Cryptography on a Customized Network

Ricardo Martinho Ferreira Miranda
ricardo.miranda@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2017

Abstract

Building a secure network to be used in real-world applications where there are constraints strictly imposed to the capabilities of the network's elements and to the data flow requires a customized cryptographic analysis in order to protect the communications and detect and minimize the system's exploitable vulnerabilities. In this document a network under such conditions is presented and one is challenged with providing an optimal topological scheme prior to choosing the network's cryptographic components embedded in the communication and data storage protocols and posteriorly analyzing their security. Among the scrutinized alternatives a single one of them is elected as the solution by a comparison in terms of performance, security and suitability under the enforced restrictions. This solution is implemented using C and Java programming languages. The selected encryption schemes and protocols are proven to be highly reasonable options and their use in practice is advised. These results are only valid for this specific case of study, for if any of the established constraints is ruled out then it is most likely the insurgence of an enhanced solution.
Keywords: ciphertext indistinguishability; block cipher mode of operation; semantic security; symmetric cryptosystem.

1. Introduction

The first approach to secure a certain piece of information dates back to the Ancient Greece. Ever since, the humankind has been continuously developing new methods for securing desired secrets and while some create the methods to secure information, others put a lot of effort on discovering weaknesses in order to retrieve the envisaged secrets. An example that reflects cryptography's tremendous relevance in modern days is World War II. The victory of the Allies was mainly due to their ability to eavesdrop on the enemy's communications after being able to break the Enigma [5] cipher and as such, this war propelled the major advancements in the fields of cipher construction and cryptanalysis. The continuous demand for protecting information is the fuel that thrives the evolution of computational security.

In the current work one is presented with a network composed by several types of devices with certain restriction with respect to memory, space and power consumption and aims to choose the most suitable topology for the network and to create a security mechanism to be included in the communication protocol with the objective of providing the satisfiability of some cryptographic properties to the messages travelling through the network.

Cryptographic tools are undoubtedly necessary in order to grant the data critical properties like confidentiality, integrity, authentication and non-repudiation, which are crucial to nowadays growing communication needs in the prosper field of information technology.

In section 2 the most relevant state-of-the art concepts related with the developed work are addressed. In section 3 the problem is introduced, the options with regard to the network's topology and communication protocols are discussed, the general protocol for the upstream and downstream data lifecycles is presented and its underlying message formats are defined. Section 4 contains a security analysis for the network defined in section 3. Section 5 contains a general overview of the results obtained and motivation for future work on the subject.

2. Basic Concepts

Cryptanalysis is the study of cryptosystems with the objective to find flaws or weaknesses that entail a gain of information from unauthorized parties, without necessarily discovering the secret key.

Definition 2.1 (Ciphertext-only attack)**.** *A ciphertext-only attack is one where the adversary possesses information regarding the ciphertext and is able to deduce either the corresponding*

plaintext or the key, without being provided any details about the plaintext itself.

Definition 2.2 (Known-plaintext attack)**.** *A Known-plaintext attack focuses on finding the secret key (or key stream) of the cryptosystem at hand, provided the knowledge of both the ciphertext and its corresponding plaintext.*

Definition 2.3 (Chosen-plaintext attack)**.** *In a chosen-plaintext attack, the adversary has access to an encryption oracle, which encrypts any plaintext given by the attacker and outputs the corresponding ciphertext.*

Definition 2.4 (Chosen-ciphertext attack)**.** *In a Chosen-ciphertext attack, the adversary has access to a decryption oracle, which decrypts any ciphertext given by the attacker and outputs the corresponding plaintext.*

Definition 2.5 (Semantic Security)**.** *Let $\mathcal{C} = (X, Y, K, \mathcal{E}, \mathcal{D})$ be a cryptosystem and $e_k \in \mathcal{E}$. A cryptosystem is said to be semantically secure if given $y = e_k(x)$, then $V(A(y, l)) = V(B(l)$, where A and B are two polynomial-time bounded adversaries, $l = |y|$ and V is the advantage of the adversary, which is defined by*

$$V(a) = Pr(a_0) - Pr(a_1) \tag{1}$$

for every adversary a, where a_i represents the event of a choosing a wrong or the right plaintext x, respectively for $i = 0$ and $i = 1$.

Consider the following scenario: an adversary $\mathcal{A}$ living in one of two worlds (left world L or right world R) is trying to break a cryptographic system $\mathcal{C}$ and has access to an encryption oracle $\mathcal{O}_e$. $\mathcal{A}$ does not know which world lives in but the world W is defined a priori and cannot be changed throughout the entire activity of $\mathcal{A}$. The encryption oracle, given any two plaintexts p_0, p_1 always returns the ciphertext $e_k(p_b)$ where e_k is the encryption function of $\mathcal{C}$ for some $k \in \mathcal{K}$ and $b \in \{0, 1\}$ is picked according to the following relation

$$b = \begin{cases} 0 & \text{if } W = L \\ 1 & \text{if } W = R \end{cases} \tag{2}$$

$\mathcal{O}_e$ is known as lr-oracle.

