Noname manuscript No.
(will be inserted by the editor)

Off-line Enterprise Rights Management leveraging Biometric Key Binding and Secure Hardware

Luigi Catuogno · Clemente Galdi · Daniel Riccio

Received: date / Accepted: date

Abstract In this paper we present a system for Enterprise Rights Management (ERM) for remote maintenance facilities. The Data provider inizializes a mobile device (terminal) by preloading a set of documents, the associated metadata along with the access policy. The envisioned scenario does not allow any further communication, so that documentation confidentiality is achieved by means of a biometric key-binding scheme featuring face recognition. We show that our scheme improves the privacy of operators' biometric templates and the overall system usability. Moreover, we show experimentally that face biometry offers a sufficient level of stability for the purpose of the key recovery. Non-interactive security functionalities and access control enforcement leverage terminals featuring cryptographic hardware. To this end we present an operator device prototype implementation based on Trusted Execution Environments (TEE).

Keywords Enterprise Rights Management · Maintenance Support Systems · Biometric Key Binding · Biometric Authentication · Trusted Execution Environment.

L. Catuogno
Dipartimento di Informatica, Università degli Studi di Salerno, Salerno (SA), Italy.
E-mail: lcatuogno@unisa.it

C. Galdi
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione,
Università degli Studi di Napoli "Federico II", Napoli (NA), Italy.
E-mail: clemente.galdi@unina.it

D. Riccio
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione,
Università degli Studi di Napoli "Federico II", Napoli (NA), Italy.
E-mail: daniel.riccio@unina.it

1 Introduction

Effective and flexible systems that govern the flow of sensitive information within the corporate boundaries are a crucial need for any kind of organization. Indeed, besides the protection of "internal data" (*e.g.,* notes, emails, executive communication), major issues are raised in enforcing access control over information which are released to different stakeholders for the sake of institutional or business processes. Moreover, private data can flow through different administrative and technological domains, compromising the organizational control capabilities.

Enterprise Rights Management (ERM) embodies methodologies and technologies which pursue the preservation of corporate data confidentiality independently from where they are stored, transferred or used. ERM Systems define corporate security policies at data creation time and enforce them at data fruition time. To this aim, in an ERM system, each restricted piece of information (*e.g.*, a document) is encrypted and linked to security "metadata" which carries its access control policy, encryption keys, etc. The document is intended to be distributed within and/or beyond the corporate boundaries. At every access request to the document, the system authenticates the requester, verifies whether the request is compliant with the access control policy stated for the document and eventually releases the decryption keys. A crucial issue here is *where* the metadata is stored. In case metadata is stored by the data owner, an access request can be served only if the data owner is available on-line. Otherwise, if metadata are attached to the document, a trustworthy mechanism which securely releases (and withdraws) the keys on behalf of the data owner must be available at the fruition place.

Access policy flexibility is an important feature of ERM systems. Specifically, security decisions could depend on *environmental* factors, such as *from/for where* the request is issued. To take in account these factors, ERM architectures have to leverage local trusted security agents or secure hardware which are in charge of gathering environmental information (*e.g.*, the requester's position through the GPS) to be attached to the access request, and trustworthily enforce the policy decision, preventing misbehaving by the local operator.

Secure operators' identification becomes crucial, especially in the case in which the data has to be accessed by personnel that does not belong to the data owner institution. In this case biometric identification systems provide an excellent tool as they are widely accepted by users and they are able to guarantee high levels of security without sacrificing ergonomy.

This paper addresses the design and implementation of an ERM Systems which guarantees data and communication confidentiality and is specifically tailored for remote maintenance facilities.

Manufacturers of large and complex machinery, such as industrial equipments and special vehicles, generally sell their products along with several maintenance and support facilities that can be intended *as part of the products themselves*. Frequently, the implementation of such services involves an in-house pools of experts, providing off site technical support, and on site in-

tervention teams composed by customer's operators. Often, documents transferred and communication occurring between the pool of experts and the technicians operating on-site (maintainers), carry sensitive information regarding the product and the customer. Exposing such information to unauthorized parties raises serious security issues such as the disclosure of manufacturer's industrial secrets or the infringement of any secrecy policy the customer has to enforce (*e.g.*, the customer is a government or military institution). In such cases, technical support facilities should feature mandatory security functionalities including data encryption, access control, users authentication and session management.

In this paper, we consider the following scenario: the manufacturer, who is the owner of the technical documentation for its products, provides the maintainers with a number of devices that are used exclusively for the purposes of the maintenance operations. In order to minimize the amount of (sensitive) technical documentation delivered through an untrusted network, the manufacturer/data provider preloads the devices with the documents that are likely to be required for a specific maintenance operation along with the corresponding security metadata including the appropriate access control policy. On-site access control enforcement possible includes the evaluation of environmental conditions and parameters such as geographical position, events and alerts. As stated above, due to the unavailability of any communication with the data owner, both security policy enforcement and local paramenter evaluation are up to a trusted agent featured by the mantainers' devices.

Our Contribution. Contributions of this paper are summarized in the following points:

- We present a variation of an existing biometric key binding scheme (Riccio et al 2016) which is used to secure the disclosure of documents encryption keys stored on a personal cryptographic device. The variation to the original scheme is three-fold: (a) our scheme is interactive, in order to properly account the presence of multiple devices; (b) privacy of user credentials is improved as, in our scheme, they never leave the user's personal device; (c) in order to improve the overall system ergonomics, we use face biometries instead of iris ones and show, for the first time, that face biometry provides a sufficient level of stability for the purpose key reconstruction.
- We present a secure system that allows the enforcement of strict access policies over the data that are encrypted and stored on the maintainers' device. The data provider sets-up the device during an initialization phase by preloading all the documents needed for a specific session, each encrypted under a different random key, along with a context-aware access policy. After the initialization phase, the access control decisions are taken by the device *without* interacting with the data provider.
- We present a specialized system architecture which features a local trustworthy security agent for the sake of policy decision and enforcement. To validate our proposal, we show how it naturally fits on a Trusted Execution Environment (TEE) standard platform.

