A new approach on IoT security: n-out-of-n case

Tuğberk KOCATEKİN[1] and Cafer ÇALIŞKAN[2]

[1] Istanbul Arel University, Istanbul, Turkey, tugberkkocatekin@arel.edu.tr
[2] Antalya Bilim University, Antalya, Turkey, cafercaliskan@gmail.com

Abstract. Internet of Things (IoT) has become an established part of our daily lives by interconnecting billions of devices in diverse areas such as health care, smart home technologies, agriculture, etc. However, IoT devices are limited in memory, energy and computational capabilities. This creates a great potential for security issues, since being constrained prevents producers from implementing mostly complex cryptographic algorithms in IoT devices. In this study, we propose a novel method to provide a low-cost and secure communication for constrained IoT devices. The proposed method is based on an n-out-of-n secret sharing scheme and mimicks the idea of visual cryptography in a digital setup. Whenever an IoT device communicates with an outer party, it establishes the communication by itself or through a mediary such as a central hub or gateway; in which the latter mostly leads to a single point of failure. Our proposed method aims for a distributed environment in which IoT devices within a secure network collaborate with each other in order to send a message to a master device over an insecure channel.

Keywords: Internet of Things, Secret Sharing Scheme, Visual Cryptography

1 Introduction

Web technologies improved dramatically in recent years. This improvement started from simple HTML web pages to Web 2.0 with social networks, online applications, wikis; which became indispensable for our daily and business lives. However, a new web technology has emerged, namely Web 3.0, which is also referred as Semantic Web. It aims to mark-up content in a standardized way to make it possible for machines to understand content without human interaction. As these recent developments come together by combining sensor networks and near field communication technologies, it leads to a new technology/area called Internet of Things (IoT) [1]. With this, IoT not only aims to provide connection among computers and mobile phones, it also aims to create a collective network combining automobiles, buildings, cities and even electrical grids. However, as the number of IoT devices are increasing tremendously, resolutions of the issues surrounding IoT devices gain more importance over time. An online survey conducted by Intel [2] shows that among 2500 adults in United States, 68 percent of them believe that smart-homes will be as common as smart-phones by 2025. This indicates that there is a positive attitude towards smart-housing. However, it is also critical to state that same survey shows that 82 percent of those people agree that integrated security is a priority. Moreover, there already exist several attacks on real-life uses of IoT devices. Some of the celebrated ones are the botnet attacks on huge number of smart home appliances such as IP cameras, DVR's, etc. and this attack even caused internet outages [4].

Another type of attack takes advantage of single point of failure as an attacker gains control of the entire network after compromising just a single IoT device. A typical IoT device mostly lacks of comprehensive control over the received information and usually follows simple protocols such as HTTP requests. Then, a legitimate question is how devices

in an IoT network can achieve a basic task such as sending a message without a central unit in a secure way. One approach is to let the devices in the network take active roles instead of depending on a central unit. This could also maintain an environment in which devices will be aware of the action of sending information to an outer source. But this requires a collective effort in which the sender proves his identity by following certain protocols with other devices and these protocols cannot be power-hungry or require much memory. This is because IoT devices are low-powered devices and it is hard to implement advanced cryptographic algorithms on them. Moreover, every IoT device in a network may not have similar capabilities in a heterogeneous structure, so a potential solution should consider all these limitations.

This study proposes a new approach by employing visual secret sharing techniques on maintaining a secure communication between IoT devices in a network and a master device. It also aims to resolve some authentication related issues in an IoT setup. This resolution mainly ignores centralization (as missing a central unit) and assigns active roles to IoT devices within their capabilities, therefore avoids the potential of experiencing single point of failure.

1.1 History and related works

Single point of failure is defined as a part of system which renders the entire system useless when occurs. There are well-known examples in real-life some of which has also taken place in literature in which attackers use this issue to compromise a network [5, 6, 7, 8, 9, 10, 11]. For instance, automobiles have become filled with electronic components in recent years. Although they provide some benefits to efficiency and safety, it also brings potential risks. Several studies [5, 6, 7, 8] evaluated them and shown how the cars and drivers are vulnerable to different attacks.