Definition 2.6 (IND-CPA)**.** *Let $\mathcal{C} = (P, C, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem with encryption and decryption functions e_k and d_k, respectively, for some $k \in \mathcal{K}$, let $\mathcal{A}$ be an adversary and $\mathcal{O}$ an encryption lr-oracle. Consider that $\mathcal{A}$ is in possession of $X = (x_1, \ldots, x_n) \in \mathcal{P}^n$ and also that $|x_i| = |x_j|, \forall_{i \neq j}$. IND-CPA is a game defined as follows:*

1. *$\mathcal{A}$ picks two messages $x'_0, x'_1 \in X$;*
2. *$\mathcal{A}$ queries the oracle $\mathcal{O}$ with (x'_0, x'_1);*
3. *$\mathcal{O}$ encrypts x'_b yielding $e_k(x'_b)$, according to 2;*
4. *$\mathcal{O}$ returns the encryption output $e_k(x'_b)$ to $\mathcal{A}$;*

   ($\mathcal{A}$ can repeat steps 1 to 4 at will);
5. *$\mathcal{A}$ chooses $b' \in \{0, 1\}$;*
6. *$\mathcal{A}$ wins if $b' = b$ and loses otherwise.*

Definition 2.7 (IND-CCA)**.** *IND-CCA is a game analogous to the one from Definition 2.6, but herein the adversary has access to two lr-oracles: an encryption lr-oracle $\mathcal{O}_e$ and a decryption lr-oracle. This game has the additional requirement that the adversary $\mathcal{A}$ may not query the decryption oracle with challenge ciphertexts.*

Property 1 (IND-CPA secure)**.** *A cryptosystem is said to be IND-CPA secure if a polynomial-time bounded adversary who plays the IND-CPA game cannot win with probability negligibly greater than $1/2$.*

Property 2 (IND-CCA secure)**.** *A cryptosystem is said to be IND-CCA secure if a polynomial-time bounded adversary who plays the IND-CCA game cannot win with probability negligibly greater than $1/2$.*

The minimum threshold of security required for a cryptosystem to be acceptable with regard to its security level is to satisfy property 1.

2.1. Cryptographic Hash Functions

A cryptographic hash function outputs a fixed-length message on any given input of arbitrary length and satisfies the following properties:

(i) **Efficiency:** The computation of the hash value must be incredibly fast.

(ii) **One-Way Function:** It's infeasible to invert.

(iii) **Avalanche Effect:** A small change in the input of the hash function produces a very distinct output.

(iv) **Collision Resistance:** It is very hard to find two distinct inputs with the same image.

Definition 2.8 (HMAC)**.** *Let m be a message, k an l-bit key and h a cryptographic hash function whose compression function's block size is of n-bits. The following function f defines the HMAC-h:*

$$f(k, m) = h((k' \oplus opad) \parallel h((k' \oplus ipad) \parallel m)) \tag{3}$$

where $ipad$ and $opad$ are fixed strings and k' is the resulting key such that for $j = n - l$:

$$k' = \begin{cases} k \parallel 0^j & \text{if } l < n \\ h(k) & \text{if } l > n \\ k & \text{otherwise} \end{cases} \tag{4}$$

2.2. Randomness

The higher known level of randomness is extracted from physical elements, like for instance the movement of electrons.

Definition 2.9 (PRF)**.** *A family of functions* $\{F_k : X \to Y\}_{k\in\{0,1\}^*}$ *is a PRF if, for a randomly chosen instance function* F_k*, its output is indistinguishable (for a polynomial-time algorithm) from the output of a random function* $R : X \to Y$*, where* X *and* Y *are the domain and range sets of the functions of the family, respectively.*

3. Network

The main purpose of this work is to develop an optimal solution for the topological and cryptographical components of a restricted network. Basically, upon being provided with network requirements, which are either constraints to the network's elements and their connections or to the capabilities of the communication channels, one is intended to choose the encryption and authentication schemes and analyze their level of security for fitting state-of-the-art properties and definitions. The goal of these schemes is to provide the data cryptographic properties that will strengthen the resilience of the data stored in the network elements' memory against possible threats. The security layer of any of the protocols used for the transmission of data between any two network parties is also a relevant subject of study for it will determine the level of security of the communications.

The imposed network consists of distinct clusters such that each is composed by a fixed number of elements, each element of the same group shares a set of features and the connections between clusters are restrained under some predefined rules. The previously mentioned parties' features range from the computational power scope and available cryptographic algorithms and their respective keys to the assigned mission of the network element. More specifically, the purpose of the network is to gather real-time data and transmit it to a secure database while granting the collected evidence confidentiality, integrity, authenticity and non-repudiation properties.

3.1. Details

Let the network be composed by three main components:

- GDs: field-deployable parties that gather the raw data (with a maximum threshold of 256kB/s), process it and subsequently send it to an authorized party via an asynchronous channel. The length of each of the generated messages is a multiple of a minimum defined length l_1 and is maximized by 256 octets. These elements are restricted with respect to memory (256kB RAM) and autonomy, as their energy source is a non-rechargeable battery and remain in the same geographic location throughout the extent of their lifetime.

- MPP: a very flexible party who is able to go near the deployed gathering devices in order to wirelessly receive the data and/or send command messages. Also possesses a serial connectivity option for posteriorly transmitting the sensitive data to an authorized party.

- MnDM: headquarters' positioned device that receives the data from the middle-point party, makes the necessary verifications and stores it in a secure centralized database. It is also capable of generating and sending command messages, whose length is a multiple of a predefined minimum length l_2.

The GDs do not communicate directly with the MnDM and vice-versa. The data flow from the GD to the MPP and subsequently to the MnDM is denoted by upstream data lifecycle UDL and in the inverse direction is denoted by downstream data lifecycle DDL. The command messages are predefined formatted messages whose contents are intended to give an instruction to another network party and can only be generated by the MPP and MnDM.

