Access control for the Cloud based on multi-device authentication

L. Gonzalez-Manzano
Dept. of Computer Science
Uni. Carlos III de Madrid
lgmanzan@inf.uc3m.es (Corresponding author)

J. M. de Fuentes
Dept of Computer Science
Uni. Carlos III de Madrid
jfuentes@inf.uc3m.es

A. Orfila
Dept of Computer Science
Uni.Carlos III de Madrid
adiaz@inf.uc3m.es

***Abstract*—Cloud-based storage services such as Box or Dropbox are proliferating. They are being commonly adopted to store private information, which is beneficial for resource-constrained devices such as smartphones. However, stealing such device must not enable the attacker to have access to cloud data. In this paper, an access control mechanism for such scenario is proposed. It leverages the fact that each person usually carries several connected devices, thus forming a personal network previously referred to as Internet–of–You (IoY). Results show that this mechanism is resilient against several attacks; it is feasible in a real world scenario; and it is specially appropriate for files larger than 20kb as bigger files reduce the capacity of attack.**

***Keywords*-Internet–of–You (IoY); Internet–of–Things (IoT); multi-device authentication; access control**

I. Introduction

In recent years, a huge amount of devices are being adopted by users. The fridge, smart TV sets or even the coffeemaker are becoming connected devices. This trend has been called Internet–of–Things (IoT) and has received a great attention from the research community. It has received different names being Body Area Networks (BAN) [1] and Internet–of–You (IoY) [2] the most prominent ones. In the remaining of this paper we will adopt the term IoY to refer to these networks.

The widespread connectivity in modern societies is promoting the increase in the amount of data managed in mobility. Thus, accessing to office reports, preparing budgets and/or commercial proposals using a tablet or even a smartphone is becoming more and more frequent. Taking into account that these devices offer a reduced amount of storage, cloud-based storage services such as Dropbox or Google Drive are attracting users from mobile devices. Large companies are making this adoption easier – for example, Office users will be able to use Dropbox from mobile applications[1].

Regardless of the considered cloud-based storage service, data should only be accessed by its owners. Different authentication mechanisms have been proposed for IoY scenarios. For example, some smartphones include multi-factor authentication (namely a fingerprint and a password[2]). Likewise, authentication by proximity is a promising approach, e.g. phones with Android 5.0 will be able to keep Chrome OS devices unlocked just by being in the area[3].

Leveraging IoY devices as a form of authentication against external services has already been explored in the past. For instance, again on the bases of proximity access control, [3] presents a protocol to access data stored in a computer only if a user's device is close to it. By contrast, focused on data stored in a remote server, [4] presents a multi-device and multi-service authentication to enable the server to verify the legitimacy of different devices.

This paper presents a quite different approach which applies the concept of proximity access control to leverage the IoY for accessing third-party services. Let's consider the case in which the user has some pictures uploaded to a cloud-based storage service (e.g. Dropbox). It would be desirable for the user to set up access control policies that enable accessing the pictures only if he carries his smartwatch, his smartphone and his RFID-enabled wallet.

In this paper, a novel access control mechanism for the considered IoY scenario is presented. It leverages on the set of devices forming the IoY to authenticate the user. Data stored in a honest-but-curious remote server is accessible once a configurable amount of IoY devices are in close proximity. In this way, the attacker needs to control as many devices as stated by the said threshold to gain access to the intended data.

The remainder of this paper is organized as follows. Section II describes the related work. Section III describes the considered model. The proposed mechanism is described in Section IV and its evaluation is shown in Section V. Section VI concludes the paper and gives future research directions.

II. Related work

Authentication is a well-known security service which has received a great research attention. Using a device as an authenticator is one of the three main ways of authentication (something you have), which complements the other two – something you know (e.g. passwords) and something you are (e.g. biometric signals).

To strengthen the authentication process, previous proposals have focused on combining the said factors. This approach is referred to as multi-factor authentication [5]. In this way, the attacker needs to get access or compromise different elements

[1] http://www.bloomberg.com/news/articles/2014-11-04/microsoft-teams-up-with-dropbox-to-target-mobile-business-users, last access February 2015.

[2] http://www.pcmag.com/article2/0,2817,2470696,00.asp, last access February 2015

[3] http://www.computerworld.com/article/2839452/android-50-security.html, last access March 2015

in order to impersonate a valid user. I. Lami et al. [6] proposes the combination of a password with users' location and time. H. Zhu et al. present Duth [7], an authentication method for Android devices that focuses on a handwriting pattern on the touch screen. The authentication is performed through heuristics composed of spatial and time characteristics. Similarly, TouchIn [8] authenticates users regarding something-they-are and something-they-known. It comprises two phases, the former to capture geometric curves chosen by the device owner and the latter to analyse authentication features, e.g. direction, concerning captured curves. J. Hu et al. [9] proposes a 3-factor authentication system for payment services based on Android. A password, a USIM card and a facial biometric recognition are applied as authentication factors. More recently, S. H. Khan et al. [10] proposes the use of random projections to biometric data using keys derived from passwords.

Each of the aforementioned factors have their own security threats and disadvantages [11]. Particularly, biometric mechanisms are often too invasive and require a particular environment to be successful. Using known information is prone to errors due to memory issues. On the other hand, devices may be lost or stolen.