Modern cars are controlled by a combination of digital components called Electronic Control Units (ECUs). These ECUs are interconnected by internal wired networks such as CANBUS and FlexRay bus, and they control a broad range of functionality in a car [5]. However, this also leads to a broad internal attack surface since the internal network connects these devices. Therefore, a single compromised device can risk the whole network. For instance, Rouf et al. [9] studied tire pressure monitoring system (TPMS) in cars which uses radio frequency to transmit the data to pressure control unit, which analyzes this data and sends it to a central computer via CAN bus. After their investigation, they observed that the central computer trusts the data without any authentication. Then, by reverse engineering the process, they were able to disable TPMS and activate warning lights on a car.

Koscher et al. [10] studied whether they could have access to an automobile remotely. They showed that remote access to a car was possible via a broad range of attack vectors varying from CD players and radios to wireless communications. It was even possible to remotely track the car and take control of it. When they investigated the short-range wireless channel using Bluetooth technology, they found several non-safe function calls. After exploiting these calls, they could take control of the ECU. They could also use a specially encoded audio file to compromise the car just by using the built-in cellular connection.

IoT devices are popular as attacking points as they can be used as entry points to a network. Without attacking the network itself, if one can take control of an IoT device, they can take control of the entire network. Even if a network is considered as secure, it is important to remember that a chain is as strong as its weakest link. Similarly, IoT devices may be considered as the weakest components in a network.

Another security issue is related to commercial products that a user can easily connect to the Internet without any security assumption. Majority of users do not have background in technology or security to become aware of potential vulnerabilities. As a result, they

do not check whether there is a secure channel for communication among devices or even a secure password policy to maintain security. There are several examples showing that commercial products may provide an open door to the networks. For instance, Vectra Threat Labs [1] successfully established an access into a network just by using a cheap consumer grade webcam which was reprogrammed as a network backdoor while operating as a camera. It is also important to mention that these type of attacks may become even easier with second-hand items and take part in organized crime and espionage.

Ronen and Shamir [11] attacked two commercial connected lighting systems, namely *Philips Lux* and *LimitlessLED*, which mainly control the color and intensity of lights. Authors tried to achieve a different effect rather than the original functionality of these devices. Firstly, they extracted data from a secure location by creating a covert LI-FI communication system by using smart lights and were able to read data over 100 meters. Afterwards, they showed that an attacker is able to strobe lights at a frequency which may trigger seizures in people with photosensitive epilepsy.

There are numerous examples of security issues originated from IoT devices and their poor security assumptions. Our goal is to provide a new approach ignoring a centralized set-up and therefore avoid a single point of failure. In this, IoT devices can take active roles despite the fact that they are constrained.

2 Preliminaries

Visual Cryptography (VC) was first introduced in 1994 by Naor and Shamir [12] as a novel way to provide secrecy on written material (printed text, notes, images, etc.) without using any complex cryptographic computation. In this, encryption is done by splitting an image into a certain number of shares, let's say n shares, and printing them onto transparent films. Then, in order to re-construct the original image, k or more, where $k \leq n$, transparent images must be stacked on top of each other. Otherwise, it is not possible to obtain the image back. This is called a k-out-of-n visual secret sharing scheme. Figure 1 provides a basic example of 2-out-of-2 scheme. It can be seen that shares alone do not reveal any information about the original image.

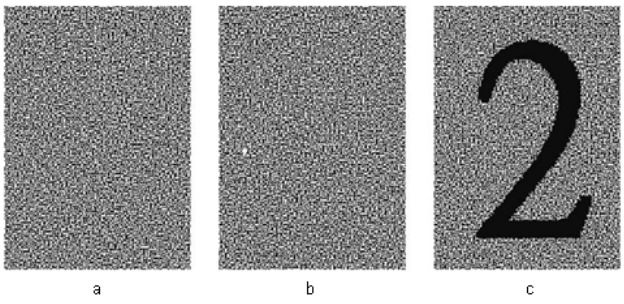

Figure 1: a) Share 1, b) Share 2, c) Reconstructing image by stacking shares up.