For secrecy, authentication and non-repudiation purposes some cryptographic algorithms are going to be used, for which are compelled cryptographic keys. Prior to deployment there must be a setup stage, in which the required keys are generated, transmitted to the envisaged target and stored in solid memory. These keys are generated inside secure headquarters, called the PMS. Figure 1 contains a simple diagram that depicts the whole step: at the PMS, a family of keys $\mathcal{K} = (\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3)$ is generated such that $\mathcal{K}_2$ and $\mathcal{K}_3$ are the sets of keys transmitted to MPP and MnDM, respectively, and $\mathcal{K}_1 = \bigcup_{j=1}^{n} \mathcal{K}_1^j$, where n is the total number of GDs and $\mathcal{K}_1^j$ is the set of keys transmitted to GD_j, $\forall_{j\in\mathbb{N}} : 1 \leq j \leq n$. When the setup stage is concluded the devices meet the required constraints for the set up of the network.

The GDs, upon deployed, can be in one of two states: active mode or sleep mode. When the devices are in active mode they keep on gathering evidence around the clock according to their data collecting schedule and send the processed data to the envisaged end party via an asynchronous communication channel [9] according to the data flow schedule. For this reason, the data stored in solid memory of the gathering devices is expected to be encrypted. The sleep mode, in turn, is a low power consumption state in which the devices are

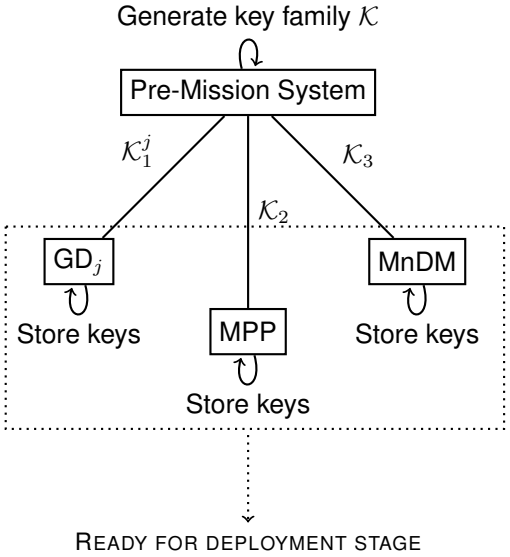

Figure 1: Pre-deployment stage

not actively performing any activity other than periodically searching for a connection to an asynchronous communication channel.

The GDs are connected to the MPP via Wi-Fi therefore the communication protocol has been decided to be WPA2-PSK. WPA2 is the most robust option for Wi-Fi communication channels and the choice of PSK over EAP follows from two facts: firstly, the gathering devices do not need to hide information from one another and secondly the EAP mode of the WPA2 protocol requires more computations for the initial authentication step and therefore it increases the energy consumption when compared with PSK.

3.2. Network Topology

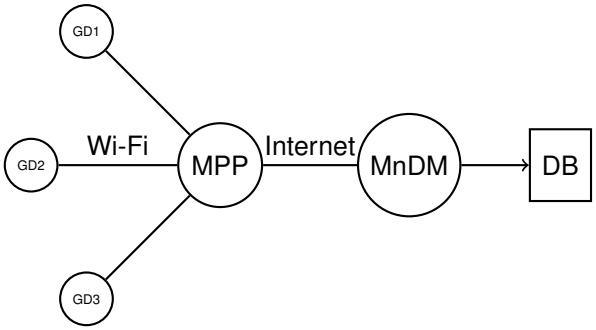

Figure 2: Topology of the chosen network.

The chosen option is represented in Figure 2 and features the MPP as the access point of the network. Since the MPP is considered to be a versatile element in the sense that it is not deployed on the field, this layout is considered to be very suitable for the features at hand.

3.3. Protocol

Let n be the number of gathering devices and GD$_i$ be the GD at hand for some fixed index $i \in \mathcal{I}$. Moreover, consider the following notation:

- $(k_A^{\texttt{GMP}})^i$: The 128-bit key shared between the GD$_i$ and the MPP, used by the AES cipher.
- $k_A^{\texttt{MPM}}$: The 128-bit key shared between the MPP and the MnDM, used by the AES cipher.
- $(k_H^{\texttt{GMP}})^i$: The 256-bit key shared between the GD$_i$ and the MPP, used in the HMAC-SHA-256 algorithm.
- $(k_H^{\texttt{GMD}})^i$: The 256-bit key shared between the GD$_i$ and the MnDM, used in the HMAC-SHA-256 algorithm.
- $k_H^{\texttt{MPM}}$: The 256-bit key shared between the MPP and the MnDM, used in the HMAC-SHA-256 algorithm.

These abbreviations are to be considered throughout the entire text.

3.4. Setup

The key generation protocol occurs at PMS and comprises the following steps:

1. The user inputs $(n, pass)$, a tuple consisting in the number of gathering devices that are going to be deployed and a password, respectively;
2. The key generation algorithm is applied with input $(n, pass)$ and outputs $\mathcal{K} = (\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3)$[1];
3. Export the keys to the envisaged devices according to the following distribution:
   - $\mathcal{K}_1^j = \{(k_A^{\texttt{GMP}})^j, (k_H^{\texttt{GMP}})^j, (k_H^{\texttt{GMD}})^j\}$;
   - $\mathcal{K}_2 = \{k_A^{\texttt{MPM}}, k_H^{\texttt{MPM}}\} \cup \{(k_A^{\texttt{GMP}})^j\}_{j \in \mathcal{I}}$;
   - $\mathcal{K}_3 = \mathcal{K}_1 \cup \mathcal{K}_2$;

The devices are now ready for deployment.

3.5. Communication Protocol

Let GD$_i$ be one of the deployed gathering devices for some $i \in \{1, \ldots, n\}$, let $f_{id} \in \mathbb{Z}_2 56$ be the unique identifier of GD$_i$ stored in its solid memory and consider the following abbreviations:

- $e_1^{IV} \equiv$ Encryption mode AES-CTR-128 with key $(k_A^{\texttt{GMP}})^i$ using IV as the initialization vector.
- $d_1^{IV} \equiv$ Decryption mode AES-CTR-128 with key $(k_A^{\texttt{GMP}})^i$ using IV as the initialization vector.