2 Related works

2.1 Enterprise Rights Management Systems

ERM technology fits the interests that industrial players have in preventing leakage of private and strategical information while used in their internal processes and communication (Abbadi and Alawneh 2008; Maniatis et al 2011; Grimm and Anderl 2013). Protecting corporate secrets has motivated considerable research efforts. Here we recall three main approaches: *(i) application-level* ERM solutions such as (Adobe Systems 2013; EMC Corporation 2003; Microsoft Corporation 2016; Catuogno et al 2016), in which security is up to the applications used to create and handle sensitive documents; *(ii) storage-level* solutions ensure confidentiality and access control over arbitrarily formatted data by instrumenting the storage facilities with transparent encryption and flexible key management mechanisms (Park et al 2015; Blasco et al 2015; Castiglione et al 2014; Catuogno et al 2014); finally *(iii) infrastructure-level ERMs* leverage multi-level network infrastructures in which a two-fold access control policy regulates, on one hand whether a platform is enabled to join the corporate network and on the other hand whether any connected platform is authorized to access the data it requests (Gasmi et al 2008; Catuogno et al 2009).

In (Catuogno et al 2016) the authors present an on-line ERM system that is specifically tailored for supporting maintenance operations in remote facilities. Such a system allows the enforcement of dynamic context-based access policies in which the data owner can decide, for each request, whether or not to grant access to the requested data. The solution presented assumes the continuous online availability of the data provider.

The differences between our solution and the one in (Catuogno et al 2016) are the following. (a) we present an off-line solution that restricts the interaction between the data provider and the device only to an initialization phase, in place of a continuous interaction; (b) in our work the device key is hidden by using a specific biometric key binding scheme presented in (Riccio et al 2016) while in the previous solution the device key is exchanged for each session between the (online) data provider and the device; (c) in order to preserve user privacy, the user biometric templates are stored on a user-trusted device and we present an interactive privacy-preserving variant of the key binding scheme in place of using cancelable multi-biometric scheme.

2.2 Access control policies

The literature on access control is huge. The RBAC model has been introduced in the seminal paper (Sandhu et al 1996) and it is a *de facto* standard. In recent years the focus on extensions of the RBAC models specifically targeting context aware access control policies for mobile devices has received a considerable amount of attention in the research community (Li et al 2015;

Kirkpatrick and Bertino 2010; Gupta et al 2014; Bonatti et al 2015), just to name a few. Since in our system the access control is executed by a mobile device, one of the key factors to be considered is the time needed to provide answers to access control queries. Our system is flexible enough to be completely *oblivious* w.r.t. to access control policy specifications.

2.3 User authentication

User authentication is a long standing problem in computer security and different solutions have been proposed during the last decades.

Although subject to known weaknesses, password-based authentication schemes are way far the most deployed systems. Amongst several attempts to enhance security and usability of old-fashioned text-based password schemes (Haller 1994; McDonald et al 1995; Blundo et al 2004; Ciaramella et al 2006), new "human affordable" authentication protocols (Suo et al 2005; Matsumoto 1996; Blonder 1996; Hopper and Blum 2001; Catuogno and Galdi 2014a,b, 2010) have been studied since the early 90s. However, such authentication schemes might not be usable in some operational contexts as they require an active interaction of the user with the authenticating device.

In such contexts a valid alternative to user authentication is the use of biometric systems. Biometric verification systems aim at verifying the compatibility of a set of biometric measurements against the ones that are present in a given template. Such systems can improve the security of "traditional" authentication protocols as biometric traits contain reliable identity information about the subject they belong to.

Single-biometric systems use a single biometric trait for the verification procedure. Since each feature has its own characteristics, this type of systems are subject to attacks related to the specific trait in use. Furthermore, they inherit the identification limits that are proper of the used trait. There exist two possible ways to overcome such limitations: the use of multiple biometries or the use of multi-factor authentication schemes (Xu et al 2018; Jiang et al 2017; Abate et al 2011).

Amongst many physiological traits, face recognition (Sirovich and Kirby 1987; Turk and Pentland 1991; Zhao et al 2003; Cai et al 2006) is probably the most attractive for a broad range of applications, being well accepted by users, very easy to acquire. Moreover, face recognition is able to provide high levels of security, provided that the quality of face images is not excessively stressed by changing in acquisition conditions and pose/illumination distortions. Hence, face biometric has the potential for being a good user identification mean in ERM systems.

Theft of user identity still represents a potential issue, since biometric data are unique and distinctive, and hence they cannot be changed if they are compromised. In order to increase the robustness of biometric systems to the user identity theft, cancelable biometric features have been proposed (Rathgeb and Uhl 2011). Cancelable features are derived from the original biometric data

by applying a non reversible transformation, which preserves the topological properties of the original feature space.

2.4 Biometric Key Binding schemes

A Biometric-Key Binding (BKB) scheme, is a methodology that, informally, allows to bind the recovery a random cryptographic key to a successful user authentication by means of some biometric scheme. In 2002, Juels and Sudan (Juels and Sudan 2002) introduced the fuzzy vault scheme, which is the simplest and most studied BKB at present. In such a scheme, the set of genuine biometric features is augmented by adding some randomly generated chaff points to construct a vault V. The vault V is then used to lock a cryptographic key. Most of the fuzzy vault based BKBs in literature construct the vault by exploiting fingerprint minutiae, as they are in limited number and provide very discriminant information. However, the major limitation of these methods is that the cryptographic key is correctly reconstructed only if the positions of minutiae is almost identical in the fingerprints used to perform locking and unlocking operations that means they must be pre-aligned prior to the fuzzy vault construction. Moreover, these approaches are difficult to extend to biometric traits different from fingerprints, as they just rely on the information provided by the minutiae positions, so being unable to completely exploit texture information. The secure online authentication protocol proposed by (Wu and Yuan 2010) is among the few examples of fuzzy vault BKBs based on face templates that have been recently presented in literature. However, experiments conducted in (Wu and Yuan 2010) show that this approach is not able to prevent an impostor to recover the cryptographic key, as the False Acceptance Rate (FAR) is non zero. Notice that zero-FAR is a mandatory requirement for high-security applications like ERM systems.

3 Motivating Scenario

Companies that produce complex objects provide their clients with technical documentation needed for remote maintenance operations. Such documentation is considered to be sensitive as it might reveal important insights on the design of the specific component that is being fixed, e.g., weapons onboard warships.