In the simplest version of visual secret sharing scheme, a message is a collection of black and white pixels and each black/white pixel is handled separately. Moreover, each pixel appears in n different versions, called *shares*, and each share is a collection of m black and white subpixels. Now, consider an n-by-m binary matrix $S = (s_{ij})$, where $s_{ij} = 1$ if the j^{th} subpixel on i^{th} share is black. Therefore, each row in S represents a

[1] Vectra Thread Labs (2016). Turning a webcam into a backdoor. [online]. Website: web.archive.org/web/20180805134121/https://blog.vectra.ai/blog/turning-a-webcam-into-a-backdoor [accessed 5 April 2022]

different share in the scheme. If one is given two shares, i.e. two rows of S, stacking them together means applying "inclusive OR" operation on these rows. Then the grey level of the resulting stacked vector is determined by the count of 1 bits. This count is basically the Hamming weight $H(v)$, where v is the resulting stacked vector. If this count is at least a fixed threshold value d, then the pixel obtained is assumed to be black. Otherwise, if $H(v) \leq d - \alpha \cdot m$, where $\alpha > 0$ is the relative difference in weight between combined shares from white and black pixels, then it is assumed to be white. In general, the matrix S represents a single pixel and rows of this matrix represent the shares which are to be distributed to n shareholders. After a single pixel (black or white) is splitted into shares, it is reconstructed as follows: First, k (out of n) rows of S come together and then their joint grey level is computed by counting the 1 bits after "inclusive OR" is applied on these k rows. If this number is less than the threshold value then the pixel is considered as white, otherwise as black. This process requires a selection of a proper matrix S depending on the pixel being black or white. Also, what makes this process interesting and useful is that any randomly chosen k (or more) rows give rise to the same outcome and any $k-1$ (or less) rows do not reveal any info on the grey level of the pixel.

Naor and Shamir [12] present some solutions on how to construct these matrices in their original article. They first consider the case where $k = n$. By adopting their notations, let us consider two sets of binary vectors each of which is of length n, namely $J_1^0, J_2^0, \ldots, J_n^0$ and $J_1^1, J_2^1, \ldots, J_n^1$. Now, it is assumed further that the vectors J_i^0 satisfy the property that any $n-1$ of them are linearly independent, but the entire set is linearly dependent. On the other side, the vectors J_i^1 are linearly independent. For any $n \geq 2$, such sets of vectors can be constructed by obtaining the following format:

$$\begin{array}{lcllcl} J_1^0 & : & 1000\ldots00 & J_1^1 & : & 1000\ldots00 \\ J_2^0 & : & 0100\ldots00 & J_2^1 & : & 0100\ldots00 \\ & & \vdots & & & \vdots \\ J_{n-1}^0 & : & 0000\ldots10 & J_{n-1}^1 & : & 0000\ldots10 \\ J_n^0 & : & 1111\ldots10 & J_n^1 & : & 0000\ldots01 \end{array}$$

Then the matrix S^0 (S^1) is constructed as follows: Let's first label the rows of the matrix with the vectors J_i^0 (J_i^1) and the columns with all possible binary vectors of length n, therefore the resulting matrix is of size n by 2^n. Then, the binary entry at the intersection of the i^{th} row and s^{th} column is computed from the inner product of the vector J_i^0 (J_i^1) and the binary vector labeling the s^{th} column.

Once these matrices S^0 and S^1 are constructed, then two collections C^0 and C^1 are produced from these matrices, respectively, by applying all possible permutations on the columns of S^0 and S^1. If one plans on transmitting a 0 bit (or 1 bit), then a random matrix from the collection C^0 (or C^1) is picked and its rows are distributed as the shares. There are two cases to consider:

(i) One picks a random matrix $S \in C^0$. Since all the columns of S are labelled by all possible binary vectors of length n, there are two columns labeled by all zero vector and the vector $0\ldots01$. Then these are the only columns (out of 2^n columns) with all zero entries. Therefore, the number of 1 bit entries is $2^n - 2$ when the rows of S are stacked together.

(ii) One picks a random matrix $S' \in C^1$. It is similar to the first case, but there is only one column with all zero entries which is labeled by the zero vector. Therefore, the number of 1 bit entries is $2^n - 1$ in this case.

In both matrices, whenever a smaller number of rows are stacked together, the outcome would have $2^n - 2$ many 1 bit entries and the only time the difference occurs is when

all rows of S' are stacked together. Hence, capturing $n-1$ rows does not reveal any information about the grey level and consequently not reveal whether the rows belong to S or S'. It requires capturing all n rows to determine the color of the pixel. This is an illustration of n-out-of-n scheme. See page 6 on the original article by Naor and Shamir [12] for further details on the proof of this scheme being an n-out-of-n scheme.