[1] As defined in section 3.1

- $e_2^{IV} \equiv$ Encryption mode AES-CTR-128 with key $k_A^{\texttt{MPM}}$ using IV as the initialization vector.
- $d_2^{IV} \equiv$ Decryption mode AES-CTR-128 with key $k_A^{\texttt{MPM}}$ using IV as the initialization vector.
- $h_1 \equiv$ HMAC-SHA-256 with key $(k_H^{\texttt{GMP}})^i$.
- $h_2 \equiv$ HMAC-SHA-256 with key $(k_H^{\texttt{GMD}})^i$.
- $h_3 \equiv$ HMAC-SHA-256 with key $k_H^{\texttt{MPM}}$.

3.5.1 Upstream Data Lifecycle

The operations that are carried out in each device will now be listed.

Gathering Devices

1. GD$_i$ transforms the gathered analog raw data to digital data *D* and assigns to it a 4-octet message identifier m_{id};
2. Compute $h_1(D)$, $h_2(D)$ and $inner\_pack := h_1(D) \parallel h_2(D) \parallel D$;
3. Generate IV_1 and compute $e_1^{IV_1}(inner\_pack)$;
4. $Pack_1 := f_{id} \parallel m_{id} \parallel IV_1 \parallel e_1^{IV_1}(inner\_pack)$;
5. Store $Pack_1$ in solid memory.

GD$_i$ sends the message subject to the WPA2-PSK protocol to MPP.

Middle-Point Party

6. Compute $d_1^{IV_1}(e_1^{IV_1}(inner\_pack))$ in order to obtain $inner\_pack$;
7. Verify D's integrity and authenticity. If the verification is unsuccessful, then consider the message at hand as compromised and abort at this step by clearing all the memory associated with it.
8. Send m_{id} as an acknowledgement[2] to GD$_i$;
9. $Pack_2 := f_{id} \parallel IV_1 \parallel h_1(D) \parallel h_2(D) \parallel e_1^{IV_1}(inner\_pack)$;
10. Generate IV_2 and compute $h_3(e_2^{IV_2}(Pack_2))$;
11. $Pack_3 := h_3(e_2^{IV_2}(Pack_2)) \parallel IV_2 \parallel e_2^{IV_2}(Pack_2)$;
12. Store $Pack_3$ in solid memory.

MPP transmits the data to MnDM subject to the TCP/IP protocol

[2]The message acknowledgement is subject to the WPA2-PSK protocol and thus is protected while travelling through the network. Upon receiving this information, the GD$_i$ will trust that this information has been successfully delivered to the intended party.

Mission and Data Manager

13. Verify the integrity and authenticity of the encrypted data. If successful, proceed to the next step, otherwise consider the message as compromised and abort at this step.
14. Compute $d_2^{IV_2}(e_2^{IV_2}(Pack_2))$ in order to obtain $Pack_2$;
15. Compute $d_1^{IV_1}(e_1^{IV_1}(inner\_pack))$ in order to obtain $inner\_pack$;
16. Verify D's integrity and authenticity. If successful send $Pack_3$ to the DB. Otherwise reject the message.

3.5.2 Downstream Data Lifecycle

The list of steps that are carried out in each device on the UDL is now presented.

Mission and Data Manager

1. Generate a message M and fix the target GD$_i$;
2. Compute $inner\_pack := h_2(M) \parallel M$
3. Generate IV_1 and compute $e_1^{IV_1}(inner\_pack)$;
4. $Pack_0 = h_3(e_1^{IV_1}(inner\_pack)) \parallel f_{id} \parallel IV_1 \parallel e_1^{IV_1}(inner\_pack)$.
5. Generate IV_2 and compute $e_2^{IV_2}(Pack_0)$;
6. $Pack_1 = IV_2 \parallel e_2^{IV_2}(Pack_0)$;
7. Store $Pack_1$ in solid memory.

$Pack_1$ is now sent via Internet to the MPP subject to the TCP/IP protocol.

Middle-Point Party

8. Compute $d_2^{IV_2}(e_2^{IV_2}(Pack_0))$ in order to obtain $Pack_0$;
9. Verify integrity and authenticity on $e_1^{IV_1}(inner\_pack)$;
10. Compute $d_1^{IV_1}(e_1^{IV_1}(inner\_pack))$ in order to obtain $inner\_pack$;
11. Build $inner\_pack_1 := flag \| h_1(M) \| h_2(M) \| M$ where $flag = 1$ and $|flag|_2 = 1$;
12. Generate IV_3, m_{id}^* and compute $e_1^{IV_3}(inner\_pack_1)$;
13. $Pack_2 := m_{id}^* \parallel IV_3 \parallel e_1^{IV_3}(inner\_pack_1)$;
14. Store $Pack_2$ in solid memory;

The message is sent to GD$_i$ subject to the WPA2-PSK protocol.

Gathering Devices

15. Compute $d_1^{IV}(e_1^{IV}(inner\_pack))$ in order to obtain $inner\_pack$;
16. Verify M's integrity and authentication.
17. Store M in the commands' FIFO stack, waiting to be applied as soon as possible. Successfully **exit** the downstream data protocol after applying the envisaged command.

3.6. Message Formats

$\mathcal{F}_1$: encrypted by a GD and decrypted by the MPP.

$\mathcal{F}_2$: encrypted by the MPP and decrypted by a GD (plaintext generated by the MPP).

