A Scalable and Secure Protocol for RFID Based on "Advanced Encryption Standard" and Elliptic Curve Cryptography

**Christiana Couto, Gabriela Moutinho de Souza Dias,
Ronaldo Moreira Salles, Gustavo Claudio Karl Couto**

[1]Military Engineering Institute
Rio de Janeiro, Brazil
christianacouto, gabriela, salles, gustavokcouto@ime.eb.br

Abstract. *Radio frequency identification systems are widely used to uniquely identify objects in many applications such as magnetic cards, security tags, and logistic management systems. Despite these advantages provided by the RFID system, there remain a multitude of security concerns related to spoofing and espionage that are all concerned with radio frequency interception. Current research analysis is promising, particularly the work of Ibrahim and Dalkiliç. Their findings are, however, limited by the amount of tags that can be processed—–which under certain circumstances can exhaust the server. We designed our new protocol and conducted a performance analysis. When compared to the Ibrahim and Dalkiliç's protocol, our results revealed a drastic reduction in the communication cost that was proportional to the amount of tags authenticated. Once the results were tallied, we modeled our proposed protocol in a discrete event simulation. As a proof of concept, our protocol was then implemented in software and analyzed through an experiment whose metrics were: tag identity search speed in the back-end database and amount of tags. Our results show that the proposed protocol offers better performance compared to current standard iterations of similar technology.*

1. Introduction

The RFID system is composed of readers, tags, and a subsystem that processes data. The tags can store and transmit information, which can either be classified as active if the tags have an energy source integrated in them, or passive, in case the reader's magnetic field is their energy source. The tags have limited memory, storage, and processing resources. As a consequence of these limitations, the implementation of security techniques is difficult.

Tags communicate with readers according to protocols in order to verify mutual legitimacy and uncover the identity of the tag. Despite the advantages that RFID technology offers, security issues continue to plague the technology due to remote data transfers. In one example, patients in hospitals were identified by RFID tags in order to secure the communication channels that store and transmit their health data [Ibrahim and Dalkiliç 2017].

The RFID system's scalability is important in many cases [Hsi et al. 2015]. For example, when retailers identify products distributed in groups to different distribution centers or points of sales by using RDIF tags. Scalability is a factor that affects many different dimensions, such as the horizontal and vertical dimensions. The former relates to equipment quantity and the latter to the processing power of a single piece of equipment.

The complexity of scalability and its applications inform the solution proposed in this work. In addition to Ibrahim and Dakiliç's proposed protocol for mutual authentication between tags and readers in RFID systems, our main contribution adds an additional step and an extra search index to Ibrahim and Dalkiliç's protocol. These new protocols still provides confidentiality, integrity, and mutual authentication, in order to protect the communication of sensitive information. Our method, however, scales successfully, while Ibrahim and Dalkiliç's previous protocol does not. A performance analysis was conducted after designing our new protocol, which ended successfully. As a proof of concept, our protocol was then implemented in software and an experiment was conducted to evaluate the results.

1.1. Contributions

Ibrahim and Dalkiliç's protocol was designed for an RFID system with only one tag and one reader in mind. In order to apply their method for scale, each protocol repetition would necessarily need to repeat itself multiple times. An optimization technique is not explained in Ibrahim and Dalkiliç's work.

Furthermore, the authors should have proposed a cryptographic key management mechanism that would regularly change user keys in order to increase security.

Therefore, we propose a new protocol to be deployed in RFID system's composed of multiple tags and reader. A prerequisite for this improvement demands that systems implement tag groupings in such a way that each group be designated to at least one reader. Our solution to improving Ibrahim and Dalkiliç's proposal is to add a collective authentication step through tag grouping. Group identities can be used to optimize the authentication search in order to increase the protocol's scalability.

Further contributions include:

- Design implementations for pseudo-identity deployment instead of real identities that further renew during each tag's identification. This method increases entropy because the key space becomes bigger.
- Possibility of ownership transfer applications to secure products that belong to different owners along a supply chain;
- Securitization against attacks, such as: Denial of Service, Replay, Impersonation, "Man-in-the-Middle" attacks, Tracking, Desynchronization and Cloning.
- A performance analysis showing a reduction in the cost of communication for the proposed protocol relative to Ibrahim and Dalkiliç's protocol. The reduction in cost is proportional to the amount of tags and authentications performed.