Naor and Shamir [12] also came up with an alternative but slightly better way of constructing an n-out-of-n scheme with $m = 2^{n-1}$ columns. Let $W = \{w_1, w_2, \ldots, w_n\}$ be a set with n elements, then assume that $E_1, E_2, \ldots, E_{2^{n-1}}$ and $O_1, O_2, \ldots, O_{2^{n-1}}$ are the subsets of even and odd cardinalities, respectively. Afterwards, n by 2^{n-1} matrices S^0 and S^1 are constructed as follows: For $1 \leq i \leq n$ and $1 \leq s \leq 2^{n-1}$, $S^0[i,j] = 1$ if and only if $w_i \in E_s$ and $S^1[i,j] = 1$ if and only if $w_i \in O_s$. Then, the collections C^0 and C^1 are obtained by permuting all the columns of S^0 and S^1, respectively.

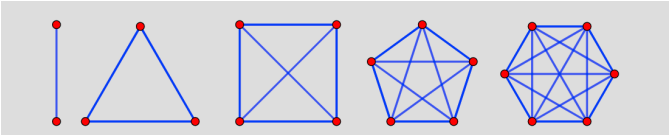

Figure 2: Complete graphs of order 2, 3, 4, 5 and 6.

In the proposed approach, finite complete graphs are used to define a topology on the network of IoT devices. Although it is assumed that the reader is familiar with basic notions in graph theory, some basic definitions in graphs are provided briefly. A graph is called *regular* if every node has the same number of neighbors. If the number of neighbors for each node is r, this graph is called r- regular. If a connected graph of order n is $(n-1)$-regular, then this graph is called a *complete* graph. See Figure 2 for some examples of complete graphs. Graphs are commonly used to define topologies on networks and there are already some criteria used when a topology is defined on a network. According to Nielsen [13], a network topology is expected to minimize the degree for hardware costs as well as the diameter for short paths for communication and maximize the network dimension for scalability. In this study, we adapt network topologies to determine the connectedness in terms of the ability of communication among the devices.

3 Model

Let us now consider a secure closed network $\mathcal{N}$ of n ($n > 2$) IoT devices and denote these devices in the network by $d_1, d_2, \ldots, d_n$. The main task is to send a message of a device d_i to a remote master device over an insecure channel in a secure way.

Assume that devices can communicate with each other pairwise in $\mathcal{N}$. Moreover, each of these devices has connection to the Internet individually through a dummy router. It also means that there is no central hub which processes and transmits messages collected from IoT devices to the master device on the behalf of IoT devices. Instead, the messages are sent to the master device by the IoT devices themselves in a collaborative environment. When the master device is near $\mathcal{N}$, it can communicate securely with each of the IoT devices within $\mathcal{N}$ and distributes their shares. But when it is away from the network, the communication takes place over an insecure channel. Whenever an IoT device d needs to send a message to the master device, it first triggers a communication with other IoT devices in $\mathcal{N}$ and then these devices, other than d itself, use their own (previously distributed) shares to transmit the message of d to the master device. Note that in this model the transmission of messages to the master device over insecure channel is assumed to be in only one direction, that is, the master device does not send messages back to IoT devices through this insecure channel.

Let us also assume that messages of the IoT devices are assumed to be predetermined and an enumeration has been applied to these messages. For instance, consider a message corresponding to a overheating issue belonging to a device, then a certain number is assigned to these type of issues. Whenever this issue occurs, a bitstring corresponding to the assigned number is transmitted to the master device through an insecure channel. This tranmission is done bitwise in a collaborative way by the IoT devices. Let us assume that each device d_i has a fixed number t_i of predetermined messages that are enumerated by numbers 0 through $t_i - 1$. If $t_{\max} = \text{Maximum}\{t_1, \ldots, t_n\}$, then each message can be represented by a bitstring of length $s = \lceil log\ t_{\max} \rceil$ by using a proper padding on shorter bitstrings. The above scenario has been considered with predetermined messages due to the fact the model is built for constrained IoT devices. Otherwise, the model can also support the dynamic messages of fixed length in a similar setup.

Now consider a master device near $\mathcal{N}$ which can communicate with each of the IoT devices securely, so this master device can distribute same number of shares to each device. Then devices become ready for sending a message collaboratively to the master device through an insecure channel. Note that the master device is near $\mathcal{N}$ on regular basis, so it can frequently communicate with the devices in $\mathcal{N}$ and also that IoT devices send their messages to this master device over an insecure channel when it is away. Figure 3 gives a visual representation of the defined model.

