Secure and Lightweight Authentication Protocol for Smart Metering System

Hind El Makhtoum
Engineering Sciences Laboratory
ENSA Kenitra, Ibn Tofail University
Kenitra, Morocco

Youssef Bentaleb
Engineering Sciences Laboratory
ENSA Kenitra, Ibn Tofail University
Kenitra, Morocco

***Abstract*—One of the main advantages of the new power grid over the traditional grid is the intelligent energy management by the customer and the Operator. Energy supply, demand response management, and consumption regulation are only possible with the smart metering system. Smart meters are the main component of that system. Hence, a compromised smart meter or a successful attack against this entity may cause data theft, data falsification, and server/device manipulation. Therefore, Smart grids' development and the guarantee of their services are related to the ability to avoid attacks and disasters by ensuring high security. This paper aims to provide a secure and lightweight security protocol that respects the IOT device constraints. The proposition deploys the distributed OTP calculations combined with the Blake2s hash function and the Ascon AEAD cipher to ensure authentication, confidentiality, and integrity. We propose a performance analysis, an informal and a formal security evaluation made by the AVISPA-SPAN tool. Also, we compare the proposed protocol to other similar works. The assessment proves that the proposed protocol is light, valid, secure, and robust against many attacks that threaten the NAN area of the smart metering system, namely, MITM and replay attacks.**

***Keywords*—Internet of things; confidentiality; integrity; authentication; Ascon; Blake2; AVISPA**

I. Introduction

The Internet of Things is growing rapidly and deploying more devices, systems, and other entities. These devices carry a lot of critical data, making these infrastructures a greedy target of attackers. The accessibility of these devices must be controlled so that only authorized users can access the destination server or gateway. Furthermore, any data leakage or corruption can cause serious problems for people, systems, and companies. Therefore, securing IOT must be an occupation for scientists to ensure the system's security and thus maintain its services' effectiveness.

One of the critical IoT applications is the smart grids, namely the smart metering system. Smart meters make information available to customers to manage their consumption behavior. They also provide the necessary information for operators to balance energy response and demand [1]. Smart meters are an energy revolution that will dramatically improve the efficiency and reliability of the power grid. Therefore, the metering system is a greedy target for attackers who could turn these benefits into an absolute disaster if security is not ensured.

Given such risks encountered by the grid, namely metering, a robust and secure system is a must. However, the nature of the smart metering system falls under the same limitations and constraints of the IoT but with major security risks. Indeed, smart metering systems face smart meters limitations such as computing and storage constraints [2] and those of wireless communication networks that further increase the risk of intrusions and attacks [3]. Under these conditions, deploying strong and efficient traditional security solutions is not feasible on smart meters as they involve cumbersome mathematical calculations. Therefore, the design of lightweight and strong security protocols is necessary to ensure the grid's safe operation, protecting data and users and thus maintaining customer confidence.

In light of these limitations, this article aims to propose a scheme to securely authenticate smart meters (namely in this work: SM/Device) to the neighborhood gateway (Namely in this work: SEVER/GATEWAY) while the association phase of the wireless communication. The proposed solution is based on the distributed OTP approach that lightens calculations and storage on the device side. We deploy lightweight and secure protocols for hashing and encryption. In addition, we use the blake2s as the hash function used in the OTP calculations and for the server's authentication. Ascon, the finalist of the Caesar Competition launched by NIST in 2014, is the AEAD cipher deployed in the proposed solution. Ascon cipher is a lightweight solution to ensure authenticity, integrity, and data confidentiality.

The contribution of this paper can be summarized as follows:

- We propose a secure and lightweight security protocol for the NAN area of the smart metering system to address the related security problems and the eventual limitations by using lightweight protocols and session varying parameters.
- We perform formal security analysis by the AVISPA-SPAN simulation to evaluate the proposed protocol's security, validity, and robustness against replay attacks and MITM attacks.
- We also perform informal security analysis to prove that the protocol is robust against many classical and well-known attacks.
- We perform a performance analysis of the protocol regarding computational costs, communication costs, and storage.

- According to the security and performance analysis, we compare the protocol against similar works, and we conclude that the proposed scheme achieves good security and performance results.

The rest of the paper is organized in the following way: Section 2, where we will introduce some related works that address the same issue. Section 3 will be dedicated to preliminaries about the Ascon cipher, the Blake2 hash function, and the distributed OTP approach. Then, we will present the proposition in Section 4 with the formal and informal security analysis. Performance analyses are made in Section 5. Before the conclusion, a comparison of the proposed work with other works is presented in Section 6.