In order to ensure trustworthiness, the company (a) provides mobile support devices to be used *exclusively* for the purposes of remote maintenance and (b) ensures that documentation is never stored in clear on such devices. Moreover, a fine-grained access control policy should be established on each part of the documentation, in order to allow the access to any specific part only when it is necessary.

It is assumed that support devices may be used by different technicians. To this end: (a) it is required that every potentially involved technician is

registered (once) in the company database and is identified before the start of each maintenance operation; (b) technicians' credential should be stored on personal tamper-resistant device.

In the above scenario we can identify three different agents:

- Data Provider (DP): the data provider can be seen as the data *owner*. The DP should be able to enforce arbitrary access control policies for its own data.
- Company Device (CD): the CD is the device, provided by the DP, that is used exclusively for the purposes of the remote maintenance. It is used to locally store encrypted data and to enforce DP-defined access control policies.
- User Device (UD): to ensure user privacy, the user stores her biometric and digital credentials on the UD. This device should be used exclusively by a single user, e.g., a personal mobile phone.

We require that the DP can communicate with the CD and with the UD only during an initialization phase. Conversely, during the actual operative use, the CD can comunicate only with the UD and they are both isolated from the DP. For the sake of simplicity and technological neutrality, the restricted contents are assumed to be organized as collections of *data units*. Each data unit may contain/consist of different data types, e.g., text, images, videos, etc. Furthermore, each data unit may point to other data units, making possible an hypertextual-like navigation.

4 Threat Model and Requirements

In the above referenced scenario, we consider the following:

4.1 Threat model

We assume that the DP is trusted and performs all the required operation correctly.

We consider attacks from so called insiders, i.e., authorized users that try to overcome the limitations imposed by the access policy. In other words, the attackers try to access information they are not authorized to access in the *current context*, e.g., accessing file X in location Y when the access policy allows the access to X only in location Z.

An adversary has, thus, access to the CD and to the UD. Specifically, the adversary has full access to unprotected storage components and to unprotected volatile memory in the CD and UD. The goal of the attack is to get access to the information stored on the CD in a given context without having the required authorization.

This type of adversary can (a) legitimately authenticate using their own biometric credentials and measurements and (b) can monitor and store the content of unprotected memory and storage components of CD and UD.

Notice that, once the content of some data unit has been released by the system, we assume that the information is public. Nevertheless, the system should forbid the attacker to circumvent the access policy of previously-released data units. This translates in the impossibility for the attacker to obtain the encryption keys used by the system.

4.2 Authentication requirements

Key ingredient in the security of the system is the possibility of securely identifying all the agents. Depending on the specific application scenario and on the sensitivity of the information being requested by the user, the data provider might deploy different authentication mechanisms. In this paper, we assume the following:

Device authentication. Each device is authenticated by means of a standard X.509 certificate that is written to the device by the data provider. It might be assumed that the CD is used exclusively for the purposes of the maintenance system.

User authentication. Given the presence of cameras on currently available mobile devices, we propose for this system the identification through face authentication. Moreover, the BKB scheme presented in (Riccio et al 2016) has been re-adapted to work in an interactive and privacy preserving fashion with face templates instead of iris bitstreams. The biometric template is stored on the UD and it is used in a privacy-preserving protocol to authenticate the operator, whose biometries are measured by the CD.

4.3 Trustworthy local key management

In our architecture, the CD is required to feature a local "trusted agent" guaranteeing that local policy enforcement and keys/credentials management are trustworthily accomplished. This is to prevent unauthorized data/keys access to (a) adversaries having stolen the CD and to (b) insiders having access to CD and UD, leveraging malicious or tampered code in order to circumvent the security policy or to obtain the access to sensitive information and keys. In particular, in our system, data confidentiality is based on the assumption that the local key management is either implemented on top of hardware security support that allows the secure storage of keys and the secure execution of code or it entails the introduction of a secure storage area that is isolated from the application execution environment (e.g., GlobalPlatforms Trusted Execution Environments [29]). In Section 7 we will describe in details the architecture underlying a possible implementation of our proposed system.

5 A Biometric Key Binding Scheme

Our first result is a variant of a Biometric Key Binding (BKB) scheme, appeared in (Riccio et al 2016). The scheme we present here differs from the one in (Riccio et al 2016) for three main reasons. First, the new scheme is interactive. The original scheme is supposed to concentrate on a single entity the user certificate, the biometric measurements and the encoded key while, in our case, these information are spread on two different devices. Second, the new scheme is privacy-preserving. The user certificate never leaves the UD and the encoded key is only available to the CD that is supposed to use it. Third, we use face biometry in place of iris ones. On the one hand, this modification improves system usability. On the other hand, it is widely known that iris biometry is more stable than the face ones. Since stability has an effect on the correctness of the key reconstruction algorithm, an experimental validation of the correctness of the key binding scheme under the new biometric features is required. Despite higher instability, we experimentally show in Section 5.3 that face biometries can be effectively used for the purpose of key recostruction.

5.1 The BKB scheme in (Riccio et al 2016)

Informally, the scheme presented in (Riccio et al 2016), starting from a biometric template B_u for a user u, generates a *biometric authorization function* f_{B_u}, that is used to encode a cryptographic key K by computing an *authorization token* $w(u) = K/f_{B_u}(B_u)$. The security of such an encoding is guaranteed against brute force attacks. At a later time, a user u who (a) holds a specification of the authorization function f_{B_u}, (b) holds the authorization token $w(u)$ and (c) whose biometric measurements B'_u are compatible with B_u can recover the hidden key K by using $f_{B_u}(B'_u)$.

In more details, the scheme presented in (Riccio et al 2016) consists of two algorithms, called BindBioCryptoKey and ReconstructCryptoKey, respectively. The first one takes as inputs a biometric template $B_u = (b_1, \ldots, b_q)$ for user u, a cryptographic key $K = (k_1, \ldots, k_t)$ and a distance parameter δ that, informally, defines the maximum tolerated distance between two compatible biometric measurements B_u and B'_u. BindBioCryptoKey combines the biometric key B_u with a random vector and it obtains the specification $f_{B_u} = (h_1, \ldots, h_q)$ of q random oscillating functions[1]. The function defined by f_{B_u} has the following properties: it is not possible to derive any information on the biometric key B_u. Furthermore, $f_{B_u}(B_u)$ is a local minimum for f_{B_u}. Given f_{B_u}, the algorithm deterministically derives a vector $(Y_1, \ldots, Y_t)$. At this point, it computes the authorization token $w(u) = (w_1, \ldots, w_t)$ for user u where, for each j, $w_j = k_j/Y_j$. The BindBioCryptoKey algorithm outputs the *helper data* $HD = (f_{B_u}, w(u)) = ((h_1, \ldots, h_q), (w_1, \ldots, w_t))$.