$\mathcal{F}_3$: encrypted by the MPP and decrypted by a GD (plaintext originally generated by the MnDM).

$\mathcal{F}_4$: encrypted by the MPP and decrypted by the MnDM.

$\mathcal{F}_5$: encrypted by the MnDM and decrypted by the MPP.

All the abovementioned message formats are built based on two other category of message formats $\mathcal{F}_i^*$ and $\mathcal{F}_i^{**}$, which are, respectively, the formats correspondent to the outer and inner layers of encryption within $\mathcal{F}_i$ for every $i \in I_5$.

Definition 3.1 (Confidential plaintext)**.** *A confidential plaintext is any element from the set of the raw data gathered by the GDs or from the set of command messages.*

Let G be the ciphertext generator operator such that

$$G(\mathcal{C}, M, v, k, x) = y \tag{5}$$

where y is the result of encrypting x via block cipher $\mathcal{C}$ using mode of operation M with initialization vector v (if applicable) and key k. Analogously, G^{-1} is the inverse operator and returns the corresponding plaintext:

$$G^{-1}(\mathcal{C}, M, v, k, y) = x \tag{6}$$

Consider the set of all words with format $\mathcal{F}_i$,

$$S_i = \{y \in \Sigma^* : y \text{ is of format } \mathcal{F}_i\} \tag{7}$$

Moreover consider a family of functions

$$E_i : \mathcal{I} \times \Sigma^* \times \Sigma^* \times \mathcal{K} \times \mathcal{P} \to S_i \tag{8}$$

such that $E_i(j, a, b, k, x)$ represents the instance that outputs an element of S_i, for some confidential plaintext x, key k, GD's identifier j and global parameters a and b. This function is called of PCgF and hereinafter will be addressed accordingly. For a fixed key j, the function

$$E_i^j : \Sigma^* \times \Sigma^* \times \mathcal{K} \times \mathcal{P} \to S_i \tag{9}$$

is an instance function from the family of PCgF with the same expression.

Definition 3.2 (Package ciphertext)**.** *Let x be a confidential plaintext, $j \in \mathcal{I}$, k a key and $a, b \in \Sigma^*$ two parameters of choice. The data resulting from the computation of $E_i(j, a, b, k, x)$ is designated as package ciphertext.*

The inverse function for the PCgF is defined by

$$D_i : \mathcal{I} \times \mathcal{K} \times S_i \to \mathcal{P} \tag{10}$$

where $\mathcal{K}$ is the set of keys and $\mathcal{P}$ is the set of all confidential plaintexts and is defined of PCuF. This means that for every $z \in \mathcal{P}$:

$$D_i^j(k, E_i^j(a, b, k, z)) = z \tag{11}$$

for every $j \in \mathcal{I}$, where

$$D_i^j : \mathcal{K} \times S_i \to \mathcal{P} \tag{12}$$

is an instance of the family D_i of PCuF.

Definition 3.3 (Packing Scheme)**.** *A packing scheme (PSch) is defined by a 3-tuple (E, D, K) where E is a family of PCgF, D is a family of PCuF and K is the key set.*

Let $\mathcal{P}_{\mathcal{S}}^i$ be the PSch associated with message format $\mathcal{F}_i$ such that $\forall_{i \in I_5}$:

$$\begin{gathered} \mathcal{P}_{\mathcal{S}}^i = (E_i, D_i, K) \\ \text{and} \\ E_i = \bigcup_{j=1}^{n} E_i^j \text{ and } D_i = \bigcup_{j=1}^{n} D_i^j \end{gathered} \tag{13}$$

where E_i^j and D_i^j are as according to expressions 9 and 12, respectively. For every $i \in I_5$, the packing schemes $\mathcal{P}_{\mathcal{S}}^i$ define the distinct message formats.

4. Security Analysis

The network considered in section 3.2 has some ingrained fragilities induced by the chosen topology or by the communication protocol descripted in section 3.3.

In the key generation stage all the required cryptographic keys are generated and in order to increase the resilience of the key against key-recovery attacks it must be generated as randomly as possible. Based on the results presented in [7] PBKDF2 is a good option for the generation of the keys, namely with HMAC-SHA-1 because it

is the keyed-hash function with better performance and provides enough security [3], even though the inherent hash function is not strong with respect to collision resistance [13]. The password serves both as the seed for the pseudo-random generator that constructs the salt byte array as well as the password passed as argument to the PBKDF2 algorithm.

Proposition 4.1. *It is infeasible for an adversary to perform replay attacks or trick the MPP into assuming the gathering material is located elsewhere.*

Proof. Let $i \neq j$ and F_j and F_i be two gathering devices with identifiers d_{id}^j and d_{id}^i, and located at positions X and Y, respectively. Consider that an adversary wants to display malicious activities at location X, which would be detected by F_j. If he is aware of the presence of F_j, he might attempt to tamper with the reports on the location by changing d_{id}^j for d_{id}^i; this way the information transmitted to the MPP would be that the malicious activity is in effect in location Y instead of X. However, changing the GD identifier value to d_{id}^i will result in the MPP calling the PCUF of $\mathcal{P}_S^1$ using the keys shared with F_i, meaning that the resulting decrypted text would be distinct from the original plaintext, due to the injectivity of AES. Thus such an attack becomes infeasible since the adversary has a very thin margin of success. The resistance to replay attacks follows directly from the WPA2 protocol [1]. □

Due to simplicity purposes, consider $E_1^j(x)$ to represent the PCgF with omitted global parameters a and b and for which the keys used in the encryption scheme are associated with GD$_j$.