II. Related Works

The authors of the work [4] provided a secure framework for IoT-based Healthcare systems that addresses four security issues in the fields. The healthcare system, an application of IoT, is also a target to malicious users that may cause dysfunctions. This paper addresses the system's access control by using AES128 with a common pre-shared key to stop the external sensors from accessing the healthcare system. The authors also ensure authenticity by using the public and private keys of the RSA-1024 to ensure that the sensors and medical persons that send messages are real and authorized to make the communication. This paper also addresses confidentiality thanks to the point-to-point encryption based on the AES128. To ensure Integrity, the authors deployed the message authentication code (MAC) based on AES-128 to ensure that the data was intact and not altered [4]. Another work that dealt with the same issue is [5]: The author proposed a protocol based on a mutual authentication based on OTP authentication for both the Device and the gateway. Once the authentication is ensured, a key is generated from an irreversible hashing function. The Key is used as an entry for the AES-GCM protocol. This work is based on AES-GCM to ensure the confidentiality and Integrity of message exchange and thus secure the channel. In the paper [6], The authors presented a security protocol in the smart home domain. The protocol is a combination of encryption algorithms (AES-GCM, RSAOAEP) implemented with SHA3-512 to ensure the confidentiality and integrity of the data communicated by the sensors. The system of [7] provides security based on X.509 certificate, RSA-based Public Key Infrastructure (PKI), hard tokens, challenge/response protocols, and operators' proxies. The system ensures confidentiality, integrity, non-repudiation, privacy, and anonymity.

The cited schemes all addressed the security issues in IoT. However, there are protocols that they deploy separated cryptographic protocols, which are slower and greedy on resource consumption. Other protocols deployed the AEAD AES-GCM, which is vulnerable to the nonce-misuse attack. This vulnerability makes the protocols vulnerable and compromises confidentiality and Integrity of the protocols.

III. Preliminaries

A. Smart Metering System

The architecture of smart grids consists of three levels: the Home Area Network (HAN) with smart meters and home appliances, the middle level or Neighborhood Area Network (NAN) that connects the HAN with the WAN through gateways and concentrators, and the top level of the Wide Area Network (WAN) that is administered by a control center and MDMS servers.

At the HAN level, the smart meters collect data about the electricity consumption of the home equipment and communicate it to the Control Center through the neighborhood gateways. This data is used to make critical decisions about users and all participating parties in the grid.

The smart metering system has a potential role in the smart electrical grid. Indeed, Operator's control centers adjust the smart meters of their client remotely in order to respond adequately to the client's needs and specifications. These bidirectional operations monitor the client's equipment in the HAN area, such as consumption, device specifications, location, and pricing, thus deciding the amount of energy delivered to the client. Hence, confidentiality, privacy, and Integrity are major concerns of the grid to protect clients' consumption and their privacy to protect the grid from attacks and malware that could affect customers' privacy and cause a real disaster [8], [9].

In addition, The smart meters are accessed by multiple agents such as technicians, customers, and operators, which makes them vulnerable to physical attacks. Consequently, smart meters must be well protected in terms of authentication. Moreover, authenticity protects the Operator's system and the other customer from malicious technicians, neighbors, customers, or any other malicious intruder.

B. AEAD CIPHERS: Ascon

ENCRYPTION is a cryptographic method that encodes information to protect it and prevent unauthorized access. The encryption performs several operations(rounds, permutations.) on both Plaintext and a key. These operations produce an unreadable ciphertext that needs the previous Key to decrypt [10].

However, classical encryption does not ensure the authenticity of the message. To deal with this, authenticated Encryption with Associated Data brings up a new level of security thanks to the associated data. The associated data is data related to the sender's time and space that ensures the authenticity of the message and the sender [10].

1) Authenticated Encryption with Associated Data AEAD: Authenticated Encryption with Associated Data is a new cryptographic algorithm that ensures confidentiality, integrity, and authenticity. The technical difference between the AEAD and classical encryption is that the first one generates the ciphertext with a tag. Consequently, decryption and Integrity are performed simultaneously, and the Plaintext is readable only if the Tag is correct [6].

The AEAD encryption is defined as a set of algorithms: the key generation algorithm, the encryption algorithm, and the decryption algorithm.

The encryption algorithm has three inputs, the Nonce N, the associated data AD, and the Plaintext (message). It outputs the ciphertext and a tag : EK: $N \times AD \times Plaintext \rightarrow Ciphertext \times Tag$.

The decryption algorithm is the inverse algorithm with the nonce N, the associated data AD, the cipher C and the tag T as inputs. It outputs the Plaintext if C and T are valid : DK: N × AD × C × T → M or Error [6].

The most popular and highly adopted lightweight AEAD algorithm is the AES-GCM, which is appropriate for the constrained devices of the IOT [6]. However, it has exposed several shortcomings. The nonce should be unique for each message and non-repeating to prevent nonce misuse attacks, which is not practical for message exchanges. Also, keys used in the hash have been found weak [11].

In order to overcome these shortcomings, NIST launched 2014 the Caesar competition to find out more efficient and lightweight encryption algorithms in terms of applicability, robustness, and security [12].

Among 54 candidates in the first round, only 29 passed to the second one, and 15 candidates were selected for the third round. The two finalist lightweight applications portfolio of the competition is Ascon and ACORN. Ascon is the selected algorithm for the proposed protocol [12].

The two variants of Ascon were selected as the first choice for lightweight applications. Several works evaluated the Ascon algorithm. All analysis supports its large security margin without practice risks, vulnerabilities, or weaknesses [13].