[1] The scheme generates q oscillating functions, one for each component b_i, $i = 1, \ldots, q$ of the biometric template B_u. Each function is used independently from the others.

The ReconstructCryptoKey algorithm takes as inputs a vector of biometric measurements B'_u, the helper data $HD = ((h_1, \ldots, h_q), (w_1, \ldots, w_t))$ and the value of δ. It derives the multipliers Y'_j by using $(h_1, \ldots, h_q)$ and B'_u and outputs $k_j = w_i Y'_i$. We refer the reader to (Riccio et al 2016) for a detailed description of the scheme.

5.2 A new interactive privacy-preserving BKB scheme

The scheme presented in (Riccio et al 2016) entails the property that helper data contains both the specification of the oscillating functions and the encoding of the key K. Unfortunately, in our model, these two information should be set apart. The oscillating functions specification depends on the user biometry and, thus, should be stored on the UD. Conversely, the authorization token that encodes the key are inherently bound to the CD that stores the encrypted information and should be therein stored.

Furthermore, it is not possible to apply the scheme in (Riccio et al 2016) as is. In the original scheme, the key reconstruction procedure is done *in clear*, in the sense that, given the helper data $HD = (f_{B_u}, w(u))$ and the biometric measurements B'_u, the algorithm returns the key k extracted from $w(u)$. Thus, in order to reconstruct the key on the CD, we need to transfer the user biometric certificate, containing the oscillating functions specification f_{B_u} from the UD to the CD. On the other hand, if the reconstruction algorithm is executed by the UD, the key K will be reconstructed in clear on this device. In each case, one of the device obtains some information that it is not supposed to acquire. We need to slightly modify the scheme in (Riccio et al 2016) to fit our new model.

We first observe that, if $HD = ((h_1, \ldots, h_q), (w_1, \ldots, w_t))$ encodes the key $K = (k_1, \ldots, k_t)$, then, for every $\lambda \neq 0$, $HD' = ((h_1, \ldots, h_q), (\lambda w_1, \ldots, \lambda w_t))$ encodes the key $K' = \lambda K = (\lambda k_1, \ldots, \lambda k_t)$.

The new BKB scheme we propose in our system is as follows. The generation of the helper data $HD = ((h_1, \ldots, h_q), (w_1, \ldots, w_t))$ is executed exactly like the one in (Riccio et al 2016). In our system, the oscillating function specification $(h_1, \ldots, h_q)$, that depend on the user biometry, is stored on the UD, while the key encoding $w(u) = (w_1, \ldots, w_t)$ is stored by the CD.

Whenever a user u needs to be identified, the user provides her identity u and her biometric measurements B'_u to the CD. The CD recovers the authorization token $w(u) = (w_1, \ldots, w_t)$ from its local storage, generates a random λ, computes $\lambda w(u) = (\lambda w_1, \ldots, \lambda w_t)$ and sends B'_u and $\lambda w(u)$ to the UD over a secure channel. The UD recovers $w(u)$ from its local storage, runs the algorithm ReconstructCryptoKey and, as observed above, recovers $\tilde{K} = \lambda K$ that is transferred back to the CD. Finally, the CD computes $K = \tilde{K}/\lambda$. Notice that, the above procedure implicitly identifies the user since biometric measurements $B'_{u'}$ that are not compatible with B_u would produce a key K' that cannot be used to decrypt the data on the device.

We further observe that, although the UD computes λK, it cannot obtain any information on K without colluding with the CD.

5.3 Experimental Validation

The aim of this section is to present the results of the different tests performed on the face based BKB scheme adopted in the proposed framework. The BKB scheme has been tested on face templates that were extracted from images included in the AR-Faces dataset (Martinez 1998). The dataset includes 126 subjects, who have been acquired twice in two different sessions separated by a time lapse. For both sessions, face images are organized in subsets according to difference in expression, illumination and presence/absence of occlusions (from 1 to 13 in the first session, and 14 to 26 in the second session). In all the experiments, cryptographic keys have been simulated by generating random sequences of 256 bits, while face templates are generated from images provided by the AR-Faces dataset, by applying the following pipeline: i) face detection, ii) face normalization, iii) feature extraction, and iv) vector dimensionality reduction.

The face and its characteristic points are located through the approach in (Milborrow and Nicolls 2008). The algorithm locates 68 interest points, which are inputted to a piecewise linear mapping functions (Goshtasby 1988) as control points to perform local registration with a reference frontal model. After have been cropped, the face region is normalized with respect to illumination changes by means of an adaptive local histogram equalization process and vectorized by stacking rows. The so obtained feature vector is normalized by applying the Median/MAD technique (Jain et al 2005). As regarding the dimensionality reduction process, the Orthogonal Laplacianfaces Projection (OLPP) technique has been applied to project face samples into the final feature vector space. Since the projecting matrix has to be learned from training data, the whole AR-Faces dataset has been split in two parts. The former consists in 122 neutral face images of 61 subjects that have been used for training, while the latter includes 130 neutral face images of the remaining 65 subjects that have been exploited for testing. The biometric face templates B_u obtained after applying the OLPP consist of 95 real values, which represent the face bio-items b_i, $i = 1, \ldots, 95$.

In our experiments, we tested the accuracy of the binding process in separating genuine and impostor requests in terms of standard indices, such as Genuine Acceptance Rate (GAR), False Acceptance Rate (FAR), Genuine Rejection Rate (GRR) and False Rejection Rate (FRR). In particular, GAR represents the percentage of cases where the cryptographic key K is correctly reconstructed by submitting the biometric key of a genuine user, while FAR measures the probability that biometric key of an impostor allows to reconstruct K.