The rest of the paper is organized as follows. In Section 2, we discuss works related to Ibrahim and Dalkiliç's protocol. Then in Section 3 we present our proposed protocol. Security, performance, and experimental evaluations are described in Section 4. Section 5 concludes the paper and proposes different directions for future work and study.

2. Related Work

A comprehensive analysis of cryptographic applications in protocols for RFID systems was conducted by Couto et al. [Couto et al. 2021]. Highlights of this study

ware the comparison among some sample protocols and an up to date evaluation of the State-of-art. The protocol proposed by Ibrahim and Dalkiliç for mutual authentication between tags and readers in RFID systems was analyzed within this study. Their protocol implemented a hybrid cryptosystem where the ECDH algorithm creates a shared key that can be encrypted by the AES algorithm. This shared key would then be used to encrypt tag identity and other sensitive information transmissions. Their protocol is composed by an initialization and an authentication step. Ibrahim and Dalkiliç further suggest that researchers design an integrated wireless sensor network of tags and readers as the basis of future work.

Despite their optimism, the following issues in the Ibrahim and Dalkiliç protocol were pointed out by researchers:

1. [Kösemen et al. 2018] point out that the protocol depends on the WISP embedded pseudo-random number generator which is proven to not be secure;
2. Ibrahim and Dalkiliç's protocol was designed for an RFID system with only one tag and one reader in mind. In order to apply their method for scale, each protocol repetition would necessarily need to repeat itself multiple times;
3. [Alaoui et al. 2021] point out that the protocol is not scalable in the vertical dimension because the server may perform an exhaustive search for the tag's ID; and
4. [Arslan et al. 2021] affirm that the protocol fails to provide forward and backward privacy. They proposed a new protocol to overcome this shortcoming.

The second and third aforementioned points are the main motivations and inspirations for this work.

3. Proposed Protocol Description

Our proposed protocol consists of four steps, and they are: initialization, renovation, collective authentication, and individual authentication.

We assume all the following:

- That the connections between the servers and the readers are safe, but that the remote connections between tags and readers are vulnerable to attacks [Deursen and Radomirovic 2008];
- That, initially, all tags must be grouped together, but that they may be later individually authenticated;
- That the same reader will be used to perform group and individual authentications; and
- That the initialization step is performed in a secure manner.

The process begins with the initialization step, which configures the system. Once this step is initiated, each new connection between a tag and the server goes through a renovation step which is performed similarly to how a blockchain works in order to renew the tag's pseudo-identity.

Whenever an authentication is required, the collective authentication is performed by a group of tags and one reader to detect if at least one tag is missing or is fake. The collective authentication creates an extra layer of security—which is the group identity.

In case individual authentication or a tag's identity retrieval is necessary, the individual authentication step can then be performed.

Applications for our new RFID protocol can affect the Internet of Things, like smart grids and the security of industrial control systems. Couto et al's study on RFID implementations in the industrial sector [Couto et al. 2020a, Couto et al. 2020b, Couto et al. 2020c] investigate such applications. Similar work done by Roman et al proposing a mutual authentication cryptographic protocol based on elliptic curve cryptography for smart grids, supports further research [Román et al. 2020].

3.1. The Initialization and Renovation Processes

Initially, the system must start a secret sharing scheme to properly run [Blakley 1979]. This scheme distributes hidden parts of data (which will be referred to as "shares") through tags. The tags can only be discovered once they are all gathered together.

The server generates and distributes "shares" to the readers and tags of each group. Group identities prevent intruders from accessing the system. Even if an adversary is able to capture all the tags' "shares" from the same group, he or she would still need to guess the reader's "share" to find the correct group identity. The reader's "share" can not be captured because it is never transmitted. $2x + 1$ "shares" must be created for each group of x tags through the following process:

1. A pseudo-random number generator must generate $2x + 1$ binary numbers (that is, the "shares") with the same length as the group identity;
2. Tag "shares" are separated into two groups of "shares," odd "shares" and even "shares" for practicality. They are then assigned to different tags;
3. The reader also receives one "share" from the server; and
4. The reader calculates the group identity through the $\oplus$ operation for all even "shares," including the reader's "share".

Separating "shares" into two types and assigning them to two groups drastically increases security. These "shares" are distributed between the tag and the reader. Each tag receives two "shares"—one odd and one even. Even types of "shares" are applied to the secret sharing scheme during the collective authentication step and for new pseudo-identity generation. The odd "shares" are solely used for pseudo-identity generation.