2) ASCON Principle: Ascon is based on the sponge and duplex construction. The algorithm of Ascon is illustrated in Fig. 1 and Fig. 2: Initialization step: An Initial Vector of 64bits (ACON-128 and ASCON-128a), the secret key K, and The nonce N are concatenated to form a 320-bit state. This state passes through the transformation "p" for "a" times/ Then, the secret Key is XORed with the transformed state [13] [14].

Associated Data: The associated data is divided into r-bit blocks and XORed with the first r-bit blocks of the state. After that, a transformation "p" is applied to the resulting state "b" times and XORed again with a 1bit constant.

After these steps, the algorithm moves either to encryption or decryption with two additional phases for each :

Encryption: The plaintext is divided into r-bit blocks, XORed with the first r bits of the state (identical to associated data). Then, it generates blocks of ciphertext that are updated with "p" transformation for "b" times.

Decryption: The ciphertext blocks are XORed with the first r bits of the state and replace the first r bits of the states. They are again updated with the p transformation b times.

The final step after encryption/decryption is the finalization which consists of XORing the secret Key with the state and applying the p transformation "a" time to the state. The secret Key K is once again XORed with the resulted state. The Tag is the least significant 128 bits of the output [13], [14].

3) Ascon Security: The best-known key recovery attack can find the Key with 2^{104} time complexity if the initialization round number is reduced to 7.

Even if the state is recovered, it is impossible to use it for key recovery or forgery attacks. Even the vulnerabilities are not useful for practical attacks.

Algorithm 1: Authenticated encryption and decryption procedures

Authenticated Encryption $\mathcal{E}_{k,r,a,b}(K,N,A,P)$	Verified Decryption $\mathcal{D}_{k,r,a,b}(K,N,A,C,T)$
Input: key $K \in \{0,1\}^k, k \le 160$, nonce $N \in \{0,1\}^{128}$, associated data $A \in \{0,1\}^*$, plaintext $P \in \{0,1\}^*$	**Input:** key $K \in \{0,1\}^k, k \le 160$, nonce $N \in \{0,1\}^{128}$, associated data $A \in \{0,1\}^*$, ciphertext $C \in \{0,1\}^*$, tag $T \in \{0,1\}^{128}$
Output: ciphertext $C \in \{0,1\}^{\lvert P\rvert}$, tag $T \in \{0,1\}^{128}$	**Output:** plaintext $P \in \{0,1\}^{\lvert C\rvert}$ or $\perp$
Initialization	**Initialization**
$S \leftarrow \mathrm{IV}_{k,r,a,b} \Vert K \Vert N$	$S \leftarrow \mathrm{IV}_{k,r,a,b} \Vert K \Vert N$
$S \leftarrow p^a(S) \oplus (0^{320-k} \Vert K)$	$S \leftarrow p^a(S) \oplus (0^{320-k} \Vert K)$
Processing Associated Data	**Processing Associated Data**
if $\lvert A\rvert > 0$ **then**	**if** $\lvert A\rvert > 0$ **then**
$A_1 \dots A_s \leftarrow r$-bit blocks of $A \Vert 1 \Vert 0^*$	$A_1 \dots A_s \leftarrow r$-bit blocks of $A \Vert 1 \Vert 0^*$
for $i = 1, \dots, s$ **do**	**for** $i = 1, \dots, s$ **do**
$S \leftarrow p^b((S_r \oplus A_i) \Vert S_c)$	$S \leftarrow p^b((S_r \oplus A_i) \Vert S_c)$
$S \leftarrow S \oplus (0^{319} \Vert 1)$	$S \leftarrow S \oplus (0^{319} \Vert 1)$
Processing Plaintext	**Processing Ciphertext**
$P_1 \dots P_t \leftarrow r$-bit blocks of $P \Vert 1 \Vert 0^*$	$C_1 \dots C_{t-1}\tilde{C}_t \leftarrow r$-bit blocks of $C, 0 \le \lvert\tilde{C}_t\rvert < r$
for $i = 1, \dots, t-1$ **do**	**for** $i = 1, \dots, t-1$ **do**
$S_r \leftarrow S_r \oplus P_i$	$P_i \leftarrow S_r \oplus C_i$
$C_i \leftarrow S_r$	$S \leftarrow C_i \Vert S_c$
$S \leftarrow p^b(S)$	$S \leftarrow p^b(S)$
$S_r \leftarrow S_r \oplus P_t$	$\tilde{P}_t \leftarrow \lvert S_r\rvert_{\lvert\tilde{C}_t\rvert} \oplus \tilde{C}_t$
$\tilde{C}_t \leftarrow \lvert S_r\rvert_{\lvert P\rvert \bmod r}$	$S_r \leftarrow S_r \oplus (\tilde{P}_t \Vert 1 \Vert 0^*)$
Finalization	**Finalization**
$S \leftarrow p^a(S \oplus (0^r \Vert K \Vert 0^{320-r-k}))$	$S \leftarrow p^a(S \oplus (0^r \Vert K \Vert 0^{320-r-k}))$
$T \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$	$T^* \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$
return $C_1 \Vert \dots \Vert C_{t-1} \Vert \tilde{C}_t, T$	**if** $T = T^*$ **return** $P_1 \Vert \dots \Vert P_{t-1} \Vert \tilde{P}_t$ **else return** $\perp$

Fig. 1. Ascon Algorithm [13].