With the spreading of small devices which can be easily carried (portable) or even weared (wearable), new authentication proposals have been presented. Chen and Sinclair have coined the term "Tangible security". In their approach, data in the user smartphone is decrypted as long as the remaining user-related tokens, e.g. wearable devices, are in the proximity [12]. The proximity is also the key in the "Zero-interaction authentication" by Corner and Noble [3]. The user is able to log into the computer just by carrying an authenticating device. Whenever such a device is separated, user's data into the computer is encrypted.

The abundance of carried devices which are routinely used, along with the increase of connectivity, makes them suitable to store personal information. Previous attempts have spread the sensitive data among the different elements, using lightweight crypto mechanisms to protect it [13]. To decrease the threat of data theft, proximity-based mechanisms have also been proposed. As an example, Peeters et al. propose an scheme in which devices cooperate to make an operation (e.g. decrypt some data) [14]. L. Shi et al. propose BANA [15], a node authentication scheme for body area networks based on variations of behaviour among sensors located in the body.

In this work, the set of devices carried by a user become an indicator of her presence. This direction has already been explored. Hulsebolch et al. propose a context-sensitive adaptive authentication, in which a given authentication mechanism is more or less stringent depending on the user context [16]. Such a context is determined by fusing data coming from each of the user-related devices. Their proposal depends on the reliability of the measured data, which is unsuitable for strict scenarios such as the access to private information. Likewise, C. Williams et al. apply identity-based cryptography to access data by an IoY device in emergencies [13]. Also in the field of authentication, J. Huang et al. present SEMMAP [4], an authentication protocol to enable the server to verify the legitimacy of different users' devices when accessing different services.

Aforementioned approaches focus on the proximity of elements. For example, they could not automatically log the user into a Dropbox account since it's not feasible to be physically close to Dropbox servers. Therefore, the use of proximity access control in regard to multiple devices to access data stored in a third party has not been already explored.

III. Model

This Section introduces the main elements of the proposed protocol. First, Section III-A describes the system entities. Afterwards, Section III-B presents the trust and adversarial model and Section III-C describes the objectives. The notation in use throughout this paper is shown in Table I.

TABLE I
NOTATION

Symbol	Meaning
K_{Gj}	Group Key of user j
K_{Aij}	Authentication Key of device i of user j
CH	Challenges
Yi'	Challenge response calculated by C
Yi	Challenge response calculated by each D_i
X_{Cij}	Secret parameter of C per D_i and user j for D-H protocol
X_{Dij}	Secret parameter of D_i of user j for D-H protocol
$E_K(F)$	File F encrypted with Key K
$D_K(F)$	File F decrypted with Key K
C	Cloud-based storage server
D_i	Device i carried by the user
MD	Master Device used by the user to connect to C
TH	Threshold value. The maximum refers to all D_i plus MD

A. Entities model

There are three entities in the considered scenario, namely the Cloud (C), the Master Device (MD) and a set of regular Devices (D_i), a minimum of one D_i in particular.

Cloud C stores users' encrypted files and manages authentication. It provides data to (or receives data from) MD when the user is properly authenticated. On the other hand, MD corresponds to a portable device that has a significant amount of memory, storage and processing power, e.g. a smartphone. Finally, the most resource-constrained entities are Devices D_i. They are wearable elements (e.g. a smartwatch, a smart brazelet, etc.) having limited memory, storage and processing power.

While C and MD are seen as unique entities, there may be several devices D_i per user. Particularly, the user sets a minimum threshold TH of devices that have to take part in the protocol (D_i plus MD).

B. Trust and adversarial model

The Cloud is considered honest-but-curious. It is assumed that this entity 1) does not tamper data, 2) honestly executes the proposed scheme and 3) tries to learn the content of stored files [17]. $TH - 1$ entities, D and/or MD, may be compromised in the authentication and the upload or download of files. Devices can be fully compromised, thus all managed data can be accessible to attackers. However, all entities are considered trusted in the initialization and in the inclusion of a new device D_i.

Regarding the adversary adv, her goal is to get access to the information stored in C. For this purpose, she can perform the following malicious actions:

- Compromise MD to alter messages received from D_i or create new ones on their behalf. Thus, the user is authenticated in the absence of the required set of D_i.
- Cause a sybil attack. A fake device D_i is involved in the authentication process.
- Cause a replay attack. adv intercepts messages exchanged from devices D_i to be used in future requests and to impersonate D_i.
- Steal MD. adv makes multiple requests to C to download as many files as possible.
- Compromise $TH-1$ entities in the authentication or in the download of files.

It must be noted that Denial of Service attack is out of the scope since it does not lead adv to her intended goal, but to interrupt the service provision.

C. Objectives

The design objectives of this mechanism are the following ones:

- Access control to cloud data: encrypted files stored in C have to be available for MD after the authentication process.
- Multi-device authentication: authentication should succeed when the right set of D_i and MD are involved in the process. Such a right set is formed by at least TH devices.
- Resource efficiency: the computation time and storage space for each D_i should be minimized along the whole mechanism.

IV. Protocol Description

This protocol consists of four phases – the initialization, the authentication, the upload/download of a file and the inclusion/removal of a device D_i. An overview of the protocol is presented in Section IV-A. Afterwards, each phase is described in a separate Section.

A. Overview

The proposed protocol aims to provide access control for cloud-based storage servers using the set of IoY devices as authentication elements. The use case scenario is the access to encrypted data stored in the cloud from the smartphone (Figure 1).

