true* value, then the Room Manager that will proceed (or deny) the authorization.

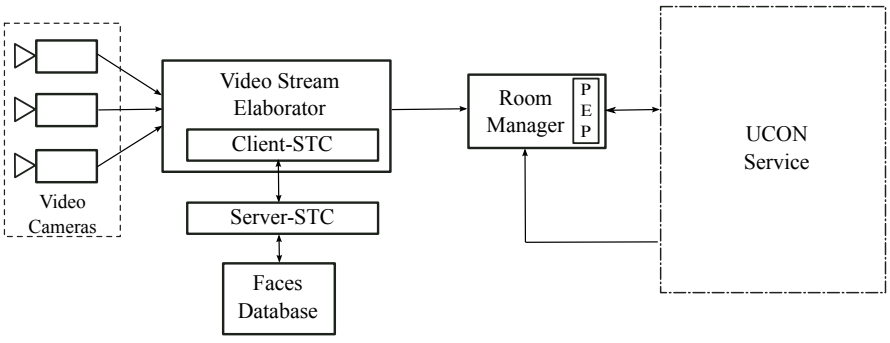


Figure 5: Architecture extended with 2PC components

[8] http://files.findlaw.com/pdf/employment/employment.findlaw.com_workplace-privacy_can-employers-use-video-cameras-to-monitor-workers.pdf

6.1 Secure Two-Party Computation

Secure Two-party Computation is a sub-field of cryptography that allows two users to jointly compute a function using their own inputs, and keeping those inputs private. The classical secure two-party computation example is the Millionaires' problem expressed in [29]. Here, two millionaires are interested in knowing which of them is richer without revealing their actual wealth. In order to accomplish this kind of tasks a number of protocols have been proposed in the past years. The first one is the Yao's protocol. The attacker model considered in this scenario is the "semi-honest". In this context, the two parties run the protocol exactly as specified (no deviations, malicious or otherwise), but they may try to learn as much as possible about the input of the other party from the protocol flow.

6.1.1 Garbled circuits

Garbled circuits are encrypted boolean circuits that play a central role in secure two-party computation. In these circuits the boolean values are replaced by encrypted signals, and the gates provide enough information to obliviously evaluate the circuits gate-by-gate to produce a garbled output. In a garbled circuit, each wire is associated with a pair of random binary strings called wire signals, corresponding to the 0 and 1 bit values in the boolean circuit. These values are known as the semantics of the signals since this reflects the meaning of the signal in the "cleartext" circuit. The correspondence between the wire signals and their semantics is random and is kept secret. Each two-input gate in the garbled circuit consists of four gate labels presented at random order that corresponds to the four possible input values. Such labels are stored in garbled computation tables. A garbled circuit is made by the garbled computation table of all its gates, the description of their interconnections, and the output description tables that map the random values on circuit-output wires to their corresponding real values. In order to evaluate a gate for a given pair of input wire signals it is necessary to locate the gate label corresponding to such signals, then some computations involving the gate label and the input signals are performed and the output wire signal is obtained. In more detail, the output wire signal is retrieved by applying two consecutive decryption processes on the correct gate label using the related wire signals. The garbled computation tables are designed in such a way that it is possible to get the gate label corresponding to a certain pair of wire signals without revealing anything about the value it is bounded to.

6.1.2 Oblivious Transfer

In cryptography, an oblivious transfer protocol (often abbreviated OT) is a type of protocol in which a sender transfers one of many pieces of information to a receiver, but remains oblivious as to what piece (if any) has been transferred. A useful form of oblivious transfer that finds application in the field of Secure Two-party Computation is the called 1-out-of-2 oblivious transfer. Such a protocol is used in a scenario involving a sender, who has two messages, and a receiver, who can chooses which one of the messages he wishes to receive. By running this protocol the sender can send to the receiver the message selected by this one without knowing which of the messages is, and without letting know anything to the receiver about the other message.

6.1.3 Yao's protocol

The sender and the receiver have respective inputs x and y, and wish to compute the output f(x,y).

1. the sender constructs a garbled circuit and sends it to the receiver;
2. the sender and the receiver interact so that the receiver gets the input-wire keys that are associated with the inputs x and y. In particular, the sender simply sends to the receiver the keys that are

associated to its own input and runs with him a 1-out-of-2 oblivious transfer protocol. During the OT protocol the receiver learns the garbled values of the input wires corresponding to its input;

3. finally, the receiver computes the garbled circuit as described above and concludes the protocol.