Ascon provides security against collision attacks, pre-image attacks, length extension attacks and second-preimage attacks for long messages, misuse attacks, side channel, implementation attack [13], [14], [11].

C. The Distributed OTP Concept [15]

OTP (One-time-password) is a password that changes with each session because it is based on the various counter. This technique is based on an out-of-band exchange of information, such as the keys, counter, and hash functions used in the password calculation process. Thus, OTP has considered one of the most secure authentication techniques because replay attacks are impossible (Fig. 3). The algorithm of the OTP consists of hashes and several calculations , which makes it difficult to be guessed.

The proposal uses lightweight protocols such as Blake 2, a hash function that achieves efficient security results. Blake2 generates hash digests based on stream encryption [16], [17]. The Blake function algorithm uses a 16-word constant combined with the message, salt value, and an initial vector and generates a 4x4 matrix. Then the matrix rows go through eight sets of permutations, and combinations [18], [19].

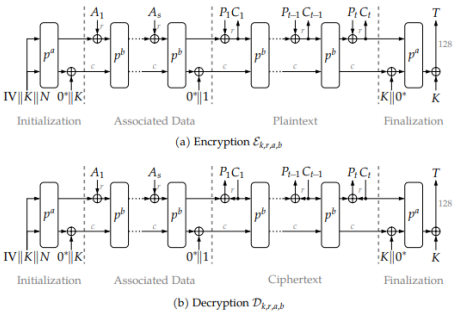

Fig. 2. Ascon Encryption and Decryption Processes [13].

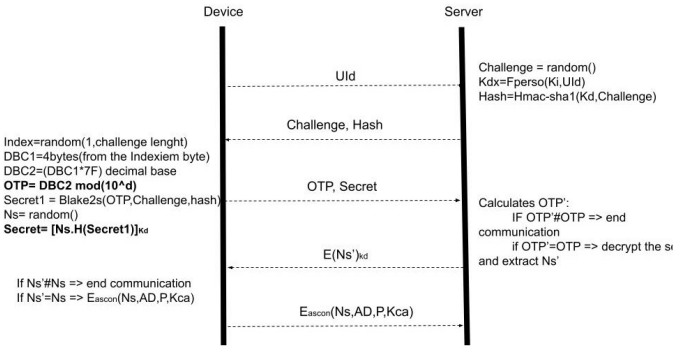

Fig. 3. Distributed OTP Authentication Protocol.

IV. Proposed Protocol

Each wireless communication starts with an association step where the communicating objects identify each other through parameters such as identifiers or passwords. At this level, if these parameters are disclosed or intercepted by a malicious third party, the network will suffer an intrusion and lose its security. Hence, the importance of introducing and deploying secure authentication protocols for these communications, such as the proposed protocol which addresses wireless communication security at the NAN level of smart grids between the smart meters and the Gateways/concentrators of the grid.

A. Proposed Protocol Model

The proposed solution aims at designing a security protocol that ensures authentication, authenticity, integrity, and confidentiality. This proposal aims to complete the proposal of [15] while boosting its security by modifying some parameters. As shown in Fig. 4, the protocol is divided into three main parts. The communication starts with an authentication of the Device based on the OTP. Then the protocol moves on to the phase of mutual authentication by the server, based on a random value calculated by the Device. Finally, as soon as both entities are authenticated, they move on to secure the communication channel through Ascon encryption to ensure confidentiality, integrity, and authenticity.

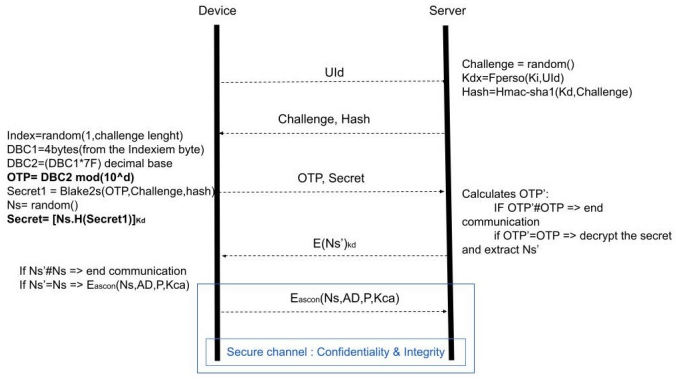

Fig. 4. Proposed Scheme.

1) Pre-Phase: Keys Exchange: This phase is an off-line phase. The symmetric key Kia is affected to the Device and the server by a trusted authority. The secret key Kd is configured on the Device in an out-of-the-line mode based on its Id. In addition, the server gets its secret Key Ki. It keeps communication with an external base where Ids and their appropriate Keys will be stored. The base will avoid attackers trying to impersonate an already authenticated Device.