In the considered scenario, Cloud C server is accessed by the user by means of a Master Device MD. To provide with a higher authentication strength, the user needs not only MD, but also a set of carried devices D_i. D_i are connected to MD by means of short-range communication technologies such as Bluetooth or NFC.

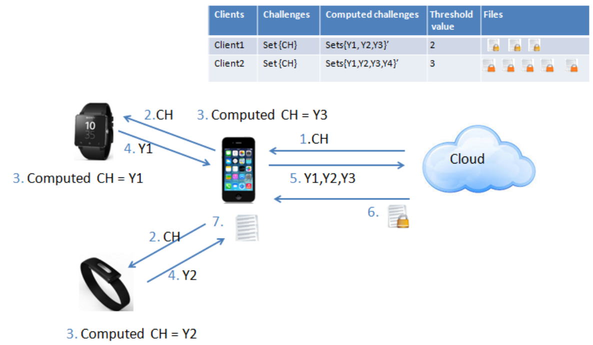


Fig. 1. System overview.

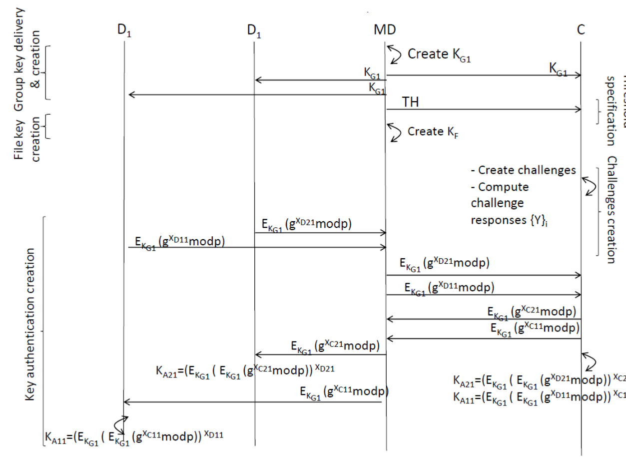


Fig. 2. Initialization phase.

In the beginning, a set of parameters are shared between D_i, MD and C. These parameters are updated each time a new D_i is introduced or a former one is no longer active.

When the user (by means of her MD) wants to upload an encrypted file to C, a proof of presence of a given set of D_i is required first. These proofs are sent in a challenge-response fashion. At least TH devices have to take part to authorize the operation. Once the authentication is successful, the file is uploaded to C. The same process is followed when the user wants to download a file. Thus, only if enough devices are present the user can access her files.

B. Initialization

The initialization consists of the creation and distribution of the Group Key (K_{Gj}), the creation of the Files Key (K_F), the specification of the threshold value (TH), the creation of challenges (CH) and the creation of the Authentication key (K_{Aij}). This phase is depicted in Figure 2.

First of all, MD creates K_{Gj} and sends it to every D_i. This key is used to ensure confidentiality between the Master Device (MD) and every D_i. It is sent to C together with the user-defined amount of devices (TH). Subsequently, MD or the user creates K_F, which is the symmetric file encryption key. K_F is stored in MD in protected form (e.g. encrypted with the user-defined password to unblock the smartphone).

Afterwards, C creates a set of challenges CH_i and the

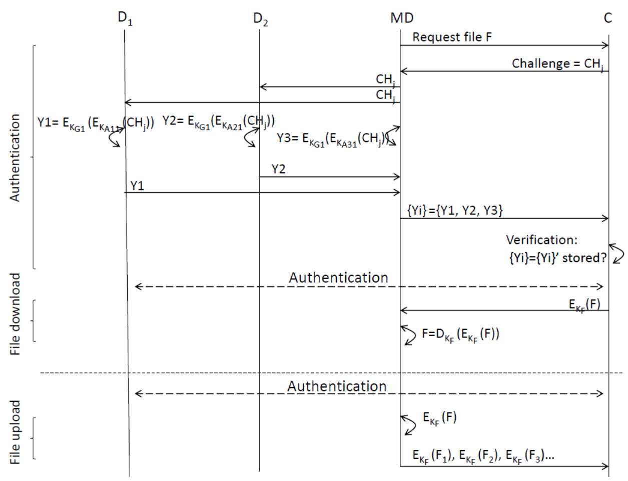


Fig. 3. Authentication and File download and upload phases.

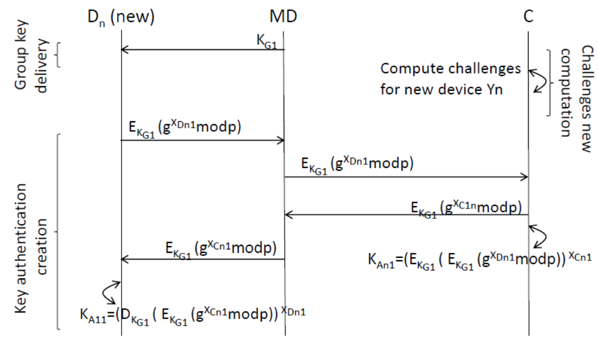


Fig. 4. Device inclusion or removal phase.

expected response (called Yi') for each one. A challenge CH_i is defined as a random string of characters of length, e.g., 1024 bits. They will be applied in the Authentication phase. Note that challenges CH are related to session time which refers to the time a user can request files without been re-authentication. Each time a session is opened a challenge CH_i is requested.