In the first experiment, we estimated a proper value for the tolerance δ, when dealing with face biometric templates. An all vs. all comparisons on

the 130 templates extracted from the testing set is performed according to the Chebyshev distance with the aim of deriving the value of δ, which better separates the genuine and impostor distributions. The sample distributions are shown in Figure 1, where genuine are represented by a solid line and the impostors with a dotted one. The parameter δ must be set to the highest distance value, for which none of the impostors would be accepted. In this case, we would set $\delta = 155$.

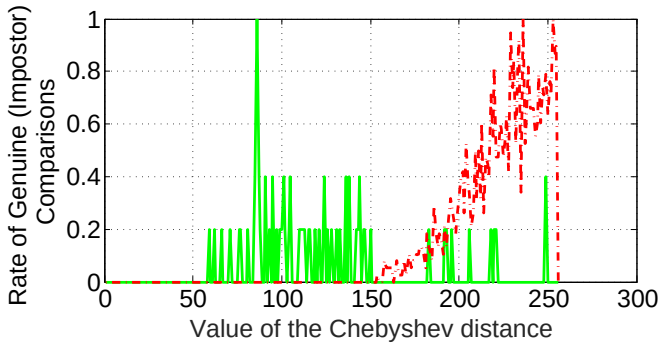

Fig. 1 The two sample distributions: i) genuine (solid line) and ii) impostors (dotted line).

In the second experiment, we investigated the performance obtained by the binding process for a cryptographic keys K of 256 bits length, when the value of δ is fixed to 154. The BKB provides a GAR of 0.836 with FAR=0.0. Curves reported in Figure 1 show that genuine and impostor distributions are sharply separated except for few face samples that produce mutual distance significantly higher than the value selected for δ. This further underlines the worthwhile contribution provided by OLPP to reduce the intra-class variability of face samples.

The third experiment is aimed at stressing the system with respect to the size and also to the variability of the biometric key. In other words, a cryptographic key K of 256 bits length was considered. For each of the 65 subjects in the testing set, K is encoded by using a portion of the first biometric key B_{u1} consisting of the first $q' \leq q$ bio-items. The cryptographic key K was decoded by using a portion of the second biometric key B_{u2}, where each bio-item b_{i2} is added with random noise $\eta_i \in [0, max_\eta]$. This experiment was repeated for four different values of q' (i.e. 16, 32, 64, and 95), eleven values of max_η (i.e. max_η ranging from 0 to 50 by a step of 5), and eleven values of δ (i.e. δ ranging from 125 to 175 by a step of 5). When the value of q' is fixed, the rate of correct key bindings (i.e. coding/decoding operations) can be represented by a surface. Thus, Figure 2 reports four different surfaces, one for each of the values considered for q'.

From surfaces shown in Figure 2, it comes out that under optimal conditions (i.e. without noise and with δ set to the maximum value for which

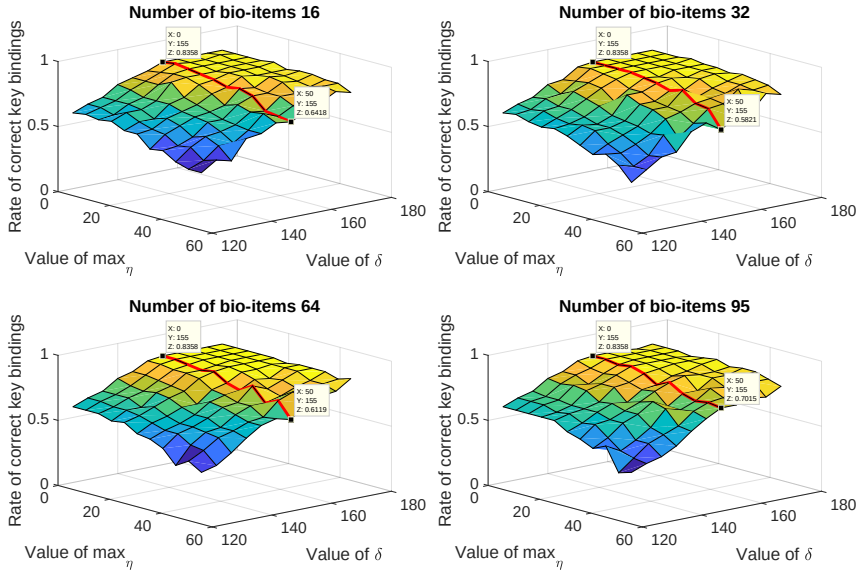

Fig. 2 The rate of correct key bindings for different sizes of the biometric key q'. The solid line in red corresponds to points on the surface for which $\delta = 155$.

the FAR is zero), the key binding scheme is able to obtain the highest rate of correct key bindings, even with a small portion of the biometric key ($q' = 16$). However, such a small number of bio-items would make the system more vulnerable to brute force attacks. As regards the parameter δ, it is observed that the rate of correct key bindings grows up as the value of this parameter increases. However, the points on the right-hand side of the red solid line correspond to values of the parameter δ for which the system performs False Acceptances, so becoming vulnerable to impersonation attacks where an impostor could reconstruct the cryptographic key encoded by a genuine user. Furthermore, although the rate of correct key bindings decreases for high values of max_η, it is observed that the system is robust to the noise in the biometric key and that this strength generally increases with the length of the biometric key.

6 System Operation

As in (Catuogno et al 2016), we identify three possible phases, a Registration, an Enrolment and an Execution one. In the first one, the data provider registers and initializes the devices that will be used during the executions of the protocol and the users that will have access to the documentation. This phase is run once for each user/device. The Enrolment phase is used to prepare the CDs that will be used in a specific maintenance operation. This is the only phase during which the DP can communicate with the CDs. The last phase

is the Execution one in which authorized users access the data stored on the CD.

6.1 The Registration phase

The data owner needs to register both the users and the devices she is going to operate with. At the end of the Registration phase, each CD has been initialised with its own certificate and each authorised user has been registered in the DP database.

User registration. During the user registration phase, the data owner identifies each user by means of her *user certificate*. The DP, using the biometric measurements B_u for user u, generates the biometric authorization function $f_{B_u} = (h_1, \ldots, h_q)$ for user u. This function is transferred and *securely* stored by the UD. Notice that this phase preserves user privacy since the UD does *not* store any information about the biometric measurements of the user. At the same time, the DP sends her certificate to the user. Furthermore, since the authorization functions are random, as observed in (Riccio et al 2016) they are inherently cancelable. Clearly, this phase might be avoided by using online certificate repositories. On the other hand, we explicitly require it as it might be useful in the case in which the DP would require to provide each user with a DP-generated certificate.