2) Device Authentication :

- Step 1: The Device will start the communication request by sending its Id to the server.
- Step 2: Once the Server receives the identifier, he will check first if it is already registered. If the Device is already registered and active, it will be dropped. Else, the server will generate a random value as a challenge and calculate the corresponding Key to the Device and a hash value with the HMAC-SHA1 of the calculated key Kd and the challenge. Then he will send the Hash value and the challenges to the Device.
- Step 3: The Device will start calculating the OTP. For this purpose, he will start calculating an Index value, a random value between 1 and the received challenge length. Then, it will truncate the 4 bytes starting from the index byte of the received hash. Next, it will calculate the DBC2 as a decimal value of the multiplication of the DBC1 and 7F. OTP is finally found by applying a modulo d to force the OTP length to ”d .”In addition, the Device will calculate a secret value based on Blake2s of the OTP and the last received values of Challenge and Hash.
- Step4: The Server calculates the OTP on its own and compares it to the received value. If the OTP is correct, the server will move to its authentication. If not, it will stop the communication.

3) Server Authentication :

- Step 1: This step starts simultaneously with the third step of the device authentication. While sending its OTP, the Device also sends a secret containing an Ns Value. The secret is a random value Ns encrypted and its concatenation with the blake2s hash of the OTP, the challenge, and the received hash.
- Step 2: The Server will start by authenticating the Device based on its OTP. Once the Device is authorized, the OTP will try to extract the Ns value from the received expression. If the server is right, it can decrypt the secret value. Then, since it could authenticate the Device, it will have the true value of the OTP and the keys required to do the reverse calculation and extract the correct Ns. Thus, we will gain in the number of exchanges and in time.
- Step 3: The Device will start authenticating the server by the Ns value. If the value is correct, they will move to secure the challenge; if not, the communication is stopped.

4) Channel Security: Confidentiality and Integrity: In order to ensure the appropriate security of the channel, we will use authenticated encryption with associated data. The main feature of this step is that a lightweight AEAD cipher encrypts the data. The Device will start encryption based on the Plaintext, the associated data (related to the time of transmission to avoid revealing the location in the IoT context that involve the privacy of the clients), the secret value Ns, and the symmetric key Kia (pre-shared between the Device and the server). Thus, confidentiality, authenticity, and Integrity are ensured, and the data is protected.

V. Security Analysis

A. Informal Analysis

1) Security Considerations:

- Efficiency:
  All used protocols in the proposal are lightweight protocols that respect the limitations of constrained devices. Blake2s and Ascon are two protocols for hashing and encryption adapted to the specifications of connected objects, which has allowed them to be outclassed in NIST competitions. In addition, OTP's computational distribution method offers more advantages to objects in terms of reduced computation and storage in objects.

- Authentication:
  The protocol consists of mutual authentication. Both the Device and the server authenticate to each other. The device authentication is based on the OTP, which is known for a high level of security. In addition, server authentication requires that it compute the OTP value and compare it to the value computed by the Device. It can also decrypt the mathematical expression to extract the received nonce that varies with each session. So, mutual authentication is ensured, safe, lightweight, and robust.

- Data Confidentiality and Integrity:
  Thanks to Ascon CIPHER's potential, it was the first candidate for the final round of the CAESAR competition. Ascon is a lightweight authenticated encryption protocol that was evaluated by many designers who prove that it is secure against many attacks on confidentiality and integrity of data and identities [13], [14].

- One Point of Failure:
  Authentication is based on the distributed calculation of OTP, which means that calculation is not concentrated. Also, both the Device and the server have secret and random parameters that are not shared clearly. Each one needs to make the distributed calculation to authenticate to the other. In other words, the knowledge of the Device and the receiver are not the same, which means there is no point in failure in the protocol.

2) Resistance against Threats and Attacks :

- Impersonation Attack:
  The attacker cannot impersonate the Device because the authentication is based on the OTP that changes every session. Also, the server registers the Id devices with their correspondent OTP to avoid malicious devices that would try to impersonate the Device. The attackers cannot either impersonate the server. At the same time, the Ns value is generated in every session. Its calculation is based on ulterior parameters, particularly the secret Key Ki, which is not exchanged in the channel.

- Eavesdropping:
  In the secure exchange step, each piece of data is protected by the Ascon AEAD. The Device uses the Ns nonce shared secret with the server and the associated data related to the precise transmission time. Combining these data allows us to verify the integrity and validity of the message securely. Thus, the attacker can neither listen nor modify the messages.

- MITM (Man in the Middle):
  The proposed security scheme emphasizes strong mutual authentication, also known as two-way authentication. The Device and the server identify each other with this authentication process before starting the data exchange. Without knowing the Device's private Key, the random and unique challenge, and nonce Ns and the corresponding computations, an attacker cannot compute the authentication or encryption data and therefore cannot validate the authentication and proceed to the data exchange. Thus, the Device and the server ensure that they communicate with legitimate correspondents. Therefore, the Man in the Middle attack is impossible in this scenario.