Subsequently, each D_i applies the Diffie-Hellman protocol to create a key K_{Aij} [18]. This key will be used for each device D_i to authenticate against C through MD, being MD unable to access these messages. This process works as follows. Let p be a public prime number p and g a primitive root modulo p. Each D_i creates $g^{X_{Dij}} mod(p)$ and encrypts it with K_{Gj}. Then, it is sent to C through MD. Subsequently, C sends $g^{X_{Cij}} mod(p)$ to MD to be delivered to D_i. Finally, D_i and C are able to derive K_{Aij} through X_{Cij} and X_{Dij} respectively, $K_{Aij} = g^{X_{Dij}} mod(p)^{X_{Cij}}$.

C. Authentication

Depicted in Figure 3, the MD requests a file to C and specifies who is the associated user. Then, C sends a challenge CH_i to MD requesting the computation of such CH_i by MD and all D_i involved in the authentication process. MD and D_i then compute the challenge response (Yi). This computation requires the encryption of CH_i with K_{Aij} and afterwards, with K_{Gj}. These responses are sent back to C through MD. C compares Yi against the ones calculated in the initialization (i.e. Yi') and verifies that the amount of matches are equal or higher than TH. In this case the authentication succeeds and the *File download or upload* may start. Otherwise, the protocol finishes.

D. File download and upload

After a successful authentication, the mechanism enables the user to download encrypted files F from C or to upload them to C. Particularly, files will be uploaded using key K_F (i.e. $E_{K_F}(F)$ is uploaded). This result is stored in C and sent back to the user (i.e. to her MD) when desired. Thus, MD is able to decrypt them ($F = D_{K_F}(E_{K_F}(F))$) since it knows K_F.

E. Device inclusion or removal

The inclusion or removal of a new device D_n is equivalent to the initialization, see Figure 4. In case of a new device, MD sends K_{Gj} to D_n and the Diffie-Hellman protocol is executed between D_n and C to establish K_{Anj}. Finally, C computes Yn attached to D_n and stores them together with K_{Anj} for future authentications.

In case of a device removal, this process is with all remaining devices D_i. Therefore, the group key K_{Gj} is updated in such a way that it is unknown to previous devices.

V. EVALUATION

In this Section, the proposed mechanism is assessed. The analysis focuses on the three design goals (recall Section III-C). Particularly, Section V-A studies how the authentication and access control objectives are met. Section V-B studies the computational/storage efficiency of the proposal considering current technologies.

Apart from these results, the last part analyzes the practical robustness of the mechanism. How difficult is for an attacker to break the system? This question is addressed on Section V-C. It is straightforward to see that this issue depends (among other factors) on the validity time of the authentication. Therefore, the session time length has to be considered. This parameter is discussed in V-B.

A. Objectives assessment

We next discuss whether the authentication and access control goals are met despite the considered adversarial model (recall Section III-B):

- Multi-device authentication: Key K_{Gj} is created by MD and shared with all user devices D_i in the initialization phase. Given that in this phase all devices are in safe mode, this key is only known to these parties. Furthermore, key $K_{A_{ij}}$ is built after a Diffie-Hellman exchange between D_i and C. Considering an appropriate value for p (i.e. big prime number), the Discrete Logarithm Problem (DLP) [19] prevents other parties (and particularly MD) to derive this value. In this way, only the intended devices may be authenticated.
- Access control to cloud data: Files are only downloadable after authentication. When the download of requested

TABLE II
ANALYSIS PARAMETERS

Parameter	Value
Shared secret size (i.e. size of modulo in Diffie-Hellman) (bits)	1024
File size (bits)	80000000
Challenge size (bits)	1024
Key size (bits)	1024
Amount of devices	2
Amount of precomputed Yi	10
Threshold size (bits)	8
File request size (bits)	4096

files finishes, authentication is required again. The authentication requires the computation of different challenges each time. Even if some computed challenges are captured, they cannot be reused because fresh ones are needed in next authentications. Consequently, even if MD is stolen, just downloaded files, unless they are appropriately removed, remain accessible to the attacker. Nonetheless, there is a situation in which access control could be compromised. Particularly, consider that MD is stolen and remains close to TH-1 D_i. This could enable access to files without the user consent. This scenario is deeply analysed in Section V-C. It must be noted that user-defined parameter TH serves as a countermeasure itself in that access control is enforced unless $TH - 1$ D_i and MD are controlled by the attacker.

B. Performance analysis

In order to assess the real-world suitability of the proposal, it is necessary to consider the state-of-the-art features of the involved devices. Section V-B1 describes the features of the considered entities. Afterwards, Section V-B2 illustrates the computational cost of individual operations. We assume that crypto-related operations are the most intensive ones in the proposed protocol. Thus, other underlying issues such as message transmission or reception are considered negligible.

Considering the previous figures, Section V-B3 analyses the cost per phase. The parameters considered for these calculations are shown in Table II. We will assume that all devices will be participating, no matter the value of TH. Furthermore, for the sake of simplicity, only one file will be at stake.

1) Considered entities features: According to [20], current smartwatches are equipped with processors that range from a single 120 Mhz. chip up to a quad-core computational unit with 1200 Mhz per core. To illustrate its performance, figures from a constrained ARM Cortex A8 processor with a single 800 Mhz core have been considered. Its frequency is much nearer to the device with lowest resources than the most powerful one, which is suitable for the sake of this analysis. In fact, this processor speed is very close to that of the Motorola Moto 360 wearable device [21]. It must be noted, however, that the processor architecture may take a critical role when it comes to performance. This fact must be taken into account when analyzing the results.