6.1.4 CBMC-GC

CBMC-GC is a compiler for C programs in the context of Secure Two-party Computation (2PC). It compiles a C program that specifies a secure computation into circuits that can be read in by an 2PC platform, which then performs the secure computation between two parties A and B. At the beginning of the compilation, CBMC-GC translates the C program into an intermediate representation featuring a simplified set of control statements. This program is then translated into circuits by symbolically executing the program. To perform the symbolic execution CBMC-GC unrolls loops and recursive function calls up to a predetermined bound. This bound is either determined automatically via an internal static analysis of the program or is given by the user in case CBMC-GC is not able to discover the bound. During the symbolic execution each statement (i.e., "if" or "for" blocks) is translated into a subcircuit, which represents the semantics of the respective statement. In such a way, the input wires to these subcircuits represent the current program state, and the output wires represent the state of the next program after the application of the statement in the current program state.

6.1.5 Attacker Model

The attacker model that is considered in the Secure-Two party Computation framework CBMC-GC is *honest-but-curious*. In this model, the attacker follows all protocol steps deigned in the 2PC session, but he/she can try to learn additional information about the other party during message exchanging phase, with the purpose to acquire some details of the identified person.

Moreover, it is important to notify that when executing a 2PC session, there is an asymmetry on the provided security guarantees. In particular, it is not possible to prevent one party from quitting the protocol prematurely, and not sending the result of the computation to the other party. We know that this situation is a weakness but we cannot overtake it due to 2PC specifications flow.

6.2 The implementation

Our privacy-preserving face recognition solution, which integrates 2PC, leverages on the CBMC-GC framework and it is tested using both SIFT and Eigenfaces algorithm for face recognition. In particular, our SIFT implementation is developed using "mpicbg[9]", which is a JAVA open source collection of algorithms focusing on image transformation, registration and interpretation. Instead, for Eigenfaces, which involves a lot of computations on matrices, we used "colt[10]" to get a set of JAVA open source libraries for high scientific performance.

During the implementation of our face recognition solution with support of 2PC, we faced a number of issues related to the complexity of the integration of the Secure Two-party Computation technique in conjunction with the face recognition algorithms. Our first attempt of integration was the porting of both face recognition algorithms in C to compile them with CBMC-GC and generate the garbled circuits needed to run a 2PC session. However, the CBMC-GC compiler was not able to compile both algorithms due to their structure complexity. In fact, to compare two images using SIFT, it requires a feature-by-feature match and SIFT extracts about 500 features per image and each feature consists of an array of

[9] `http://fly.mpi-cbg.de/~saalfeld/Projects/`
[10] `https://dst.lbl.gov/ACSSoftware/colt/`

128 values: so each matching operation involves about 250.000 comparisons between features, and each comparison consists in the calculation of the difference vector between 128-dimensional vectors. Slightly better is the Eigenfaces case, projecting a facial image of "P" pixels into the N-dimensional eigenfaces space, involves "N" scalar products between P-dimensional vectors (in our case, $P = 300x250 = 75.000$). Thus, in both cases it was not possible to generate the necessary circuits to run a secure computation session. To overcome these issues, we decided to use a simplified version of the two algorithms, and to reduce the portion of the algorithms run in the privacy-preserving mode. For SIFT, we reduced the number of features describing an image and the length of the feature descriptors, but this led to an unacceptable reduction of the algorithm performances in terms of accuracy. Due to the limitations occurred with the integration with SIFT, we decided to use Eigenfaces but leaving the part related to the face projection in the eigenfaces space not involved in the secure computation. So, we designed and implemented the following face recognition privacy-preserving flows:

1. *Pre*-2PC session:
   (a) the Server shares with the Client the average face and the eigenfaces computed over the Face Database. The Client will use this to project its image into the eigenfaces space;
   (b) the Server projects all images contained in the Face Database into the eigenfaces space;
   (c) the Client projects its image, which it wants to recognizes, into the eigenfaces space;
2. *During*-2PC session:
   (a) the Server and the Client run a Secure Two-party Computation session.
   (b) for each image in the Face DataBase, the Server gives as input the projection of that image, the ID of the image, the corresponding Role and a threshold[11];
   (c) the Client gives as input the projection of the face it wants to authenticate and a random integer number;
   (d) When the 2PC session ends, two cases are possible:
      i. *Match Found*: the 2PC session produces three integer numbers in the form:

$$\{0,1\};\{0,1\};123456 \tag{1}$$

The number "1" in the first position indicates that a matching is found, "0" otherwise. The integer in the second position indicates: "1" that he/she is an employee, "0" that he/she is a guest. This integer is got during the execution of the 2PC session and provided by the Server from the Face Database. The third integer is a fake ID number that is obtained by summing the real ID of the picture saved in the Face Database and the number sent by the Client. Even if the Server gets this sum, it does not know the number sent by the client since it is not unveiled in the 2PC. So, the sum result is meaningless for the Server. Instead, the Client gets the real ID by subtracting to the fake ID the random ID number that only it knows.
At this point, the Client stores the real ID number in a local table and associates a new random[12] ID to be communicated to the Room Manager. In this way, each time that the Client receives a fake ID from the 2PC session, it first calculates the real ID, checks in its table the associate random ID, and then communicates it to the Room Manger.