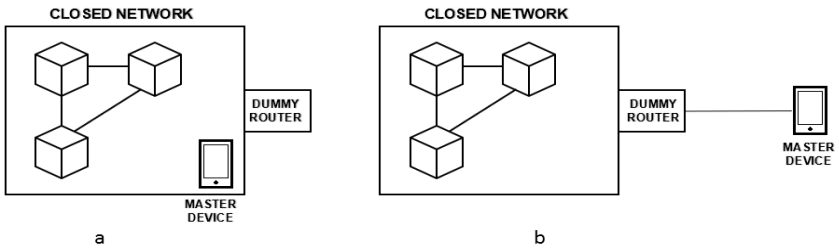

Figure 3: Illustration of a single master device a) near case b) away case

In what follows, we discuss the model from the IoT devices' perspective.

We first assume a network topology for $\mathcal{N}$ that is based on a complete graph $\mathcal{G}$ of order n in which the IoT devices are represented by the nodes in $\mathcal{G}$ and the adjacency relation defines a neighborhood among the devices. Since $\mathcal{G}$ is complete, any node is a neighbor of any other. This neighborhood is used in our model whenever a device transmits a message to the master device. In this process, only the neighboring devices collaborate with each other in sending the messages. This means the entire network collaborates when a complete graph is adapted as a topology i.e. all $n - 1$ neighboring nodes collaborate to send a message of a particular node. Whenever a device plans on sending a message to a remote master device, all its neighbors are informed about the message, so they take action accordingly and also are aware of the messages contentwise and countwise.

Before devices start sending messages, the master device distributes their shares when it is near $\mathcal{N}$. For sending a bitstring of length s, each participating IoT device uses s shares in total by consuming exactly one share for every bit of the message. However, since it is not known in advance which bitstring will be sent, the master device distributes $2 \cdot s$ shares to each device by considering all possible messages of length s. We assume that the master device predicts the number of messages sent by the IoT devices until next scheduled share distribution process, so it distributes same and enough number of shares accordingly. As part of share distribution, the master device first constructs S^0, S^1 matrices by following the construction steps described above and then obtains the corresponding collections C^0, C^1 by permuting the columns of these matrices. For a candidate message of length s, the

master device needs to pick s pairs of random matrices from $C_0 \times C_1$ with the condition that matrices in a pair differ from each other by at least two rows. Note that each matrix has exactly n rows and a row is considered as a share to be distributed to a device. That means, from a single pair of matrices, the master device distributes $2 \cdot n$ shares in total as each device receives exactly two shares, one for sending a 0-bit and the other for a 1-bit. When it comes to sending its shares, a device uses only one of these shares depending on the message bit. For instance, let us assume that, for the j^{th} bit of a potential message, the rows of M_j^0 and M_j^1 have been distributed to the IoT devices. Then, in the actual message, each device uses their shares belonging to M_j^0 if the j^{th} bit is a 0-bit and shares belonging to M_j^1 if the j^{th} bit is a 1-bit.

Now, let us assume that the master device is away after share distribution is completed, and an IoT device d plans on sending the message $m = m_1 m_2 \dots m_s \in \{0,1\}^s$ to the master device over an insecure channel. We denote the neighboring devices of d by $d_1, d_2, \dots, d_{n-1}$. Then, d broadcasts the first bit of m, namely m_1, to all its neighbors. Upon receiving m_1, each d_i itself sends its current share that is a row of a matrix M_1^0 if $m_1 = 0$ or M_1^1 if $m_1 = 1$ to the master device through a dummy router over an insecure channel. However, the device d does not send its share to the master device, so, for every bit of the message m, there are exactly $n-1$ shares, i.e. $n-1$ rows of a particular matrix, sent to the master devices over the insecure channel. This process continues similarly for m_2, m_3 and so on until the shares for m_s are all sent to the master device. Upon receiving the shares in the same order that they are sent for the message m, the master device checks whether these shares belong to the matrices M_j^0 or M_j^1 $(1 \le j \le s)$ and it concurrently determines the sender device which initiated the communication, since the master device stores all matrices, keeps the records on how it distributed the shares and the share of d was not sent to the master device. Note that determination of the matrix per bit is equivalent to determination of the bit value. The master device does not face any ambiguity in this process, since matrices in a pair differ from each other by at least two rows and the master device receives exactly $n-1$ rows per bit. By receiving all bits of m, the master device can reconstruct the message sent by d.