With respect to the Master Device (MD) processor, specifications of a middle-price smartphone call for a 1 Ghz, dual core unit [22]. To illustrate its behavior, a NVIDIA Tegra 250 GPU will be considered. Such device is also present in a broad range of current smartphones [23].

TABLE III
CRYPTO PERFORMANCE OF CONSIDERED DEVICES

Operation	Time ARM Cortex A8 (ms)	Time NVIDIA Tegra 250 (ms)	Time Intel Xeon E5-620 (ms)
Diffie-Hellman (1024). Keypair generation	43.533	7.952	0.253
Diffie-Hellman (1024). Secret derivation	43.474	7.931	0.249
Symmetric Encryption/Decryption (AES CTR 256) (per byte)	5.04 e^{-5}	2.90 e^{-6}	1.89 e^{-7}
Asymmetric Encryption (RSA 1024). Keypair generation	1635.939	365.887	7.721
Asymmetric Encryption (RSA 1024). Encryption (per byte)	6.59 e^{-3}	1.83 e^{-3}	1.06 e^{-4}
Asymmetric Encryption (RSA 1024). Decryption (per byte)	0.310	0.055	0.002
Hash (SHA-256). Hash (per byte)	3.10 e^{-5}	1.14 e^{-5}	1.64 e^{-6}
Signature (RSA 1024). Creation (per byte)	0.310	0.055	0.002
Signature (RSA 1024). Verification (per byte)	0.006	0.001	9.15 e^{-5}

For the sake of completeness, processing capabilities of the cloud have also been considered. In particular, Amazon EC2 offers specific instances for data storage such as the I2 model. This model features four Intel Xeon E5-2670 v2 processors. Given that no performance figures have been published regarding this platform, in this analysis an Intel Xeon E5-620 will be considered.

2) Computational cost of operations: Previous cryptographic benchmarks have shown the computational cost of cryptographic operations in the considered platforms (see Table III) [24].

3) Analysis per phase: For the sake of clarity, the analysis will be divided into the Initialization, Authentication and File download/upload phases. According to the protocol description, the Device inclusion/removal phase is formed by a subset of the operations carried out into the Initialization. For each phase, the amount of data stored and sent by each participant will be shown, as well as the computation time taken.

- Initialization phase. Table IV shows the computation time as well as the amount of data exchanged by each entity. In this phase, MD does not perform any cryptographic computation – all computation workload is on D_i and C sides. It is clear that the most constrained device (D_i) takes most of the time spent in this phase. Regarding data stored, MD needs to keep the group key K_Gj and the file encryption key K_F. The main data exchanged is the key K_Gj as well as the Diffie-Hellman key exchange.
- Authentication phase. The performance figures of this phase are summarized in Table V. It must be noted that thanks to the precomputations made in the Initialization phase, no crypto operations are carried out by C. Regarding data exchanged, challenges and responses are sent back and forth through MD. It must be noted that no

TABLE IV
PERFORMANCE FIGURES FOR INITIALIZATION PHASE

	Master Device (MD)	Device (Di)	Cloud (C)
Computation time (ms)	0	87.019	5.338 e^{-4}
Stored information (bits)	2048	2048	23552
Data sent (bits)	7176	1024	2048
Data received (bits)	4096	2048	3080

TABLE V
PERFORMANCE FIGURES FOR THE AUTHENTICATION PHASE

	Master Device (MD)	Device (Di)	Cloud (C)
Computation time (ms)	7.43 e^{-4}	0.012	0
Stored information (bits)	0	0	0
Data sent (bits)	9216	1024	1024
Data received (bits)	3072	1024	7168

entity has to store anything, which is very convenient for scalability purposes.

- File upload/download. In this phase, devices D_i do not take part in the protocol. Only MD and C participate, transferring the file at stake. For the sake of brevity, only the calculation for the computation time of MD to upload a file is explained in detail. The computation of MD is to encrypt the file at stake. The time taken is proportional to the file size.

$$\begin{aligned} T_{comput}(MD) &= T_{enc}(b\ bytes) * (FileSize(bytes)/b) \\ &= 2.9e^{-6} * (10^7/1) = 29.049ms \end{aligned} \tag{1}$$

In general terms the amount of stored information is suitable for each considered device. Particularly, devices D_i only require to store 2048 bits, which is much less than their current storage capabilities. For example, Sony's SWR50 smartwatch is equipped with 4 Gb of Flash memory[4].

4) Session time analysis: Considering the protocol design, one of the most relevant security-related parameters is the session time. If it is very long, then the attacker may be able to steal MD (i.e. the smartphone) and continue downloading/uploading even without the required devices – once authentication is performed, it will last for a period of time to avoid wasting resources. On the contrary, if the session is too short, it may not be enough for the intended files to be transmitted. For the sake of clarity, in the following the download operation will be considered, although it is the same calculation for the upload one.

The general expression for the time of authenticating the devices and downloading the intended files is given by the following Equation.

$$T_{download}(File) = T_{MD} + T_{Di} + T_{comm} \tag{2}$$

The time $T_{download}(File)$ is given by the sum of the time taken by MD (T_{MD}), that of the portable devices (T_{Di}) and the communication time (T_{comm}). It must be noted that as all devices D_i perform their calculations in parallel, this time does not depend on the amount of participating devices nor the value of the threshold TH.