[11] This threshold establishes the accuracy of the identification by calculating the Euclidean distance between the two projections.
[12] This new ID is generated only once by the Client each time that it receives an ID not already stored in its local table. Then, the random ID is sent to the Room Manager that will work with this ID without knowing the real ID associated to the identified employee/guest.

ii. *Match NOT Found*: the 2PC session produces one simple integer in the form:

$$\{0,1\} \tag{2}$$

Where, the number "0" confirms that no match is found.

6.2.1 Results

To test our solution, we used a Face Database of 20 images 125x150px, each represented as a linear combination of 11 eigenfaces. Since CMBC-GC can only produce circuits operating on integer values, we multiplied per 100 the vectors representing the faces in the database, and we considered only the integer part of the results for the inputs during the 2PC session without leading to any observable decrease of performances.

Once the VSE detects the face in the video stream, it extrapolates the position of the eyes[13] and use it to transform the face in a format compliant with the face stored in the Face Database. Then the 2PC session is run, and the boolean result is got in about 25 seconds in a privacy-preserving fashion compared to the 6sec. needed to run Eigenfaces without the privacy-preserving feature.

Concluding, it is worth noting that our tests highlights the sensibility issue of Eigenfaces about the variation in lighting and orientation reported in literature as come out from the Holistic approaches.

6.2.2 Considerations

From the results obtained in our tests, we observed that the time needed to run a face recognition session with the privacy-preserving technique is not negligible. However, the effort employed in obtained this such a feature is quite relevant since the use of a privacy-preserving approach can limit the criticality that face recognition has in some context, like offices and so on.

To get a proper integration with our Usage Control system, we put a lot of effort to avoid that the server achieves any information about the identified person. Although this property is kept by the 2PC technique, our Usage Control system needs some details to continuously monitor people in the room. In particular, the Room Manger must know from the VSE the "ID" of the employee and his/her role. But, if the Server knows the ID at the end of the 2PC session it may understand which person was identified. To overtake this issue, we use the integer number sent by the Client at the step 2.*C* that is summed to the real ID of the employee. At the end, only the Client, which knows the random number, is able to get the real ID of the employee, instead the Server receives a meaningless number.

7 Conclusion

In this paper, we proposed the adoption of a Usage Control based authorization system to establish whether one o more people are authorized to access and stay in a monitored room. We validated our approach by developing a prototype of the proposed architecture and testing the enforcement of the policy described in Table 1. Components of our prototype where tested to verify that people who accessed the room were correctly authenticated and their behavior corresponded to what the policy declared. On the contrary, if the policy is not respected, we verified that the Usage Control system was able to detect such violations.

We presented a flexible architecture composed by two main parts: the first involves the authentication step, while the second the Usage Control. The choice to provide an architecture based on separate modules allows us to modify one part without altering the other. With this modular architecture we

[13] Specifically, the more aligned the eyes are, the more accurate the eigenfaces algorithm is.

designed and deployed to privacy preserving solutions: in the first solution, we record the video stream only in case of policy violation to preserve the privacy of the monitored people. Instead, in the second solution, we added two additional components to make our face recognition phase in privacy preserving manner using the Secure-Two party Computation techniques.

Finally, we presented the results of our implementations showing that the 2PC technique helps to keep private employees' identification, although it adds a not negligible computational time.

8 Acknowledgments

Work partially supported by the CNR Smart Campus Project, the H2020-MSCA-ITN-2015-NeCS EU project (GA #675320) and the PRIN project *Security Horizon*.

References

[1] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 50–57, May 2005.

[2] D. A. Fidaleo, H.-A. Nguyen, and M. Trivedi, "The networked sensor tapestry (nest): A privacy enhanced software architecture for interactive analysis of data in video-sensor networks," in *Proc. of the 2nd ACM International Workshop on Video Surveillance & Sensor Networks (VSSN'04). New York, New York, USA*. ACM, October 2004, pp. 46–53.

[3] P. Korshunov and T. Ebrahimi, "Using warping for privacy protection in video surveillance," in *Proc. of the 18th International Conference on Digital Signal Processing (DSP'13), Santorini, Greece*. IEEE, October 2013.