- Forward Secrecy:
  The leakage of a secret key does not affect the rest of the communication. All used keys are combined with the session variable. Even if an attacker intercepts a key, he will not be able to compromise privacy or confidentiality because the Key alone is insufficient to make an attack. In addition, the used keys are even pre-shared or require specific calculations. Hence, a non-legitimate participant has no way to compromise the forward secrecy.

- Forgery Attack:
  An intruder will not be able to authenticate and forge an appropriate request while he has to own the Id and the secret Key Kd. In addition, as we mentioned before, the scheme resists the replay attack and provides mutual authentication. Hence, a forgery attack is prevented.

- Replay Attack:
  In the authentication phase, the OTP and the random parameters Challenge, Ns varies every session. Then, in the transmission phase, the associated data is related to timestamps, which prevents an illegitimate entity that intercept communication from reproducing the same parameters and authenticates to one of the entities.

B. Formal Analysis: Avispa

We perform the formal analysis of the proposed protocol with the AVISPA simulation tool [20] to prove that it is safe and robust against MITM and replay attacks.

The AVISPA tool uses the HLPSL language to make security simulations. It allows the designers to verify their protocols' security and robustness.

1) CAS+: CAS+ is a simple syntax based on the Alice and Bob notation. While CAS+ has a simpler syntax than the HLPSL [20], we made the basic CAS+ algorithm, presented in the Fig. 5 that will be translated later to the HLPSL. The CAS+, even if it is simpler than the HLPSL, the CAS+ is not as performant as the HLPSL language. Also, in some cases, the translator cannot translate correctly [21], [20] Consequently, we started with the CAS+ algorithm. We made the required modifications and added the additional specifications of the protocol directly to the HLPSL file.

```
protocol OurProposition;
identifiers
Device, Server : user;
Challenge,Id,OTP,X,Ns,P,AD,Ok : number;
Kd, Kca : symmetric_key;
H,E : function;
messages
1. Device -> Server : Device,Id
2. Server -> Device : {Challenge,X,Id}Kd
3. Device -> Server : {OTP,H(Challenge,X)}Kd
4. Device -> Server : {Ns,H(Challenge,X,Id)}Kd
5. Server -> Device : {Ns}Kd
6. Device -> Server : {Ns,AD,P}Kca
knowledge
Device : Device, Server, Kd, Kca, Id,P,Ns,AD,
OTP;
Server : Device, Server, Kd, Kca, OTP,X,AD,
Challenge;
session_instances
[Device:Alice,Server:bob,Id:id,Kd:kd,Kca:kca]
[Device:Alice,Server:bob,Kd:kd,
Challenge:challenge,
X:x,Kca:kca];
intruder_knowledge
Alice,bob,challenge,x,id;
goal
Device authenticates Server on Ns;
Server authenticates Device on OTP;
secrecy_of P [Server,Device] ;
```

Fig. 5. Cas+ : proposed solution

2) HLPSL: SPAN is the tool that provides the translation CAS+/HLPSL and also makes the protocol simulation and verification [21], [20] Based on the CAS+ translation made with the SPAN, we added the proposed advanced functions and defined the goals and verifications. We started by defining the two roles, Server and Device. Then, we set up the sessions, environment, verifications, and security goals. Regarding the exchanged messages, we worked on the state's expressions to make actions look alike the proposed protocol. Advanced calculations like truncate and multiplications were performed as concatenations and Xor. We set goals as the verification of the mutual authentication based on OTP and Ns. In addition, we set verification of the secrecy of the plaintext P.

3) Protocol Simulation: Once the HLPSL is executable by SPAN, we get the diagram in Fig. 6. This diagram proves that the whole protocol is readable by the verifications and that the security verifications cover the whole protocol. we moved then to launch the security verifications by the OFMC, and ATSE [22]

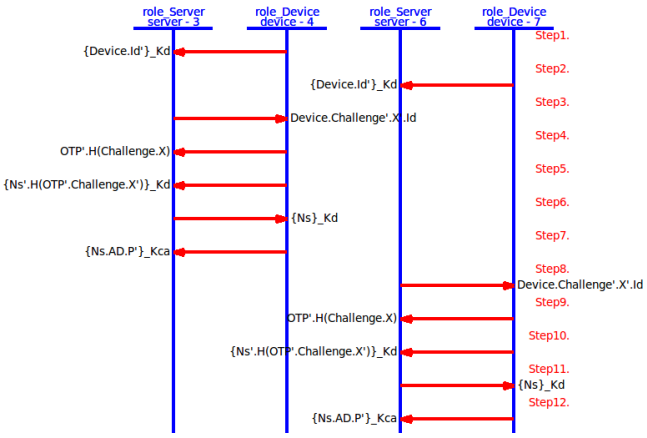

Fig. 6. AVISPA-Span Simulation.