Regarding the time T_{MD}, it is due to the preparation of its own challenge response, as well as the time taken to decrypt the file itself. As the encryption method is symmetric, we

[4]http://www.sonymobile.com/es/products/smartwear/smartwatch-3-swr50/specifications/, last access February 2015

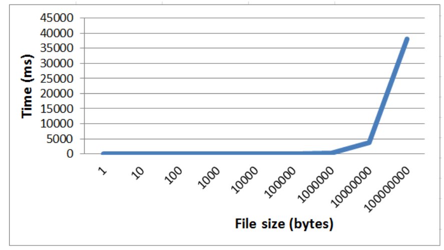


Fig. 5. Evolution of minimum session time depending on file size

will assume that the computation time for decryption grows linearly with the file size. Thus, the time taken to decrypt 2 Mb will be twice that to decrypt 1 Mb. Furthermore, we will assume that the time may be applied to more than one file – it is assumed that the time to decrypt a single 2 Mb file is the same as to decrypt two files, 1 Mb each.

$$\begin{aligned} T_{MD} = 2 * (T_{encMD}(1bytes) * CH\ Size(bytes)) + \\ (T_{decMD}(1byte) * File\ size(bytes)) \end{aligned} \tag{3}$$

With respect to the time T_{Di}, it is the one needed to compute the challenge response. It is important to note that the computation time will be bigger than that of MD due to the resource constraint.

$$T_{Di} = 2 * (T_{encDi}(1\ bytes) * CH\ Size(bytes)) \tag{4}$$

Finally, the communication time is the required to transmit the challenges CH and their computed answers Yi, as well as the files at stake. Given that the size of CH is assumed to be negligible as compared to that of the files, the communication time is reduced to the file transmission. The resulting expression is as follows, being $nDev$ the amount of participating devices.

$$\begin{aligned} T_{comm} = (nDev + 1) * (T_{trans}(1\ bytes) * CH\ Size(bytes)) \\ + (T_{trans}(1\ bytes) * File\ Size(bytes)) \\ \approx (T_{trans}(1\ bytes) * File\ Size(bytes)) \end{aligned} \tag{5}$$

From Equations (3), (4) and (5) above, it is intuitive to see that the variable that has a biggest impact on the value of the session is the file size. Particularly, the session time $sess_{time}$ is related to the file size as follows, considering the said Equations and value for CH size, and 21 Mbit/s as the transmission speed. The last value is taken from existing figures for 3G communications [5].

$$\begin{aligned} sess_{time} \geq T_{download}(File) = (2 * (2.9e^{-6} * 128) \\ + (2.9e^{-6} * File\ size(bytes))) + (2 * (5.04e^{-5} * 128) \\ + (1/((21 * 10^3)/8) * File\ Size(bytes)) \\ \approx 0.013 + 3.8e^{-4} * File\ size(bytes) \end{aligned} \tag{6}$$

C. Practical robustness

In order to authenticate against the cloud server, an adversary needs to control MD and be close to $TH - 1$ devices. Alternatively, she has to be able to mimic their behaviour.

[5]http://en.wikipedia.org/wiki/4G, last access February 2015

Assuming the channel between the MD and the cloud is properly authenticated and confidential, an adversary needs to steal or compromise the MD first in order to read files. If the MD is compromised she can change the group key given she is in the proximity of $TH-1$ D_i. Therefore the system would become compromised. If the MD is stolen the authentication process can not take place unless the MD is close to $TH-1$ D_i and the files can neither be retrieved nor the group key changed. Thus, an stolen MD needs $TH-1$ correct responses to compromise the system.

Under previous considerations, let r_i be the distance between D_i and MD at the moment of a theft/ loss (considering r_i does not change) and R the nominal transmission range of MD. For simplicity, it will be assumed that the attacker steals MD and stops moving while the victim moves at speed S, the time the attack may success requires the victim and the attacker to be within the transmission range, as well as the attacker to receive $TH-1$ correct responses. Given that responses are simultaneously received, the time is bounded by the D_i with the maximum r_i as it would be the first one out of range. Then:

$$Time(attack) = \frac{R - max_{TH-1}(r_i)}{S} \quad (7)$$

Let p be the probability of theft/loss of MD, the attacking capacity ($CAP(attack)$) is measured as p multiplied by the amount of times the protocol can be executed within $Time(attack)$ where both issues are considered independent events. Accordingly, $if\ r_i > R$ for $TH-1$ D_i, $CAP(attack)$=0. But if $r_i < R$ this capacity is measured as follows:

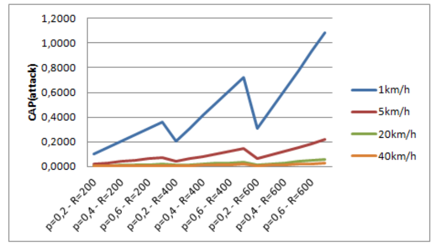
$$CAP(attack) = p * num\_executions = p * \frac{Time(attack)}{sess_{time}} \quad (8)$$

where $sess_{time}$ depends on the file size to be downloaded (recalling Section V-B4).