The user inputs data to the server concerning the total number of groups and tags in each group. Once entered, the server generates group identities and odd or even "shares". Once the "shares" are distributed to the tag, each reader receives one "share," is assigned to at least one group and receives a relative number of tags for those groups. The relative number of tags in a group is important as it limits the range of connections a reader can accept.

The purpose of the renovation step is to generate new pseudo-identities for each tag. A key generation mechanism is adapted to protect the tag's identity by generating and deploying new pseudo-identities for each tag. The key generation approach is inspired by Basha et al.'s secret key generator [Basha et al. 2019]. The mechanism inputs consist of the even and odd "shares" that are combined to generate new pseudo-identities through a hash key derivative function (HKDF).

Every time the server starts communicating with a tag the key generating process is triggered to create a new pseudo-identity.

In parallel, a public key encryption mechanism is implemented to generate new odd "shares" that are required to create new inputs for the HDKF function.

A server-side counter and an internal tag counter keep track of the renovation step's executions. The counters must be synchronized in order to verify that the information on the tag is correct and undamaged. In case any tag is damaged, the counter value can be used to restore the tag "shares" and pseudo-identity.

Combining the secret sharing and the pseudo-identity optimizes the process of generation, which encourages new variables generation.

3.2. Collective Authentication Step

The authentication process must start through the collective authentication of a group of tags. This step begins when a group of tags and a reader connect to each other. Data on even "share" tags is automatically read by the reader if the tag is nearby. To guarantee that network errors are detected, the SHA3 256 bits algorithm hash function is implemented to preserve the "shares"' integrity.

When the reader receives both the tag's "share" and its hash value, it will verify the integrity of the tag's "share" through the hash function algorithm. If the tallied value matches the received hash value, the tag's "share" is considered valid. Each valid tag's "share" is stored in a list. When the number of valid "shares" matches the number of tags in each group, the reader will use the "shares" $\oplus$ operation to sum all "shares" in the group identity.

Once a group identity is created, the reader sends it to the server, which then looks for a match in the database. If a match is found with an existing group identity that is assigned to a specific reader, the server confirms the legitimacy of the identity to the reader. If the server reports an error, it is presumed that at least one tag is fake or missing. Should such a case arise, corrective actions must be performed by the system's administrator in order to verify the suspect group identity.

If the collective authentication step is successful, then the individual authentication step can be performed by the same reader.

3.3. Individual Authentication Step

After the collective authentication is performed, the user may have to authenticate tags individually or retrieve their identities to access the system.

The individual authentication step implements the elliptic curve digital signature standard (ECDSA), elliptic curve Diffie-Helman (ECDH) algorithm, and advanced encryption standard (AES) algorithms to encrypt each tag's identity. The AES algorithm encrypts the tag identity. The AES algorithm's shared secret key is encrypted and transmitted by the ECDH algorithm through a secret sharing scheme. The ECDSA is then used to sign the ECDH public keys to avoid man-in-the-middle attacks.

Our research is conducted using a secp192r1 standard implementation of the ECDH algorithm and ECDSA [Standards for Efficient Cryptography Group]. The first

step requires that the reader and tag agree on the elliptic curve domain parameters. Once this validation step is complete, the reader will generate a random number $l \in [1, n-1]$ as its private key and calculate a unique public key $L = lG$ that will be used in the ECDH algorithm. Once created, the reader will send L to the tag.

Similarly, the tag will generate a random number $e \in [1, n-1]$ as its private key and calculate a unique public key $(E = eG)$ that will be used in the ECDH algorithm. Once the ECDH keys are generated, the tag will generate keys e', E' similarly to how the previous keys were generated for the ECDSA. The ECDSA then generates digital signatures (o, p) for the tag's ECDH public key.

It is important to note that readers don't need to sign their public key because they are already authenticated by the collective authentication step. We assume that collective authentication removes the need to authenticate the reader more than once.

Once the tag receives the reader's public key the tag creates a secret key $K_{LE} = eL = elG$ through the ECDH algorithm. After the secret key's creation, the tag uses it as an input in the AES algorithm to encrypt its pseudo-identities or any other type of secret information. Once the previous step is complete, the tag sends its public keys, the ECDSA signature, and the encrypted data to the reader.

The reader verifies the tag's ECDH public key signature through the ECDSA once the key is received. If the verification fails, the session ends immediately. If everything is verified correctly, the reader calculates the shared secret key $K_{LE} = lE = leG$. The generated key is used to decrypt the tag pseudo-identity. The reader sends the pseudo-identity to the server for validation. If the server finds a matching identity in the database, it will return a validated result to the reader. Should the server return an invalidated result, the session will end immediately.