4) Execution Results: In order to evaluate protocols and verify whether the goals set out in HLPSL's algorithm are verified, the backends execute the protocol through multiple finite iterations until the protocol is considered safe for the number of sessions or an attack is found. The OFMC and CL-AtSe backends check if a legitimate entity can execute the protocol correctly while introducing a passive attacker in Fig. 7 and Fig. 8. The four backends of the AVISPA are:

- On-the-fly Model-Checker(OFMC): This uses several symbolic techniques to represent the state-space to perform protocol falsification and verification for the boundless number of sessions in a demand-driven fashion [23].
- Constraint-Logic-based Attack Searcher (CLAtSe): This is a constraint-based approach. It uses some simplification and redundancy elimination techniques to integrate a new specification for cryptographic

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/0904f.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.03s
  visitedNodes: 18 nodes
  depth: 4 plies
```

Fig. 7. OFMC Verification Results.

functions [23].

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/0904f.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 38 states
  Reachable  : 8 states
  Translation: 0.02 seconds
  Computation: 0.00 seconds
```

Fig. 8. ATSE Verification Results.

- Satisfiability-based Model-Checker (SATMC) and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP): SATMC and TA4SP results were inconclusive because they do not support Xor and modulus operators [24].

VI. Performance Analysis

we will evaluate each participant's computation and communication performance. We will separate the Device's computations and those made by the servers because of the different capacities of each.

Table I presents the different costs of the system's entities. While gateways are more powerful than devices in capacities and resources, we separate calculations to assess the protocols' weight on the constrained entity.

T refers to The time taken while (h: hashing, e: encryption, r: random, c: ascon cipher, and other: small calculations such as multiplications, truncating, and XORing). The storage is expressed by the estimated size of each parameter involved in the communication.

Table II depicts different costs generated from the three phases of the protocol in comparison with similar works that address the security of the communication between smart meters and the neighbor gateways.

As aforementioned in Table I , the proposition requires less storage than the other works. Communication cost is regular but not the best, while the [28]'s work has better results. Regarding computation, the parameter that makes a difference is the Tc related to Ascon cipher performance. However, the difference in encryption requirement of all other works is more than the double value of the four encryptions used in this work. In addition, Ascon is a lightweight encryption cipher that is more optimized than the standard encryption algorithm.

TABLE I. Costs Performance Comparison
(CompC:Computational Cost; ComC: Communication Cost

Entity	CompC	ComC	Storage
Device/SM	2Th+2Te+1Tr+3To+1Tc	2	516
Server/Gateway	1Th+2Te+1Tc+4To	2	844
Total	4Th+4Te+2Tc+7o	4	1360

VII. Discussion

Due to the multiple risks that threaten the smart metering system, accessible from different entities, We estimate that the risks are related to privacy and integrity authenticity. Therefore, we claim that the proposed solution is adapted to the smart metering context. The message size of smart meters is around 100bits and thus covered with the selected protocols, namely Ascon and blake2. In addition, the distributed calculation of OTP respects the constrained nature of smart meters. On the other hand, we claim the mutual authentication that requires higher costs by the side of the gateway does not affect the robustness of the protocol, while the gateway is not constrained in terms of resources. In terms of security, combining the selected protocols with the OTP distribution covers all well-known attacks against the Metering system of the smart grids.

The solution could be extended to other IoT applications. However, the obvious limitation is mainly related to energy aspects which have not been addressed in the scope of the proposed solution since smart meters are always powered by continuous power. However, it is worth considering this issue in a sustainable and ecological approach.

VIII. Conclusion and Perspectives

In this article, we have proposed a protocol that considers the three pillars of security: authentication, confidentiality, and Integrity. We started from the principle of distributing OTP for authentication, which alleviates the objects and respects their constraints in terms of performance. We have carefully chosen secure and lightweight cryptographic algorithms Ascon Cipher and Blake2, two finalists of the competitions launched by NIST, to effectively choose lightweight protocols. The informal and formal security evaluation through AVISPA revealed that the protocol is secure. The comparison has shown that the protocol is better optimized than other similar proposals. However, the costs may seem to be a drawback to the protocol's performance which needs more optimization, especially in the context of smart meters requiring real-time communications.

In addition, the generic architecture of smart metering systems can be a key for deploying innovative technologies within the overall architecture to boost performance and security more efficiently. Hence, it is worth investigating new and

TABLE II. Costs Performance Comparison
(CompC:Computational Cost; ComC: Communication Cost)

works	CompC	ComC	Storage (Kb)
Proposition	4Th+4Te+2Tc+7o	4	1.3
[25]	17Th+10Te	7	NA
[26]	10Th+1Te	NA	2.6
[27]	96Th+10Te+1To	NA	4
[28]	8Th+8Te+2To	2	3.7

powerful technologies such as AI and blockchains, which have advantages in terms of time, efficiency, security, and lightning of constrained resources.

REFERENCES

[1] M. Meliani, A. E. Barkany, I. E. Abbassi, A. M. Darcherif, and M. Mahmoudi, "Energy management in the smart grid: State-of-the-art and future trends," *International Journal of Engineering Business Management*, vol. 13, p. 18479790211032920, 2021.