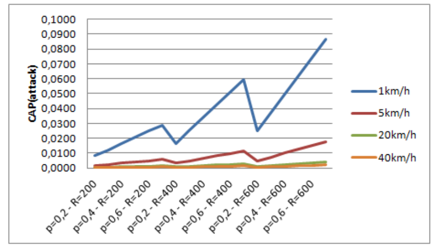
Now the point is to identify elements which affect $CAP(attack)$. We assume the following general scenario in which a MD is stolen. R ranges from 100m to 600m increasing in 200 units, S is set to {1, 5, 20, 40} km/h, file size is set to {20kb, 2mb, 20mb} and for the sake of simplicity but without losing generality, r_i=1 $\forall$ D_i. Note that 40 km/h is set to be the maximum running speed [6] and 5 km/h is set to be the average walking speed [25] of a human. Depicted in Plots 6(a)-6(c), it is noteworthy that S is the factor that affects $CAP(attack)$ the most. A successful attack involves being within R. Thus, the lower S, the longer the time the adversary can stay within R and then, the higher the available $Time(Attack)$. For example, for file size=20kb, p=0.3 and R=400, if S=5km/h $CAP(attack)$=0.06 and if S=20km/h $CAP(attack)$=0.015. In this regard, with S=20km/h $CAP(attack)$ decreases 75% and with S=40km/h $CAP(attack)$ decreases 88%.

File size affects $CAP(attack)$ as well, although to a lesser degree. Indeed, the joint analysis of the file size and S is particularly remarkable. Being 2.9 the maximum $CAP(attack)$ which is achieved when p=0,7, R=400, file size = 10kb and S=1km/h, $CAP(attack)$ can be considered negligible A) for files bigger than 20kb when A.1) S=5km/h ($CAP(attack)$=0.09 on average) and A.2) S=20km/h ($CAP(attack)$=0.02 on average) and B) for all kind of files when S is higher than S=20km/h ($CAP(attack)$ order of $10^{(-3)}$).

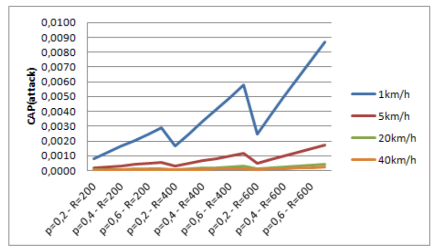
[6]http://www.topendsports.com/resources/records/speed.htm , last access February 2015



(a) $CAP(attack)$ for file 20Kb



(b) $CAP(attack)$ for file 2Mb



(c) $CAP(attack)$ for file 20Mb

Fig. 6. $CAP(attack)$ analysis

Besides, p and R also affects $CAP(attack)$, which increases according to both parameters. For example, given file size=20kb and p=0.4, for R=400, if S=5km/h then $CAP(attack)$=0.08 and if S=20km/h then $CAP(attack)$=0.02; and for R=200, if S=5km/h then $CAP(attack)$=0.04 and if S=20km/h then $CAP(attack)$=0.01. In both cases the change of R affects 50% $CAP(attack)$. However, as $CAP(attack)$ is significantly small when S >10km/h (order of $10^{(-2)}$), it can be concluded that R does not affect $CAP(attack)$ to a great extent. Similar conclusions are drawn when p is modified, according to Equation 8, changes in p are directly proportional to changes in $CAP(attack)$.

As a result, though $CAP(attack)$ is highly dependent on S, as the victim does not know her speed in case of MD is stolen, the best choice to get a small $CAP(attack)$ is the use of the cloud for the storage of large files. Considering S=1km/h, which is one possible worst case, $CAP(attack)$ is significantly small for files of size higher than 20kb ($CAP(attack)$ <0.49 on average) and it is almost negligible for files of size higher than 20Mb ($CAP(attack)$ <0.003 on average).

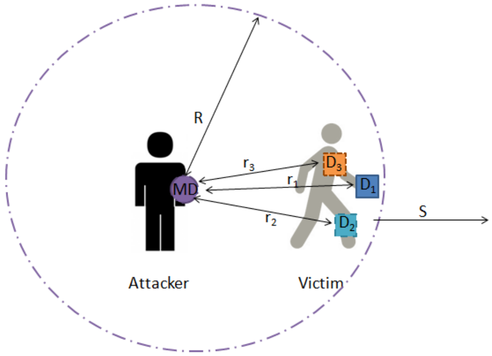


Fig. 7. Relationship between TH and $Time(attack)$. $TH-1$=3

One last appreciation is that TH indirectly affects $CAP(attack)$ because the more devices are involved in the protocol, the more possibilities to reach a lower $Time(attack)$. For instance, depicted in Figure 7, being $TH-1$=3, D_1 is close to the maximum R, thus $R - r_1$ is small and $Time(attack)$ becomes a low value.

VI. Conclusions

The use of sensitive information from portable devices is growing every day. As these devices have constrained storage, data may be saved in the cloud. In order to prevent unauthorized access to such information, access control mechanisms are needed.

In this paper, we have proposed a novel access control mechanism for this scenario. It leverages on the Internet-of-You (IoY), i.e. the set of connected devices that are usually carried by a user. Therefore, the user may upload or download data from the cloud as long as a predefined set of her devices are present. In this way, the attacker does not only need to compromise the device connected to the cloud (e.g. the smartphone), but also be close to a subset of devices forming the IoY. Results show that the proposed mechanism is resilient against a regular adversary. Furthermore, it is feasible in a real world scenario in terms of computation time, storage and bandwidth. It is specially suitable for files larger than 20kb because they reduce the practical capacity of attack.