4. Evaluation

4.1. Security analysis

The security analysis follows an architecture that is similar to that constructed by Ibrahim and Dalkiliç. The protocol's security analysis differs in several noteworthy ways:

- In our protocol, the readers and tags are authenticated based on the "share"' ownership. This means that should the tag fall under attack, the attacker would be unable to authenticate the RFID tag;
- The protocol is resistant against man-in-the-middle attacks because the server and the tags only communicate with each other during the renovation step. In addition, the renovation step is not necessary for authentication. However, it is important to consider the maximum amount of tags a reader can assign in order to avoid over-processing—which could result in unavailability;
- An attacker cannot personify a reader during the collective authentication step because the reader's "share" is kept secret;
- If an attacker tries to tamper with any "shares" or hashes during the collective authentication step, the tags will not be validated by the reader due to the verification process; and
- Pseudo-identity deployment increases the tag's privacy.

4.2. Communication Costs

It is important to optimize the communication cost protocol of the tag-reader because RFID tags have limited processing power. The communication costs between servers and readers are considered negligible, therefore there is impetus to repeat processes that verify information between the reader and server. The communication cost-efficiency data is provided by Ibrahim and Dalkiliç, Liao and Hsiao, Zhao, Chou, Arslan et al. Zhang and Qi, and is available on the Table 1.

Table 1. Communication costs comparison of mutual authentication protocols.

Protocol	**Communication Cost (bytes)**
Proposed Protocol	160
[Zhang and Qingqing 2014]	160
[Liao and Hsiao 2014]	168
[Zhao 2014]	168
[Arslan et al. 2021]	168
[Ibrahim and Dalkiliç 2017]	176
[Chou 2014]	184

It is clear from Table 1 that the communication costs of the proposed protocol are lower or equal to the communication costs listed on Table 1.

4.3. Experiment

As a proof of concept, our protocol was then implemented in software. In order to understand the scalability of the proposed protocol, an experiment analyzing the horizontal dimension of the system's scalability was performed. The platform chosen was a Linux 18 ubuntu with a intel core i7 8th generation 3GHz/s processor. To set the experiment's metrics, an assigned number of tags for every group for the horizontal dimension study and the identities' search speed within a server's database for the vertical dimension study.

The researchers assumed that the relationship between a high amount of tags and the processing times is proportional.

A thousand samples were collected and the number of tags in each group was $\in$ [2,100]. In total, the protocol ran 9900 times. The result is shown in Figure 1. The samples are shown in light blue and the standard deviation in dark blue. The average sample value is 2.79μs and the average standard deviation value is 0.037s. The peak around a 85-tag group can be explained by the computer's scheduler time variations.

The second result is shown in Figure 2. The samples are shown in light orange and the standard deviation in dark orange. The average sample value is 0.069s. One can observe that the result is almost linear.

The results show that the server-side protocol communication cost processing time