Proposition 4.2. *PSch $\mathcal{P}_S^i$ grants secrecy, integrity and authenticity to the confidential plaintext, for every $i \in I_5$.*

Proof. Let $\mathcal{A}$ be an adversary and $j \in \mathcal{I}$ a fixed identifier representing the operational gathering device GD$_j$. For some unknown confidential plaintext x suppose that $\mathcal{A}$ is in possession of $y := E_i^j(x)$, the associated package ciphertext.

- $i = 1$: The secrecy of the plaintext follows directly from the secrecy property of the AES-CTR encryption scheme; $\mathcal{A}$ cannot obtain x simply with the knowledge of y because the former is encrypted with AES-CTR using the key $(k_A^{\texttt{GMP}})^j \in \mathcal{K}_1$, which is of private knowledge uniquely to the MPP and GD$_j$.

  Based on figure **??**, recall that $E_1^j(x) = d_{id} \parallel m_{id} \| IV \| g(\mathsf{AES}, \mathsf{CTR}, IV, (k_A^{\texttt{GMP}})^j, w)$, where $w = h_1(x) \parallel h_2(x) \parallel x$. Clearly there is no integrity nor authenticity protection to the encrypted message and the malleability property of the AES-CTR encryption scheme gives the attacker an opportunity of tampering the ciphertext. In case $\mathcal{A}$ interferes with the ciphertext, then after decryption one ends up with the word $w^* = h_1(x)^* \| h_2(x)^* \| x^*$, due to the properties of AES-CTR. It is certainly very unlikely that $\forall_{i \in I_2} : h_i(x)^* = h_i(x^*)$ because of the avalanche property of cryptographic hash functions, which means that the adversary has a negligible probability of corrupting an encrypted message without compromising the plaintext's integrity check. The integrity and authenticity follows from the fact that the HMAC keys $(k_H^{\texttt{GMP}})^j$ and $(k_H^{\texttt{GMD}})^j$ are uniquely distributed among the pairs (GD$_j$, MPP) and (GD$_j$, MnDM), respectively.

- The same arguments can be applied analogously for $i = 2, \ldots, 5$.

□

Proposition 4.3 (Non-repudiation)**.** *The data stored in the DB is granted the non-repudiation property, provided complete trust on the user of the PMS.*

Proof. Let $\mathcal{U}$ be the user of the PMS, $\mathcal{J}$ the umpire, $\mathcal{V}$ the entity asking for the verification and y an arbitrary message stored at DB. In order for $\mathcal{J}$ to show $\mathcal{V}$ that y corresponds to a package ciphertext of some message that originated at the GD$_j$, he requires of $\mathcal{U}$ a simulation of the key generation stage. Because $\mathcal{U}$ is the only one with access to the password used at the process then, having access to the correct PCUF f, $\mathcal{V}$ just needs to execute the function f with the keys that were assigned to GD$_j$. If the output is a valid message then all the verifications for the unpacking procedure succeeded and $\mathcal{J}$ has proved to $\mathcal{V}$ that y is legitimate, as well as its secret contents. □

4.0.1 Semantic security

Proposition 4.4. *For every $i \in I_5$, the PSch $\mathcal{P}_S^i$ is semantically secure against chosen-plaintext attacks.*

Proof. The fields d_{id} and m_{id} within message format $\mathcal{F}_1$ are independent of the plaintext and so $\mathcal{A}$ cannot infer any relation with the associated plaintext. As has already been stated, the IV must be exposed as plaintext for the security of the AES-CTR encryption scheme and it does not leak any information for the adversary to exploit. Therefore, for $i = 1, 2, 3$ the semantic security of $\mathcal{P}_S^i$ follows from the IND-CPA security of AES-CTR proved in [4] and from the equivalence between IND-CPA and SEM-CPA proved in [10]. □

The previous proposition holds for plaintexts of equal length. However, it so happens that the confidential plaintexts are not all of the same size and no padding method is used in the packing schemes, thus POAs are infeasible and the process is more efficient but it means that the ciphertext transmits the plaintext's length information to the attacker. This unfortunate leak of information makes the cryptosystem vulnerable to chosen-plaintext attacks in which the adversary is able to pick plaintexts of non-equal length.

Proposition 4.5. *For every $i \in I_5$, PSch $\mathcal{P}_{\mathcal{S}}^i$ is not semantically secure against chosen-ciphertext attacks.*

Proof. Let $\mathcal{A}$ be an adversary playing the IND-CCA game [4] with the words $w_0 = 0^n$ and $w_1 = 1^n$ for some $n \in \mathbb{N}$ and w.l.o.g. fix $j \in \mathcal{I}$. Let $\mathcal{O}_e$ and $\mathcal{O}_d$ be the oracles with access to the PCgF and PCuF of $\mathcal{P}_{\mathcal{S}}^1$. The following strategy grants $\mathcal{A}$ a non-negligible IND-CCA advantage:

1. $\mathcal{A}$ queries $\mathcal{O}_e$ with (w_0, w_1);
2. $\mathcal{O}_e$ encrypts w_b into $y := E_1^j(w_b)$, for $b \in \mathbb{Z}_2$ and returns it to $\mathcal{A}$;
3. $\mathcal{A}$ flips the last bit of y and obtains y';
4. $\mathcal{A}$ queries $\mathcal{O}_d$ with y';
5. $\mathcal{O}_d$ returns w_b';
6. If $w_b' = 0^{n-1}1$ then $\mathcal{A}$ chooses $b' = 0$. If $w_b' = 1^{n-1}0$ then $\mathcal{A}$ chooses $b' = 1$.

The last step is only feasible due to the properties of error propagation of the CTR mode of operation [11]. Thus, $\mathcal{A}$ can tell to which confidential plaintext belongs the package ciphertext with probability far from $1/2$, meaning that $\mathcal{P}_{\mathcal{S}}^1$ is not IND-CCA secure. The result follows from the equivalence between semantic security and ciphertext indistinguishability in [15].