To illustrate the model, we present the following example with some small-sized matrices. Let's construct the matrices S^0 and S^1 and generate the collections C^0 and C^1 as follows:

$$S^0\colon \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \xrightarrow[\text{columns}]{\text{permuting}} C^0\colon \left\{ \dots, \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \dots \right\}$$

$$S^1\colon \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \xrightarrow[\text{columns}]{\text{permuting}} C^1\colon \left\{ \dots, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \dots \right\}$$

We pick the following pairs of matrices from the collections C^0 and C^1, then without loss of generality the first rows of matrices are all distributed to device d_1, the second rows to device d_2 and third rows to device d_3.

$$M_1^0 : M_1^1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} : \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

$$M_2^0 : M_2^1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} : \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

$$M_3^0 : M_3^1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} : \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Now, let's say d_2 plans on sending the message $m = m_1 m_2 m_3 = 001$, then devices d_1 and d_3 send the following shares to the master device in the order given below:

$$d_1 : \begin{bmatrix}0 & 0 & 1 & 1\end{bmatrix}_{m_1} \begin{bmatrix}0 & 0 & 1 & 1\end{bmatrix}_{m_2} \begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix}_{m_3}$$
$$d_3 : \begin{bmatrix}0 & 1 & 1 & 0\end{bmatrix}_{m_1} \begin{bmatrix}0 & 1 & 0 & 1\end{bmatrix}_{m_2} \begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}_{m_3}$$

When the master device receives the shares $\begin{bmatrix}0 & 0 & 1 & 1\end{bmatrix}_{m_1}$ and $\begin{bmatrix}0 & 1 & 1 & 0\end{bmatrix}_{m_1}$, then it can compute the bit value of m_1 as follows: Firstly, it already stores the pairs of matrices in the order the shares have been distributed, so it has the matrices M_1^0 and M_1^1. It also knows that the shares sent for m_1 are from only one of these matrices. Moreover, these two matrices differ from each other by at least two rows, so the master device can locate the shares as the first and third rows of M_1^0. This implies that $m_1 = 0$. Also, the master device can detect that the shares have been sent by d_1 and d_3, so this leads to the conclusion that d_2 has not sent its share so the message belongs to d_2. A similar process is conducted for m_2 and m_3 and the master device finally obtains the message as $m = 001$.

4 Implementation of the model and its security

Although the proposed model is presented from a theoretical perspective, we also discuss briefly various aspects of the implementation of the proposed model and its security in this section.

In the proposed model, we adopt a complete graph as the topology over the IoT network $\mathcal{N}$ and this topology determines connectivity of the devices corresponding to adjacency of the nodes on graph. Since the graph is complete, any device can communicate with any other in $\mathcal{N}$. In the proposed model, the main focus is to transmit messages collaboratively in a secure way over an insecure channel, so the communication among the devices within $\mathcal{N}$ is assumed to be pairwise and established securely. Whenever a device plans on sending a message, it triggers the pairwise communications by sharing its message with others. This creates an environment in which any device is aware of any communication initiated in $\mathcal{N}$. Moreover, devices are all aware of which messages are to be transmitted and in which order. This helps devices with keeping the track of their shares, and so maintaining the synchronization among the IoT devices in $\mathcal{N}$. Note that adding or removing device to/from network permanently or temporarily requires a set of rules on how to handle such cases/errors by the master device. We leave such discussions out in this article.

Handling concurrent communications initiated by different devices is another aspect of implementing the model. In this model, the transmitted share does not include any ownership related information, so the master device has some delay as it waits for receiving all shares for the entire message. During the transmission of a message, the master device expects a certain number of shares as the message length is fixed. Upon receiving the entire set of shares, the master device compares the received shares with its own records and determines which devices sent their current shares and which particular device has not. The model provides an authentication mechanism in a natural way.

It also helps on security side as the master device can detect alterations or modifications after a basic comparison of received shares and their order with its records. However, there is no mechanism to distinguish the shares for concurrent message transmissions. Therefore, a candidate resolution may require implementing a scheduling or priority mechanism among the IoT devices. For instance, a queueing mechanism can take a time-stamp or a prioritization added to the broadcasted messages into consideration, so the first message broadcasted is taken care of first, with a first-come-first-served logic. Before all shares of a particular message is transmitted, shares of another message are not sent. Other implementaion methods which handle concurrent communications without delay may require a different assumption on authentication.