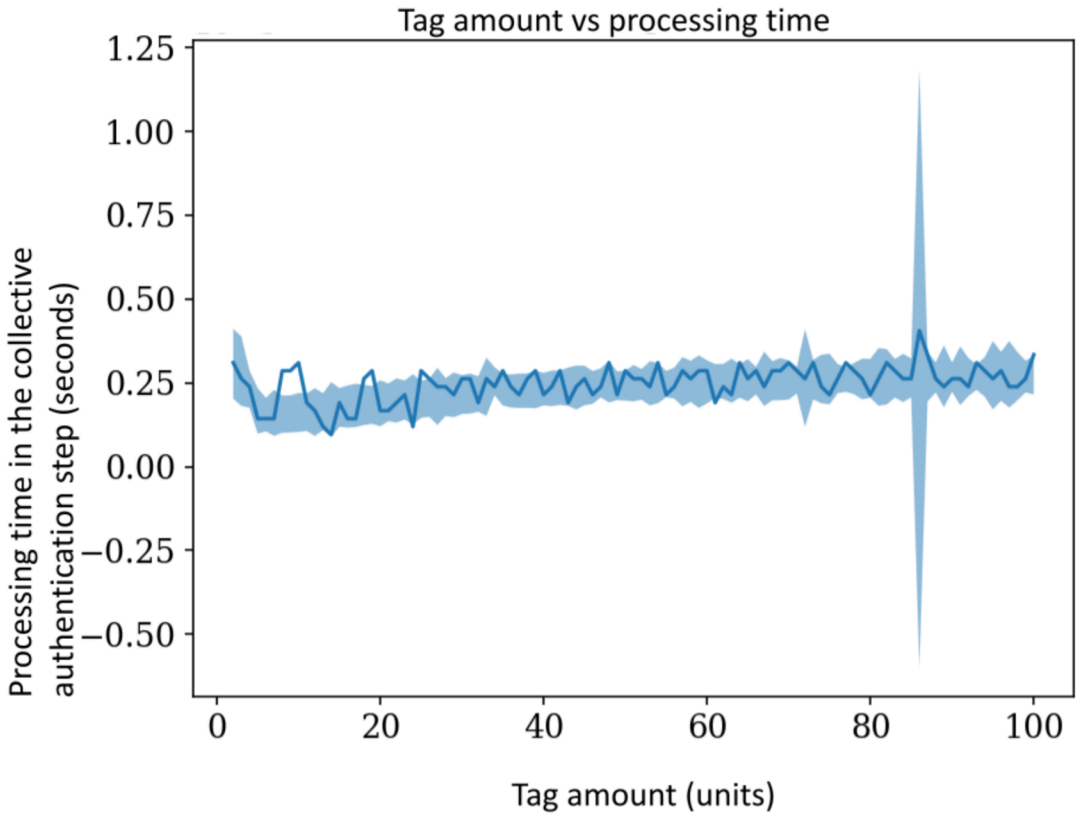

Figure 1. Amount of tags in a group versus group identity search time, in the server, during the collective authentication step.

is approximately constant when compared to the linear results provided by Ibrahim and Dalkiliç.

5. Conclusion

This work seeks to improve Ibrahim and Dalkiliç's proposed protocol by considering the question of scalability. We add a new step to their protocol which performs a collective authentication on every group of tags. Our solution consists of four steps: initialization, renovation, collective authentication, and individual authentication.

As a proof of concept, we tested our protocol by implementing it in software and then analyzing the data through an experiment whose metrics were: amount of tags and identities' search speed within a server's database. The results show that the protocol communication cost processing time complexity on the server-side is constant compared to the linear results provided by Ibrahim and Dalkiliç's protocol. The cost of communication when using this protocol is notably more cost-effective.

Future research and study may include:

- Create and implement a tag identity search mechanism (e.g. binary trees) in the server-side;
- Implement a threshold secret sharing scheme or a proactive sharing scheme in the collective authentication step;