For $i = 2, 3$ and 5 the proof is analogous, with a single remark for the case $i = 5$, in which there are two layers of encryption on the confidential plaintext but the result is the same as in the previous cases due to the direct error transmission property of CTR.

For $i = 4$ one faces an encrypt-then-MAC procedure which is the only layout susceptible to be IND-CCA secure. However, the fact that the IV is not targeted by the HMAC entails that $\mathcal{P}_{\mathcal{S}}^4$ is also IND-CPA insecure. In fact, note that if an adversary has access to a decryption oracle and the HMAC does not include the IV then the adversary can change the value of the IV at will in order to claim the keystream for the new IV. Hence $\mathcal{A}$ will be able to decrypt any message that was encrypted using any of these new IVs. □

If an encrypt-then-MAC mechanism was adopted and the header of the message was targeted by the HMAC the previous result would not hold.

4.0.2 Encryption Schemes

In order to make use of asymmetric cryptosystems in the development of digital signatures there is the need for a CA whose role is to evaluate the authenticity of the messages travelling throughout the nodes of the network. The need for a CA immediately deprecates the use of asymmetric cryptography because it is required that it provides all the demanding certificates on the fly and the chosen network topology does not include any permanent party on the field other than the GD. Moreover the overhead in terms of memory entailed by the usage of asymmetric cryptographic systems and the increased key size makes these types of systems not suitable under the current restrictions, either for authentication or privacy purposes.

AES was the chosen cipher to provide secrecy to the confidential messages. Since this is a very fast algorithm and a it is hardware-implemented in the GD makes it the most suitable choice for the case. CTR was the chosen mode of operation, since ECB is not semantically secure, CBC requires an unpredictable IV and CFB is extremely vulnerable to transmission errors and has a non-parallelizable encryption.

CTR-H mode was chosen over is CCM and GCM because its run-time execution is faster due to the fact that it is implemented in the chosen device's hardware, opposite to the latter; Moreover it uses an extra key (the authenticity and privacy keys are distinct) thus it is a high resilient system against brute-force attacks.

4.1. Attacks

Let $\mathcal{A}$ be a polynomial-time bounded adversary with physical access to the GD. By propositions 4.2 and 4.4 $\mathcal{A}$ cannot directly retrieve the secret data gathered by GD. Moreover, the adversary will not be able to retrieve the keys from memory because these are stored in a memory location of restricted access [6]. Now assume that the GD are deployed at time 0 and that at time t there is an event $\mathbb{E}$ which will directly or indirectly provide information to be gathered by some GD. The latter will get to know this information upon collecting the data within the time interval $T = [t, t + \epsilon[$, where $\epsilon > 0$ is the duration of the intelligence leak of the event $\mathbb{E}$. Suppose also that $\mathcal{A}$ is aware of both $\mathbb{E}$ and ϵ. Then, the adversary just needs to interfere with the envisaged GD in the time interval T in order to prevent them from collecting any of the relevant data. Thus, for an intelligent adversary who is able to physically

interfere with the GD, this technique is less likely to be spotted by a tamper detection mechanism, while allowing $\mathcal{A}$ to optimize his/her energy consumption on the attack.

Another attack that could be performed is the reading of volatile memory [2] by a capable adversary. No message is ever stored as plaintext in solid memory, but both before encryption and after decryption, the confidential plaintext is automatically stored in volatile memory, even if for a short time frame. Nonetheless, this time frame may suffice for the adversary to harvest the secret information.

Assuming that $\mathcal{A}$ cannot physically harm or interfere with the GD a straightforward attack would be for the adversary to perform a MiM attack known as wireless denial of service attack that consists in jamming the communication channel with junk data, making the exchange of data between the GD and the MPP impossible, provided that he finds the correct frequency.

4.2. Possible Solutions

A reasonable approach to prevent an attack where the adversary takes advantage on the physical exposure of the device is to choose a device whose hardware provides a tamper detection mechanism and memory management [8] such that in case of memory compromise it clears all the memory associated with the confidential data. This is a last resource and leads to the disablement of the device at hand. With respect to the jamming attack on the wireless communication channel, there is no way to prevent it from happening. The only solution would be to wire the connection, where the adversary would be unable to jam the channel without being targeted by the MPP.

4.2.1 Chosen-plaintext attack

Let $x_1, \ldots, x_k$ be confidential plaintexts for some $k \in \mathbb{N}$ and $y_1, \ldots, y_k$ their correspondent package ciphertexts, respectively. Recall that $\forall_{i \in I_5} : \mathcal{P}_{\mathcal{S}}^i$ is not secure against chosen-plaintext attacks in which the adversary is able to pick plaintexts of distinct length. The length L_j of confidential plaintext x_j is a multiple of a minimum length l, i.e.,

$$L_j = lm_j \ \forall_{j \in I_k} \tag{14}$$

Furthermore, let h be the length of the header of a package ciphertext, which is assumed to be fixed for any PSch, for simplicity purposes. If one does not consider the header's overhead, then it is equivalent to partition each message of length L_j into m_j messages of length l and the system would become resistant to variable-length chosen-plaintext attacks. This is uniquely a theoretical solution for this problem because the assumption of the absence of the header does not hold in practice. The only way to overcome this issue is to define all confidential plaintexts to have the same length.