Another implementation aspect is the performance of the participating devices. In this model, the performance of the master device is irrelevant, since it is chosen to be a device with relatively adequate computing power, such as a mobile phone or a PC. However, the

IoT devices should have adequate computational power to realize this model by handling basic actions such as triggering a communication, broadcasting bit values of its message in $\mathcal{N}$, keeping the track of bits shared with itself and its own shares, sending its current share to the master device upon receiving a bit value from another device, etc. Roughly speaking, none of these operations are costly and consequently do not require high computation power on an IoT device. Hence, the model seems manageable by constrained devices in performance perspective.

The model is assumed to handle messages (bitstrings) of fixed length s, then for every bit of a message of length s, the master device distributes in advance two shares (rows) of length 2^{n-1} to each of n devices in $\mathcal{N}$. Depending on the bit values of messages, devices decide which shares to use. If there are approximately p messages that are sent to the master device during a period from one share distribution to the next one, then an IoT device requires to keep at least $p \cdot s \cdot 2^n$ bits in the order distributed by the master device. For instance, in a scenario with an IoT network of n devices in which overall one message out of 128 predetermined messages is sent every minute and share distribution process is conducted on daily basis, each device needs to keep at least $2^{n+12.3}$ bits or approximately 2^n kB. In a small-sized network, this requires IoT devices to have low memory capacities and, as the number of devices in a network increases, their individual memory capacities are required to increase accordingly. This may be considered as a limitation during the implementation of the model, however dividing the original network into small-sized sub-networks can be considered as a resolution, but then it requires managing messages from different sub-networks. Another resolution may be adopting a k-regular graph, where $k << n$, as a topology and adjusting the model accordingly.

Security. The proposed model mimicks the idea of the visual secret sharing scheme introduced by Naor and Shamir [12] and this paper discusses how to use the scheme in a digital setup to send messages of IoT devices collaboratively to a remote master device. It intends to avoid a single point of failure which is a common security issue for IoT networks. Instead of using a central hub that is processing the messages on the behalf, the model lets IoT devices send their own shares through a dummy router. The original paper [12] provides the reader with a construction of n-out-of-n secret sharing scheme which was adopted in this paper as well. It also includes a proof in which the collections C^0, C^1 (as described and constructed above) are proved to result in an n-out-of-n scheme. For a randomly picked matrix with n rows from one of the collections, if its rows are used as shares, then possession of any $n-1$ rows does not reveal any information of the membership of this matrix in collections i.e. rows in collections C^0, C^1 have uniform distribution. Therefore, a probabilistic approach does not give an opponent an advantage either.

In the model, an IoT device d does not send its rows (shares) as its neighbors transmit their current rows to convey the message of d to a remote master device, so an eavesdropping third party misses a row of every matrix. Even if it captures all other $n-1$ rows, this does not reveal whether the transmitted rows belong to a matrix from C^0 or C^1, so it does not reveal whether the collaboratively transmitted bit value is 0 or 1. On the other side, it is assumed that matrices in each pair differ from each other by at least two rows, so transmitting just $n-1$ rows does not create any ambiguity on the master device side. Since the master device stores all pairs of matrices, it can still determine the matrix, and so the corresponding bit value, upon receiving $n-1$ rows.

The master device not only stores the rows, but it also stores them in the order it distributed to the IoT devices. Depending on the value of next transmitted bit, it already knows what to expect next, i.e. it is either a set of $n-1$ rows of a matrix M^0 in C^0 or M^1 in C^1, and the master device is aware of that the pair of matrices (M^0, M^1) is next. For some reason, if the transmitted rows or their order are replaced or modified, the master device detects it. This prevents a replay attack.

In case of malicious devices existing in $\mathcal{N}$, there are two possible types of attacks. If the malicious device is not the owner of the message, then it may send a row (share) other than it is supposed to. Unless all other rows transmitted to the master device by other devices are included in M^0 and M^1 concurrently, this does not create an ambiguity on the master device and it will be detected. If multiple malicious devices collaborate within the network, they still need to know other devices' shares in advance. However, if the malicious device is considered to be the owner of the messages, it can trigger the communication frequently and try sending redundant messages back to back. This may cause denial of services. The master device can keep the track of sent messages, their owners and frequencies. Proper precautious methods can be implemented to detect such attacks, but this may be costly as the malicious nodes need to be removed and it requires a new share distribution.