[4] H. Sohn, W. De Neve, and Y. M. Ro, "Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in jpeg xr," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 2, pp. 170–177, February 2011.

[5] M. Saini, P. Atrey, S. Mehrotra, and M. Kankanhalli, "W3-privacy: Understanding what, when, and where inference channels in multi-camera surveillance video," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 135–158, January 2014.

[6] M. Saini, P. Atrey, S. Mehrotra, and M. Kankanhalli, "Anonymous surveillance," in *Proc. of the 2011 IEEE International Conference on Multimedia and Expo (ICME'11), Barcelona, Spain*. IEEE, July 2011, pp. 1–6.

[7] D. N. Serpanos and A. Papalambrou, "Security and privacy in distributed smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1678–1687, October 2008.

[8] T. Winkler and B. Rinner, "Security and privacy protection in visual sensor networks: A survey," *ACM Computing Survey (CSUR)*, vol. 47, no. 2, pp. 1–42, July 2014.

[9] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *Proc. of the 12th International Conference on Information, Security and Cryptology (ICISC'09), Seoul, Korea*, ser. Lecture Notes in Computer Science, vol. 5984. Springer Berlin Heidelberg, December 2010, pp. 229–244.

[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT'99), Prague, Czech Republic*, ser. Lecture Notes in Computer Science, vol. 1592. Springer Berlin Heidelberg, April 1999, pp. 223–238.

[11] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Proc. of the 12th Australasian Conference on Information Security and Privacy (ACISP'07), Townsville, Australia*, ser. Lecture Notes in Computer Science, vol. 4586. Springer Berlin Heidelberg, July 2007, pp. 416–430.

[12] I. Damgard, M. Geisler, and M. Kroigard, "A correction to ëfficient and secure comparison for on-line auctions," *International Journal of Applied Cryptography*, vol. 1, no. 4, pp. 323–324, August 2009.

[13] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "Scifi - a system for secure face identification," in *Proc. of the 31st IEEE Symposium on Security and Privacy (SP'10), California, USA*. IEEE, May 2010, pp. 239–254.

[14] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Image feature extraction in encrypted domain with privacy-preserving sift," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4593–4607, November 2012.

[15] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, "Privacy protecting data collection in media spaces," in *Proc. of the 12th Annual ACM International Conference on Multimedia (MULTIMEDIA'04), New York, New York, USA*. ACM, October 2004, pp. 48–55.

[16] P. Birnstill and A. Pretschner, "Enforcing privacy through usage-controlled video surveillance," in *Proc. of the 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'13), Kraków, Poland*. IEEE, October 2013, pp. 318–323.

[17] J. Park and R. Sandhu, "The $UCON_{ABC}$ usage control model," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, pp. 128–174, February 2004.

[18] X. Zhang, F. Parisi-Presicce, R. Sandhu, and J. Park, "Formal model and policy specification of usage control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 4, pp. 351–387, November 2005.

[19] A. Pretschner, M. Hilty, and D. Basin, "Distributed usage control," *Commununcations of the ACM - Privacy and Security in Highly Dynamic Systems*, vol. 49, no. 9, pp. 39–44, September 2006.

[20] A. Lazouski, F. Martinelli, and P. Mori, "Usage control in computer security: A survey," *Computer Science Review*, vol. 4, no. 2, pp. 81–99, May 2010.

[21] J. Park, X. Zhang, and R. Sandhu, "Attribute mutability in usage control," in *Research Directions in Data and Applications Security XVIII - Proc. of the 18th Annual IFIP WG 11.3 Conference on Data and Applications Security (DBSec'16), Sitges, Catalonia, Spain*, ser. IFIP International Federation for Information Processing (IFIPAICT), vol. 144. Springer Boston MA, July 2004, pp. 15–29.

[22] X. Zhang, M. Nakae, M. J. Covington, and R. Sandhu, "Toward a usage-based security framework for collaborative computing systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 11, no. 1, pp. 3:1–3:36, February 2008.

[23] P. Kumari, A. Pretschner, J. Peschla, and J. Kuhn, "Distributed data usage control for web applications: A social network implementation," in *Proc. of the 1st ACM Conference on Data and Application Security and Privacy (CODASPY'11), San Antonio, Texas, USA*. ACM, February 2011, pp. 85–96.

[24] F. M. Aliaksandr Lazouski, Gaetano Mancini and P. Mori, "Usage control in cloud systems," in *Proc. of the 7th International Conference for Internet Technology and Secured Transactions (ICITST'12), London, United Kingdom*. IEEE, December 2012, pp. 202–207.