4.2.2 Chosen-ciphertext attack

Proposition 4.5 refers to the fragility of the PSch $\mathcal{P}_{\mathcal{S}}^i$ with respect to chosen-ciphertext attacks. The previous result holds under the assumption that the adversary $\mathcal{A}$ has access to a decryption oracle. However, in practice there is no feasible way for $\mathcal{A}$ to be admitted such resources. The only way $\mathcal{A}$ would be able to succeed would be to impersonate an element of the network, use the correct[3] PCGF and then send it to the end party and be able to extract the plaintext from its volatile memory at the moment of decryption. This approach is infeasible in practice because no element of the network can be impersonated by a polynomial-time bounded adversary, as shown in proposition 4.2.

5. Results

The study, decisions and analysis of this specific network were performed under the supervision of analysts and developers of the company GMVIS Skysoft, S.A..

The keys' generation process is very reliable in the sense that it is not only performed within secured headquarters but also a very efficient method regarding security and time. More specifically, it is a linear-time process with respect to the number of gathering devices that are to be deployed.

The selected network topology is considered to be the one that better suits the practical needs of the mission, whilst in theory an ad-hoc network might have a better performance when combined with elliptic curves [14].

The set of chosen packing schemes is considered to be a robust and secure option for the case, but would achieve a higher level of security if adopting an encrypt-then-MAC method of encryption and authentication with addition to including the header in the input to the HMAC; this approach would assure the system to be IND-CCA secure. However, even though it would strengthen the theoretical level of security, it would have no impact in practice because the variable size of the plaintexts induce an inexorable fragility. As for the encryption scheme, AES-GCM should be preferred over AES-CTR-HMAC in order to grant authenticity, integrity and privacy to the plaintexts in a theoretical point of view. The former is underqualified simply because it is not implemented in the hardware of this particular type of devices. Would any other devices

[3] By correct one means the correct function using the correct keys.

with distinct characteristics have been chosen, the outcome would certainly differ from the one presented. All packing schemes are vulnerable to chosen ciphertext attacks which is a fact of some concern because an attacker with access to a decryption oracle might be able to break the system, even if just partially. There is virtually no way of preventing an adversary of performing a lunchtime attack [12] when the devices are in sleep mode.

All the data processing methods are linear in the size of the input. Given the GDs' memory limitation, the size of the confidential plaintexts generated by these elements is upper bounded by some constant value, which implies that the running time of the previously mentioned data processing algorithms is also upper bounded due to this constraint, for their time complexity is $\mathcal{O}(n)$. Thus, in practice, these algorithms are time-efficient.

5.1. Future Work

One possible improvement to the amplitude of the given network would be to allow the parallel activity of more than one MPP. This would require more keys to be generated not only for privacy and integrity purposes on the data, but such that all the MPPs are uniquely recognizable by the network parties (that is, provide an authentication mechanism).

Another subject with good prospects is the hardware improvement of the devices such that their capabilities allow more efficient and secure packing schemes. By efficient one means both in terms of time and space complexity. A good example is to implement in hardware a randomized primitive that makes use of analogue entropy sources in order to obtain fairly randomized values. This feature would be extremely useful for the IV scheduler within the GDs and, in the event of increasing the devices' battery lifetime, it would also be very fruitful for the development and maintenance of a key scheduler algorithm.

In addition, a potential improvement would be to implement in hardware standardized authenticated modes of operation such as GCM. Thus, adopting AES-GCM instead of AES-CTR-H would optimize the system's memory usage and therefore allow the GDs to be able to store more messages as well as shorten the GDs' sleep mode time-frame.

References

[1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements. *IEEE Std. 802.11i™-2004*, July 2004.

[2] K. Amari. *Techniques and Tools for Recovering and Analyzing Data from Volatile Memory*. March 2009.

[3] Bellare, Mihir. *New Proofs for NMAC and HMAC: Security without Collision-Resistance*. June 2006.

[4] Bellare, Mihir and Rogaway, Phillip. *Course Notes: Introduction to Modern Cryptography*. University of California, San Diego.

[5] G. Bertrand. *Enigma: ou, La plus grande énigme de la guerre 1939-1945*. Plon, 1973.

[6] D. Cook. Measuring memory protection. In *Proceedings of the 3rd International Conference on Software Engineering*, ICSE '78, pages 281–287, Piscataway, NJ, USA, 1978. IEEE Press.

[7] Ertaul, Levent and Kaur, Manpreet and Gudise, V. A. K. R . *Implementation and Performance Analysis of PBKDF2, Bcrypt, Scrypt Algorithms*.

[8] Grand, Joe. Practical Secure Hardware Design for Embedded Systems. In *Proceedings of the 2004 Embedded Systems Conference*. CMP Media, April 2004.

[9] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer New York, 2012.

[10] A. Nascimento and P. Barreto. *Information Theoretic Security: 9th International Conference, ICITS 2016, Tacoma, WA, USA, August 9-12, 2016, Revised Selected Papers*. Lecture Notes in Computer Science. Springer International Publishing, 2016.

[11] National Institute of Standards and Technology. *Recommendation for Block Cipher Modes of Operation*. National Institute of Standards and Technology, Gaithersburg, MD, USA, 2001.

[12] P. Rogaway, D. Pointcheval, A. Desai, and M. Bellare. *Relations Among Notions of Security for Public-Key Encryption Schemes*. June 2001.

[13] Stevens, Marc and Bursztein, Elie and Albertini, Ange and Markov, Yaric. *The first collision for full SHA-1*. 2017.

[14] D. Stinson. Elliptic curves. In *Cryptography: Theory and Practice,Third Edition*, pages 254–266. CRC/C&H, 2005.

[15] Y. Watanabe, J. Shikata, and H. Imai. *Equivalence between Semantic Security and Indistinguishability against Chosen Ciphertext Attacks*. RIKEN Brain Science Institute, 2003.