Device registration. This phase is mandatory for each CD. During this phase the DP pre-loads the device with its own *device certificate*. Such a certificate will be used to create secure point-to-point communication between the CD and the DP.

6.2 The Enrolment phase

During this phase the DP communicates with the CDs in order to preload all the information that will be needed for their actual use on the field. Clearly, in a complete off-line solution, i.e., in the case in which *no* communication is possible between DP and CD *after* the device has been sent to the client, this phase has to be fused with the previous one. We leave the enrolment phase distinct from the registration one in order to allow a more flexible solution that allows a partial update of the devices, if needed.

Notice that this phase might be executed while the device is on the field. For this reason, the first step in the enrolment phase is the creation of a mutually authenticated secure channel between the DP and the CD by using the device certificate stored by the DP and the DP certificate stored by the CD during the registration phase.

The underlying idea is the following. Each data unit is encrypted using a different random key, referred to as the *file key*. Such keys are encrypted

using a random *device key*. Preventing the access to the device key is sufficient to logically forbid the access to the file keys that, in turn, will prevent the access to data. Finally, the device key is encoded using the scheme presented in Section 5. In order to withstand spoofing attacks, we modify the latter scheme so to require multi-factor authentication, i.e., the user authentication depends on the possession of the UD *and* on her biometric authentication.

The following procedure is executed for each CD independently. The DP identifies (a) the set of data units $D = \{d_1, \ldots, d_\ell\}$ that will be used for a specific operation (b) the set of users $U = \{u_1, \ldots, u_n\}$ that might have access to the data units in D and (c) the access policy A that will be enforced by the CD.

The DP first generates a random device key K. Each data unit $d_i \in D$ is encrypted using a random file key k_{d_i}. Each encryption key k_{d_i} is encrypted with K. For each user $u \in U$, the CD recovers the user authentication function f_{B_u}, and encodes K using the scheme presented in Section 5 by computing $w(u) = K/f_{B_u}(B_u)$.

The DP transfers to the CD the following:

- An authentication token for each user in U, i.e., $w(u_1), \ldots, w(u_n)$
- The encrypted data units $E(d_1, k_{d_1}), \ldots, E(d_\ell, k_{d_\ell})$
- The set of file keys encrypted under the device key, $E(k_{d_1}, K), \ldots, E(k_{d_\ell}, K)$
- The access policy A

Since the mentioned information is transferred over a secure mutually authenticated channel, we can assume that it reaches the CD unaltered. Given the above information, the only way to access the encrypted data is to extract the device key by using the authorization tokens, use it to decrypt data units keys and, finally, decrypt data units.

Notice that the DP sends to the CDs the authorization tokens but it does *not* transfer the authorization functions. The reason for this choice is that since the CD does not hold the authorization function, it cannot recover the device key k by itself but it needs to interact with the UD.

The above procedure implicitly assumes that none of the required data units is stored on the device. In a more dynamic situation, some of the data units required for the current mission might have been loaded on the device for a previous operation. In this case, the device already stores some data units, each encrypted under a random key. The DP might avoid the retransmission of such units by retransmitting the random file key encrypted under the new device key. This strategy clearly minimizes the communication *overhead* since it is sufficient to transfer a single file key to enable the access to an entire data unit. Clearly, this operation can be useful if devices are frequently reinitialized or in case of limited bandwidth. On the one hand this architecture creates a slight overhead for the DP as she needs to keep track of the state of each device. On the other hand this infrastructure has many advantages. Indeed it allows the DP to autonomously create the list of data units that are currently stored on every device. Furthermore, given the information stored by the DP, it is possible to enforce remote file removal policies by pushing random keys to

the CD. Finally, and probably most importantly, it allows the DP to enforce dynamic and fine-grained access control policies at each enrolment phase since access to files are granted only after an explicit authorisation of the DP.

6.3 Execution Phase

Once the registration and the enrolment phases are completed, the CD holds almost all the information needed to execute the maintenance operation. The execution phase is assumed to be done offline, i.e., it is *not* possible to contact the DP. On the other hand, since CD and UD need to communicate, we assume the creation of a mutually authenticated channel between these two devices, e.g., by means of the Secure Element API provided by TEE specifications.

Whenever a new maintenance session is started, the system needs to authorize the operator. At this point, the identification procedure defined in Section 5.2 is executed. That is, the user provides her identify u and her biometric measurements to the CD that, by interacting with the UD, computes the device key K.

Once the device key is known, the device is ready to answer the user requests. Whenever the operator requires access to a specific data unit d_i, if the access policy allows it, the system decrypts the corresponding file key k_{d_i} using K, decrypt the d_i using k_{d_i}, and provides the data di the user.

The proposed scheme works using the underlying assumption that session and file keys are not accessible to the adversary. Indeed, if the adversary can collect such keys, she can use them at a later time and bypass the access control system. For this reason, the keys cannot be stored into the unprotected storage area or volatile memory on the CD. On the other hand, since no communication is possible with the DP, the only way to enforce such assumption is to rely on some underlying hardware security support, as we will detail in Section 7.

7 A TEE-based Secure Architecture

The system proposed in the previous sections provides data units' security and reliability based on the underlying assumptions that an adversary (a) cannot alter or circumvent the access policy cannot and (b) cannot read the device/file keys.

In order to provide the above guarantees, our system implementation leverages the *Trusted Execution Environment (TEE)* technology (GlobalPlatform 2011). TEE-enabled devices feature two distinct execution environments characterized by different security properties: the Rich Execution Environment and the Trusted Execution Environment. The former is devoted to the execution of legacy OSes and applications (so called RichOS and Rich Apps or RA) and provides a "normal" security level; the latter runs a Trusted Operating System (TrustOS) which, in turn, takes care of guaranteeing the integrity and the execution of Trusted Applications (TA) as isolated workloads. Trusted

Applications are intended as "secure service providers" for Rich Applications. Communication between such worlds only takes place by means of a strictly defined protocol implemented, on both sides, by a proper set of APIs. Rich Applications invoke secure services through the TEE Client API, whereas Trusted Applications provide their services through an interface implemented using the TEE Internal API. Figure 3 shows the details of the system's architecture.