[2] V. Kumar, R. Kumar, and S. K. Pandey, "Lkm-ami: a lightweight key management scheme for secure two way communications between smart meters and han devices of ami system in smart grid," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 82–100, 2021.

[3] B. Nithya, J. C. Mathew, G. Kavya, N. Anutha, and A. Kumari, "An analysis on cryptographic algorithms for handling network security threats," in *2022 IEEE Delhi Section Conference (DELCON)*. IEEE, 2022, pp. 1–9.

[4] A. K. Chattopadhyay, A. Nag, D. Ghosh, and K. Chanda, "A secure framework for iot-based healthcare system," in *Proceedings of International Ethical Hacking Conference 2018*, M. Chakraborty, S. Chakrabarti, V. E. Balas, and J. K. Mandal, Eds. Singapore: Springer Singapore, 2019, pp. 383–393.

[5] M. T. Hammi, "Sécurisation de l'internet des objets," Ph.D. dissertation, Université Paris-Saclay (ComUE), 2018.

[6] M. Agrawal, J. Zhou, and D. Chang, "A survey on lightweight authenticated encryption and challenges for securing industrial iot," in *Security and Privacy Trends in the Industrial Internet of Things*. Springer, 2019, pp. 71–94.

[7] M. Mumtaz, J. Akram, and L. Ping, "An rsa based authentication system for smart iot environment," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2019, pp. 758–765.

[8] A. Zambroni de Souza and M. Castilla, "Microgrids design and implementation," 2019.

[9] A. E. Ibhaze, M. U. Akpabio, T. O. Akinbulire *et al.*, "A review on smart metering infrastructure," *Int. J. Energy Technology and Policy*, vol. 16, no. 3, p. 277, 2020.

[10] G. Shay, J. Ashwin, and N. Mridul, "counter mode encryption with authentication tag nist 2020," Sep 2020. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/comet-spec.pdf

[11] S. Koteshwara, A. Das, and K. K. Parhi, "Fpga implementation and comparison of aes-gcm and deoxys authenticated encryption schemes," in *2017 IEEE International symposium on circuits and systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[12] NIST, "Caesar: Competition for authenticated encryption: Security, applicability, and robustness"."

[13] d. christoph, e. aria, m. florian, and s. martin, "Ascon v1.2. submission to nist - csrc," Sep 2019. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf

[14] Z. Çamur, "A study of lightweight cryptography," Master's thesis, Middle East Technical University, 2020.

[15] H. E. Makhtoum and Y. Bentaleb, "An improved iot authentication process based on distributed otp and blake2," 2021.

[16] J.-P. Aumasson, W. Meier, R. C. Phan, and L. Henzen, "The hash function blake," 2014.

[17] H. EL Makhtoum and Y. Bentaleb, "Comparative study of keccak and blake2 hash functions," in *Networking, Intelligent Systems and Security*. Springer, 2022, pp. 343–350.

[18] V. Rao and K. Prema, "Comparative study of lightweight hashing functions for resource constrained devices of iot," in *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, vol. 4. IEEE, 2019, pp. 1–5.

[19] ——, "Light-weight hashing method for user authentication in internet-of-things," *Ad Hoc Networks*, vol. 89, pp. 97–106, 2019.

[20] A. Team *et al.*, "Hlpsl tutorial: A beginners guide to modelling and analysing internet security protocols," *Information Society Technologies*, 2006.

[21] T. Genet, "A short span+ avispa tutorial," Ph.D. dissertation, IRISA, 2015.

[22] P. R. Yogesh *et al.*, "Formal verification of secure evidence collection protocol using ban logic and avispa," *Procedia Computer Science*, vol. 167, pp. 1334–1344, 2020.

[23] M. Singh, M. Ranganathan *et al.*, "Formal verification of bootstrapping remote secure key infrastructures (brski) protocol using avispa," 2020.

[24] A. Javed, "Formal analysis of cwa 14890-1," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2011, pp. 314–335.

[25] V. Abreu, A. O. Santin, E. K. Viegas, and V. V. Cogo, "Identity and access management for IoT in smart grid," in *Advanced Information Networking and Applications*, L. Barolli, F. Amato, F. Moscato, T. Enokido, and M. Takizawa, Eds. Springer International Publishing, 2020, vol. 1151, pp. 1215–1226.

[26] N. Saxena, B. J. Choi, and R. Lu, "Authentication and authorization scheme for various user roles and devices in smart grid," vol. 11, no. 5, pp. 907–921, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7366583/

[27] Y. Liu, C. Cheng, T. Gu, T. Jiang, and X. Li, "A lightweight authenticated communication scheme for smart grid," vol. 16, no. 3, pp. 836–842, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7295548/

[28] L. Wu, J. Wang, S. Zeadally, and D. He, "Anonymous and efficient message authentication scheme for smart grid," vol. 2019, pp. 1–12, 2019. [Online]. Available: https://www.hindawi.com/journals/scn/2019/4836016/