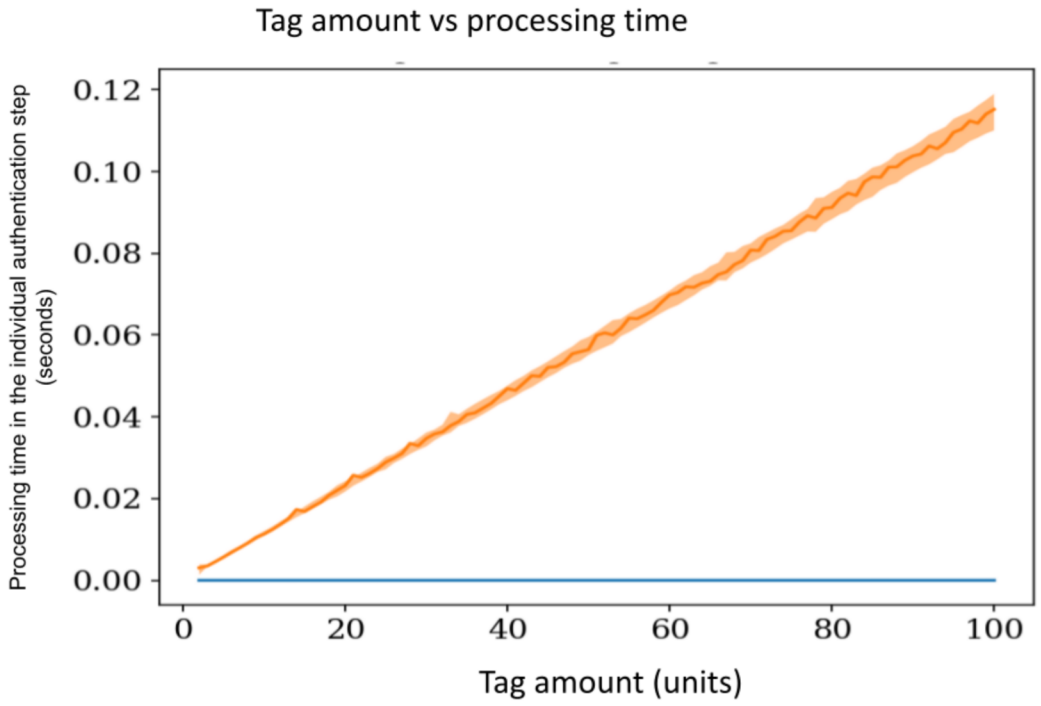

Figure 2. Amount of tags in a group versus all the tag's identities search duration, in the server, during the individual authentication step.

- Develop tag's communication in such a way the tags can exchange messages among each other in a secure manner;
- Implement the protocol in a test-bed. The WISP deployment would be recommended.

6. Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. The authors thank Alan Bariman for his helpful editing review.

References

Alaoui, H., el Ghazi, A., Zbakh, M., Touhafi, A., and Braeken, A. (2021). A highly efficient ECC-based authentication protocol for RFID. *Journal of Sensors*, 2021:1–16.

Arslan, A., Aldirmaz, S., and Erturk, S. (2021). A secure and privacy friendly ECC based RFID authentication protocol for practical applications. *Wireless Personal Communications*, 120.

Basha, M., Alalak, S., and Idrees, A. (2019). Secret key generation in wireless sensor network using public key encryption. In *ICICT 2019, © 2019 Association for Computing Machinery. ACM*.

Blakley, G. R. (1979). Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318.

Chou, J.-S. (2014). An efficient mutual authentication RFID scheme based on elliptic curve cryptography. *The Journal of Supercomputing ¿ Issue 1/2014*, pages 75–94.

Couto, C., Couto, G. C. K., and da Cunha, A. E. C. (2020a). Análise da segurança de redes em sistemas de automação e controle industriais: estudo de caso com a planta mecatrime. In *VIII Simpósio Brasileiro de Sistemas Elétricos*.

Couto, C., Couto, G. C. K., and da Cunha, A. E. C. (2020b). Modelagem da segurança da informação em sistemas de automação e controle industriais: estudo de caso com a planta mecatrime. In *VIII Simpósio Brasileiro de Sistemas Elétricos*.

Couto, C., Couto, G. C. K., and da Cunha, A. E. C. (2020c). Rumo a conformidade da segurança da informação em sistemas de automação e controle industriais: estudo de caso com a planta mecatrime. In *XXII Congresso de Computação e Sistemas de Informação (ENCOINFO)*.

Couto, C., Salles, R. M., de Souza Dias, G. M., and Couto, G. C. K. (2021). Cryptography applications in protocols for rfid systems. In *XV Simpósio Brasileiro de Automação Inteligente*.

Deursen, T. and Radomirovic, S. (2008). Attacks on RFID protocols. *IACR Cryptology ePrint Archive*, -:310.

Hsi, C.-T., Lien, Y.-H., Hui, C., and Chang, H. (2015). Solving scalability problems on secure rfid grouping-proof protocol. *Wireless Personal Communications*, 84.

Ibrahim, A. and Dalkiliç, G. (2017). An advanced encryption standard powered mutual authentication protocol based on elliptic curve cryptography for RFID, proven on WISP. *Journal of Sensors*, 2017:2367312:1–2367312:10. Acesso em: 08-12-2021.

Kösemen, C., Dalkiliç, G., and Aydin, O. (2018). Genetic programming based pseudo-random number generator for wireless identification and sensing platform. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26:–.

Liao, Y.-P. and Hsiao, C.-M. (2014). A secure ECC-based RFID authentication scheme integrated with id-verifier transfer protocol. *Ad Hoc Networks*, 18:133–146.

Román, L., Gondim, P., and Lopes, A. (2020). A lightweight authentication protocol for advanced metering infrastructure in smart grid. In *Anais do XII Simpósio Brasileiro de Computação Ubíqua e Pervasiva*, pages 21–30, Porto Alegre, RS, Brasil. SBC.

Standards for Efficient Cryptography Group. Sec 2. standards for efficient cryptography group: Recommended elliptic curve domain parameters. Acess in: 08-12-2021.

Zhang, Z. and Qingqing, Q. (2014). An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography. *Journal of medical systems vol. 38,5 (2014): 47*.

Zhao, Z. (2014). A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem. *Journal Medical Systems*, page 38(5):46.