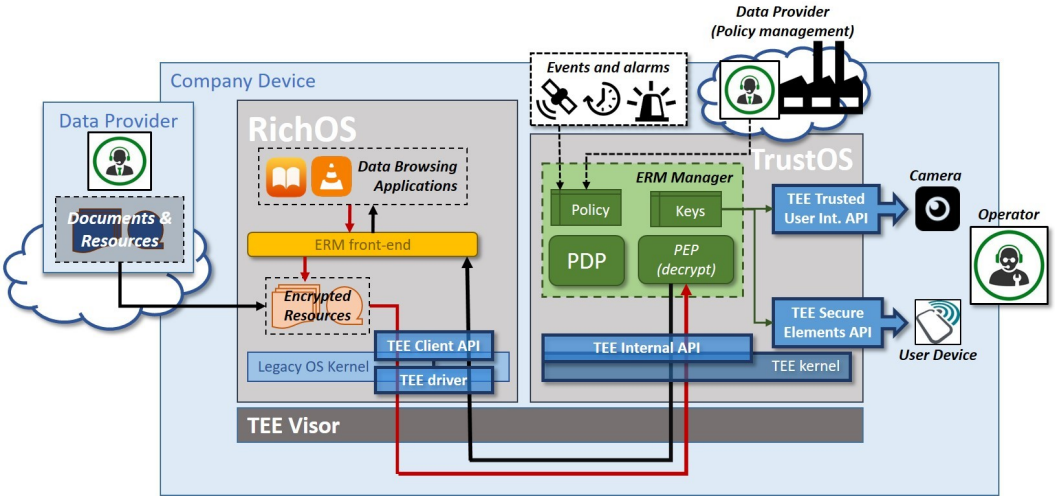

Fig. 3 System Architecture.

User authentication is accomplished by collecting biometric measurement by means of the CD's camera and matching it with the biometric templates securely stored on the UDThe UD is a *secure element* (*e.g.*, an NFC tag, a smart card or a secure SD card). The operator's biometric templates are stored on the UD at *user registration* time. It is assumed that this operation is done in a secure environment. During the authentication, the *ERM Manager* communicates with both these devices through a *trusted path*. In particular, the interaction with the UD takes place through the TEE Security Elements API.

The ERM Manager is the core component of our architecture. It is in charge of user authentication, it acts as Policy Decision Point (PDP) and Policy Enforcement Point (PEP). To this end, it features a private area which stores policy specifications, credentials and keying material, for the sake of interaction amongst every system component. The ERM manager is implemented as a *Trusted Application* and the secure storage is cryptographically bound to such component by the Trusted Operating System. Policies, keys and credentials are transferred to the ERM manager both at device registration and at enrolment time. In both cases, the delivery takes place through a trusted channel established between the DP and the CD .

Policy decisions are taken according to the policy specifications (along with any relevant environmental parameter such as the CD geographical location, alarms) and are enforced by denying to process the data the operator sends for decryption.

Policy enforcement is achieved by means of data encryption. Protected data to be accessed are sent to the ERM manager which: (a) checks if the issued request is compliant to the policy, (b) retrieves the proper key and (c) sends back the decrypted data unit to the requestor. Access requests, coming from any legacy Rich Application, are forwarded to the ERM Manager by the *ERM front-end* which is its counterpart running in the Rich Operating System.

The ERM front-end. Operators use legacy applications such as HTML browsers and multimedia players in the RichOS to access encrypted documentation resources. Such *Document Browsing Applications (DBA)* access data through the *ERM Front-end* service. Such a component plays as a proxy for standard protocols such as HTTP and RTSP used by DBAs. This component, does not have any role neither into the security policy enforcement nor into data encryption/decryption, as it simply forwards the requests by DBAs to the ERM Manager throughout the TEE infrastructure and delivers its replies to the proper requester.

8 Conclusions

In this paper we have described a solution to the problem of securely and reliably storing confidential documentation over a mobile device, even in hostile environments where communication can be unreliable and no connection to the data provider is possible. Our ERM system encrypts the technical documentation and pre-loads it on the mobile device. The featured documentation is composed of self-containing data-units. A fine grained access control policy is dynamically enforced every single time the operator accesses any data unit, so that, the data owner is enabled to govern the disclosure of information according to the operator identity and role, as well as the documentation usage patterns and the circumstances. In order to make up the impossibility of contacting the data provider, the system relies on a novel biometric key binding scheme and on the usage of the security primitives provided by the hardware devices. As for the former, we have presented an interactive variant of a known scheme that allows the preservation of the operators' privacy. At the same time, we have modified the scheme so that it can use face biometries and we have experimentally validated its correctness under the new (less stable) biometry. Regarding the latter, we have presented a possible TEE-based architecture that leverages the security services provided by the current hardware mobile device.

References

Abate AF, De Marsico M, Riccio D, Tortora G (2011) MUBAI: multiagent biometrics for ambient intelligence. J Ambient Intelligence and Humanized Computing 2(2):81–89, DOI 10.1007/s12652-010-0030-2, URL https://doi.org/10.1007/s12652-010-0030-2

Abbadi IM, Alawneh M (2008) Preventing insider information leakage for enterprises. In: Second Intl. Conf. on Emerging Security Information, Systems and Technologies (SECURWARE'08)., IEEE, pp 99–106

Adobe Systems (2013) Adobe livecycle es4. http://www.adobe.com/products/livecycle.html

Blasco J, Tapiador JE, Peris-Lopez P, Suarez-Tangil G (2015) Hindering data theft with encrypted data trees. Journal of Systems and Software 101:147–158

Blonder GE (1996) Graphical passwords. Lucent Technologies Inc, Murray Hill, NJ (US), US Patent no. 5559961

Blundo C, D'Arco P, Santis AD, Galdi C (2004) Hyppocrates: a new proactive password checker. Journal of Systems and Software 71(1-2):163–175

Bonatti PA, Galdi C, Torres D (2015) Event-driven RBAC. Journal of Computer Security 23(6):709–757, DOI 10.3233/JCS-150539

Cai D, He X, Han J, Zhang HJ (2006) Orthogonal laplacianfaces for face recognition. IEEE transactions on image processing 15(11):3608–3614