Future work will be focused on two aspects. First, the mechanism will be adapted to inter-IoY scenarios, thus supporting having access to information only if more than one person is present. Second, the use of lightweight cryptographic primitives will be explored, as well as aggregation mechanisms, to improve the proposal efficiency.

Acknowledgments

This work was supported by the MINECO grant TIN2013-46469-R (SPINY: Security and Privacy in the Internet of You).

References

[1] M. Patel and J. Wang, “Applications, challenges, and prospective in emerging body area networking technologies,” *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 80–88, February 2010.

[2] C. Mims, “The internet of you: How the future of computing became screens and sensors on every appendage,” http://qz.com/42632/the-internet-of-you-how-the-future-of-computing-became-screens-and-sensors-on-every-appendage/, 2013.

[3] M. D. Corner and B. D. Noble, “Zero-interaction authentication,” in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’02. ACM, 2002, pp. 1–11.

[4] J. Huang and C.-T. Huang, “A secure and efficient multi-device and multi-service authentication protocol (semmap) for 3gpp-lte networks,” in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, July 2012, pp. 1–7.

[5] A. Bhargav-Spantzel, “Privacy preserving multi-factor authentication with biometrics,” *Journal of Computer Security*, 2007. [Online]. Available: http://iospress.metapress.com/content/3626171362X61240

[6] I. A. Lami, T. Kuseler, H. Al-Assam, and S. Jassim, “Locbiometrics: Mobile phone based multifactor biometric authentication with time and location assurance,” in *Proc. 18th Telecommunications Forum, IEEE Telfor*, 2010.

[7] H. Zhu, X. Lin, Y. Zhang, and R. Lu, “Duth: a user-friendly dual-factor authentication for android smartphone devices,” *Security and Communication Networks*, 2014.

[8] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, “Touchin: Sightless two-factor authentication on multi-touch mobile devices,” in *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE, 2014, pp. 436–444.

[9] J.-Y. Hu, C.-C. Sueng, W.-H. Liao, and C. Ho, “Android-based mobile payment service protected by 3-factor authentication and virtual private ad hoc networking,” in *Computing, Communications and Applications Conference (ComComAp), 2012*, Jan 2012, pp. 111–116.

[10] S. H. Khan, M. A. Akbar, F. Shahzad, M. Farooq, and Z. Khan, “Secure biometric template generation for multi-factor authentication,” *Pattern Recognition*, vol. 48, no. 2, pp. 458–472, 2015.

[11] A. Dmitrienko, C. Liebchen, C. Rossow, and A.-R. Sadeghi, “On the (in) security of mobile two-factor authentication,” in *Financial Cryptography and Data Security*. Springer, 2014, pp. 365–383.

[12] Y. Chen and M. Sinclair, “Tangible security for mobile devices,” in *Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, ser. Mobiquitous ’08, 2008, pp. 19:1–19:4.

[13] C. C. Tan, H. Wang, S. Zhong, and Q. Li, “Body sensor network security: An identity-based cryptography approach,” in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec ’08. New York, NY, USA: ACM, 2008, pp. 148–153. [Online]. Available: http://doi.acm.org/10.1145/1352533.1352557

[14] R. Peeters, M. Kohlweiss, and B. Preneel, “Threshold things that think: Authorisation for resharing,” in *iNetSec 2009 Open Research Problems in Network Security*, ser. IFIP Advances in Information and Communication Technology, J. Camenisch and D. Kesdogan, Eds. Springer Berlin Heidelberg, 2009, vol. 309, pp. 111–124.

[15] L. Shi, M. Li, S. Yu, and J. Yuan, “Bana: body area network authentication exploiting channel characteristics,” *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 9, pp. 1803–1816, 2013.

[16] R. Hulsebosch, M. Bargh, G. Lenzini, P. Ebben, and S. Iacob, “Context sensitive adaptive authentication,” in *Smart Sensing and Context*, ser. Lecture Notes in Computer Science, G. Kortuem, J. Finney, R. Lea, and V. Sundramoorthy, Eds. Springer Berlin Heidelberg, 2007, vol. 4793, pp. 93–109.

[17] Q. Chai and G. Gong, “Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers,” in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 917–922.

[18] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theor.*, vol. 22, no. 6, pp. 644–654, Sep. 2006.

[19] K. S. McCurley, “The discrete logarithm problem,” in *Proc. of Symp. in Applied Math*, vol. 42, 1990, pp. 49–74.

[20] (2014) 2014 smartwatch comparison guide. [Online]. Available: http://www.gizmag.com/compare-best-smartwatches-2014/34880/

[21] Motorola moto 360 specifications. [Online]. Available: https://moto360.motorola.com/

[22] (2015) bq aquaris 3.5 specifications. [Online]. Available: http://www.bq.com/es/productos/aquaris-3-5.html

[23] Tegra features. [Online]. Available: http://www.nvidia.co.uk/object/tegra-features-uk.html

[24] (2015) ebacs: Ecrypt benchmarking of cryptographic systems. [Online]. Available: http://bench.cr.yp.to/computers.html

[25] K. Aspelin and N. Carey, “Establishing pedestrian walking speeds,” *Portland State University*, 2005.