[25] A. Lazouski, G. Mancini, F. Martinelli, and P. Mori, "Architecture, workflows, and prototype for stateful data usage control in cloud," in *Proc. of the Security and Privacy Workshops (SPW'14), San Jose, California, USA*. IEEE, May 2014, pp. 23–30.

[26] V. K. Singh and M. S. Kankanhalli, "Adversary aware surveillance systems," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 552–563, September 2009.

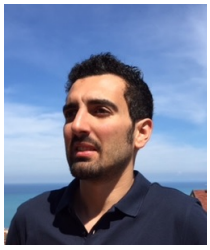
[27] OASIS, "extensible access control markup language (xacml) version 3.0," January 2013, http://www.oasis-open.org/committees/xacml [Online; Accessed on December 10, 2016].

[28] R. Biswas and E. Ort, "The java persistence api - a simpler programming model for entity persistence," May 2006, http://www.oracle.com/technetwork/articles/java/jpa-137156.html [Online; Accessed on December 10, 2016].

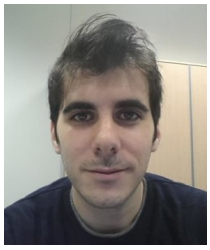
[29] A. C. Yao, "Protocols for secure computations," in *Proc. of the 23rd Annual Symposium on Foundations of Computer Science (SFCS '82)*. IEEE, November 1982, pp. 160–164.

Authors Biography

Enrico Carniani (M.Sc. 2000) IT professional with over twenty years of commercial experience. Graduated at the university of Bologna, recently specialised in Internet Of Things at the University of Pisa. He has a notable background in unix systems administration, network design and security assessment, both from theoretical and pragmatical point of views. Actively involved in software integration and development in several of languages ranging from Assembly to Java and Swift. He is currently focused on software design and architecture for mobile devices.

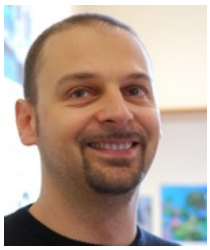


Gianpiero Costantino (M.Sc. 2007, Ph.D. 2011) is a Researcher at the Italian National Research Council (CNR). He has been working for the Security group within the Institute of Informatics and Telematics located in Pisa. From November 2007 to March 2011 he was a Ph.D. student at University of Catania and he conducted his research on Trust, Reputation and Power-Saving within Mobile Ad hoc networks. From 2011 to August 2015 he was a Post-Doc Researcher and his research focused on Security and Privacy aspects within Opportunist Networks, Internet of Things, Social Networking. Currently, Dr. Costantino is involved in the CoCoCloud — FP7 Projects —, HC@WORKS-2 — EIT ICT Labs project —, Securing Smart Airport — Enisa Project —, and his research covers privacy aspects on the Cloud using cryptographic techniques, and Trust solutions for automotive.



Francesco Marino (M.Sc. 2013) received is Master of Science in Computer Engineering from the University of Palermo in 2013. In 2014 he joined the IIT-CNR in Pisa, where he worked in the field of Smart-Cities, especially addressing security and privacy aspects. He is currently a PhD student at Scuola Superiore Sant'Anna in Pisa focusing on automatic negotiation of API in open M2M scenarios, security and privacy in the Internet of Things, Identity Management in the Internet of Things.

Fabio Martinelli (M.Sc. 1995, Ph.D.1999) is a senior researcher at "Istituto di Informatica e Telematica" of "Consiglio Nazionale delle Ricerche" of Italy where he leads the security project. He is co-author of more than two hundreds of papers on international journals and conference/workshop proceedings. His main research interests involve security and privacy in distributed and mobile systems and foundations of security and trust. He chaired the WG on security and trust management of ERCIM, and he is involved in several Steering Committees of international conferences/workshops. He has been/is involved in European and Italian projects on information and communication security.



Paolo Mori (M.Sc. 1998, Ph.D. 2003) is a researcher at "Istituto di Informatica e Telematica" of "Consiglio Nazionale delle Ricerche" of Italy. His main research interests involve trust, security and privacy in distributed systems, focusing on access/Usage Control and trust management in Cloud and for mobile devices. He usually serves in the Organization and Program Committees of international conference/-workshops. He is (co-)author of several papers published on international journals and conference/workshop proceedings. He is usually actively involved in research projects on information and communication security, such as the European Commission funded "Confidential and Compliant Clouds" (CoCoCloud) and the High Impact Initiative "Trusted Data Management with Service Ecosystem" funded by EIT Digital.