Castiglione A, Catuogno L, Del Sorbo A, Fiore U, Palmieri F (2014) A secure file sharing service for distributed computing environments. Journal of Supercomputing 67(3):691–710, DOI 10.1007/s11227-013-0975-y

Catuogno L, Galdi C (2010) On the security of a two-factor authentication scheme. In: Proc. of 4th Intl. Workshop on Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks (WISTP), pp 245–252

Catuogno L, Galdi C (2014a) Analysis of a two-factor graphical password scheme. Int Journal of Information Security 13(5):421–437

Catuogno L, Galdi C (2014b) On user authentication by means of video events recognition. J Ambient Intelligence and Humanized Computing 5(6):909–918

Catuogno L, Dmitrienko A, Eriksson K, Kuhlmann D, Ramunno G, Sadeghi AR, Schulz S, Schunter M, Winandy M, Zhan J (2009) Trusted virtual domains–design, implementation and lessons learned. In: Trusted Systems, Springer, pp 156–179

Catuogno L, Löhr H, Winandy M, Sadeghi AR (2014) A trusted versioning file system for passive mobile storage devices. Journal of Network and Computer Applications 38:65–75

Catuogno L, Galdi C, Riccio D (2016) Flexible and robust enterprise right management. In: IEEE Symposium on Computers and Communication, ISCC 2016, Messina, Italy, June 27-30, 2016, pp 1257–1262, DOI 10.1109/ISCC.2016.7543909

Ciaramella A, D'Arco P, De Santis A, Galdi C, Tagliaferri R (2006) Neural network techniques for proactive password checking. IEEE Trans on Dependable and Secure Computing 3(4):327–339

EMC Corporation (2003) Emc documentum. http://www.emc.com/enterprise-content-management/documentum/index.htm

Gasmi Y, Sadeghi AR, Stewin P, Unger M, Winandy M, Husseiki R, Stüble C (2008) Flexible and secure enterprise rights management based on trusted virtual domains. In: Proceedings of the 3rd ACM workshop on Scalable trusted computing, ACM, pp 71–80

GlobalPlatform (2011) Tee system architecture v1.0. http://www.globalplatform.org, last visited: Jan. 9, 2018

Goshtasby A (1988) Image registration by local approximation methods. Image and Vision Computing 6(4):255–261

Grimm M, Anderl R (2013) Intellectual property protection and secure knowledge management in collaborative systems engineering. Procedia Computer Science 16:571–580

Gupta A, Kirkpatrick M, Bertino E (2014) A formal proximity model for rbac systems. Computers and Security 41:52–67, DOI 10.1016/j.cose.2013.08.012

Haller NM (1994) The S/KEY one-time password system. In: Proceedings of the Symposium on Network and Distributed System Security, pp 151–157

Hopper NJ, Blum M (2001) Secure human identification protocols. In: Proc. of 7th Intl. Conf. on the Theory and Application of Cryptology and Information Security, (ASIACRYPT),

pp 52–66
Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern recognition 38(12):2270–2285
Jiang Q, Chen Z, Li B, Shen J, Yang L, Ma J (2017) Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. Journal of Ambient Intelligence and Humanized Computing DOI 10.1007/s12652-017-0516-2, URL https://doi.org/10.1007/s12652-017-0516-2
Juels A, Sudan M (2002) A fuzzy vault scheme. In: Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on, IEEE, p 408
Kirkpatrick MS, Bertino E (2010) Enforcing spatial constraints for mobile RBAC systems. In: Proc. of the 15th ACM Symp. on Access Control Models and Technologies (SACMAT), pp 99–108, DOI 10.1145/1809842.1809860
Li F, Rahulamathavan Y, Conti M, Rajarajan M (2015) Robust access control framework for mobile cloud computing network. Computer Communications 68:61–72
Maniatis P, Akhawe D, Fall KR, Shi E, Song D (2011) Do you know where your data are? secure data capsules for deployable data protection. In: HotOS, vol 7, pp 193–205
Martinez AM (1998) The AR face database. CVC Technical Report24
Matsumoto T (1996) Human-computer cryptography: An attempt. In: ACM Conf. on Computer and Communications Security, pp 68–75
McDonald DL, Atkinson RJ, Metz C (1995) One time passwords in everything (OPIE): Experiences with building and using stronger authentication. In: Fifth USENIX UNIX Security Symposium
Microsoft Corporation (2016) Azure information protection. https://azure.microsoft.com/en-gb/services/information-protection/
Milborrow S, Nicolls F (2008) Locating facial features with an extended active shape model. In: European conference on computer vision, Springer, pp 504–513
Park SW, Lim J, Kim JN (2015) A secure storage system for sensitive data protection based on mobile virtualization. International Journal of Distributed Sensor Networks 2015
Rathgeb C, Uhl A (2011) A survey on biometric cryptosystems and cancelable biometrics. EURASIP Journal on Information Security 2011(1):1–25, DOI 10.1186/1687-417X-2011-3
Riccio D, Galdi C, Manzo R (2016) Biometric/cryptographic keys binding based on function minimization. In: 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)
Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2):38–47
Sirovich L, Kirby M (1987) Low-dimensional procedure for the characterization of human faces. Josa a 4(3):519–524
Suo X, Zhu Y, Owen GS (2005) Graphical passwords: a survey. In: Proceedings of 21st Annual Computer Security Application Conference (ACSAC), pp 463–472
Turk MA, Pentland AP (1991) Face recognition using eigenfaces. In: Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on, IEEE, pp 586–591
Wu L, Yuan S (2010) A face based fuzzy vault scheme for secure online authentication. In: Proceedings of the 2010 Second International Symposium on Data, Privacy, and E-Commerce, IEEE Computer Society, Washington, DC, USA, ISDPE '10, pp 45–49, DOI 10.1109/ISDPE.2010.13, URL http://dx.doi.org/10.1109/ISDPE.2010.13
Xu D, Chen J, Liu Q (2018) Provably secure anonymous three-factor authentication scheme for multi-server environments. Journal of Ambient Intelligence and Humanized Computing DOI 10.1007/s12652-018-0710-x, URL https://doi.org/10.1007/s12652-018-0710-x
Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: A literature survey. ACM computing surveys (CSUR) 35(4):399–458