5 Discussion

This study proposes a new approach for IoT devices securely communicating their messages to a remote master device through an insecure channel. The solution uses the idea and structure of the visual secret sharing scheme introduced by Naor and Shamir [12] in a digital format and intends to achieve a method which eliminates a central unit that handles messages. Instead, IoT devices collaborate in sending the message of a device. The necessary operations done by IoT devices are not complex or do not require high memory capacity, so it is easily adaptable by constrained devices. In practice, there may be some restrictions on implementation of the model in a larger scale due to the memory requirements, however there are some potential resolutions, one of which is adopting a k-regular graph as a topology instead of a complete graph.

If there are n devices in $\mathcal{N}$, then a bit string of length 2^{n-1} is transmitted by IoT devices for every bit of a message. This may be considered as a redundancy, but in return, a resolution for avoiding a single point of failure becomes available which can also be considered as an alternative way of maintaining security and authentication.

The proposed method requires share distribution on a regular basis. However, removing or adding nodes require some additional distributions other than the scheduled ones. It also assumes that predetermined messages of fixed lengths are transmitted to the master device, but not in the reverse direction. An easy modification on the model may allow sending dynamic messages of fixed length instead of predetermined messages depending on the capacities of the IoT devices, but implementation of the master device sending messages back to IoT devices needs further assumptions.

References

[1] Whitmore A, Agarwal A, Da Xu L. The internet of things-a survey of topics and trends. Information systems frontiers 2015. 17 (2): 261-274. doi: 10.1007/s10796-014-9489-2

[2] Intel Corp., Could smart homes be as commonplace as smartphones by 2025?, https://download.intel.com/newsroom/kits/iot/pdfs/IntelSmartHomeSurveyBackgrounder.pdf, 2015.

[3] Wortmann F, Flüchter K. Internet of things. Business & Information Systems Engineering 2015. 57 (3): 221-224. doi: 10.1007/s12599-015-0383-3

[4] Sinha P, Boukhtouta A, Belarde VH, Debbabi M. Insights from the analysis of the Mariposa botnet. In: 2010 Fifth International Conference on Risks and Security of Internet and Systems; Montreal, QC, Canada; 2010. pp. 1-9. doi: 10.1109/CRISIS.2010.5764915

[5] Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H et al. Comprehensive experimental analyses of automotive attack surfaces. In: 20th USENIX Security Symposium 2011; San Francisco, Ca, USA: pp. 77-92.

[6] Larson UE, Nilsson DK. Securing vehicles against cyber attacks. In: Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead; Oak Ridge, TE, USA; 2008, pp. 1-3. doi: 10.1145/1413140.1413174

[7] Zaho Y. Telematics: safe and fun driving. IEEE Intelligent Systems 2022; 17 (1): 10-14. doi: 10.1109/5254.988442

[8] Wolf M, Weimerskirch A, Wollinger T. State of the art: Embedding security in vehicles. EURASIP Journal of Embedded Systems 2007; 1-16. doi: 10.1155/2007/74706

[9] Rouf I, Miller R, Mustafa H, Taylor T, Oh S, Xu W et al. Security and privacy vulnerabilites of in-car wireless networks: a tire pressure monitoring system case study. In: 19th USENIX Security Symposium 2010; Washington, DC, USA; 2010. pp. 323-338.

[10] Koscher K, Czeskis A, Roesner F, Patel S, Kohno T et al. Experimental security analysis of a modern automobile. In: 2010 IEEE Symposium on security and privacy; Oakland, CA, USA; 2010. pp. 447-462. doi: 10.1109/SP.2010.34

[11] Ronen E, Shamir A. Extended functionality attacks on iot devices: The case of smart lights. In: 2016 IEEE European Symposium on Security and Privacy; Saarbrücken, Germany; 2016. pp. 3-12. doi: 10.1109/EuroSP.2016.13

[12] Naor M, Shamir A. Visual Cryptography. In: Advances in Cryptography - EUROCRYPT'94; Perugia, Italy; 1994. pp. 1-12. doi: 10.1007/BFb0053419

[13] Nielsen F. Topology of interconnection networks. In: Introduction to HPC and MPI for Data Science. USA: Springer Cham, 2016, pp. 63-97. doi: 10.1007/978-3-319-21903-5_3