Design of an Anonymous Lightweight Communication Protocol for Smart Grid and Its Implementation on 8-bit AVR and 32-bit ARM

Dariush Abbasinezhad-Mood, Arezou Ostad-Sharif, and Morteza Nikooghadam

(Corresponding author: Dariush Abbasinezhad-Mood)

The Department of Computer Engineering and Information Technology, Imam Reza International University
Razavi Khorasan Province, Mashhad, Sanabaad, Daneshgah Avenue, Mashhad, Iran
(Email: dariush.abbasinezhad@imamreza.ac.ir.)

(Received Sept. 12, 2017; revised and accepted Oct. 13, 2018; First Online June 1, 2019)

Abstract

Upgrading the conventional electrical grid to smart grid offers more efficiency, resiliency, and reliability. Thus, the smart grid adoption is essential in today's modern countries and the information age. In smart grid, consumption reports are gathered from smart meters and sent to the control center and some control messages are sent vice versa. These bidirectional communications are subject to various security challenges. Because of the constrained resources of smart meters, employing lightweight communication protocols is critical. For this purpose, recently, scholars have proposed several lightweight communication protocols. Nonetheless, most of these protocols are not anonymous or fail to assuage the entire desired security features. Therefore, in this paper, we propose an efficient communication scheme that not only is anonymous, but also can thwart the well-known attacks. Our actual hardware performance analysis, which has been done on both 8-bit AVR and 32-bit ARM microcontrollers, confirms the outperformance of the proposed scheme.

Keywords: ARM; AVR; Lightweight; Secure Communications; Smart Grid Security

1 Introduction

The legacy energy grid cannot fulfil the actual needs of today's modern countries and the information era [15, 20]. Hence, in near future, the adoption of smart grid (SG) will become so critical as it promises to offer more efficiency, resiliency, and reliability [2]. In *SG*, the smart meters (SMs), which are some resource-constrained electronic measurement devices, gather the energy usage of consumers and send them to the control center via some intermediary gateways, such as neighborhood gateways (NGs) [3]. The communications of *SM*s and *NG*s are bidirectional and the *NG*s may also send some commands to *SM*s [4]. As these two-way communications are susceptible to several security threats, proposing secure communication protocols is vital. Evidently, overlooking the security concerns will hamper the wide adoption of *SG*. The security needs to be fully considered from the very beginning of usage reports collection by *SM*s up to their reception at the control and power management center. Further, the constrained resources of *SM*s in terms of flash storage and computational capability should be fully taken into consideration [4, 20].

The security challenges have taken much attention from the academia [11, 17, 26], and same as other fields [9, 12–14, 16, 18, 23–25, 27], for *SG*, many scholars have presented key agreement schemes [2, 5, 6, 10, 21, 22, 29] and secure communication protocols [4, 19, 20, 28]. Nevertheless, careful assessment of the related works shows that the existing protocols cannot totally fulfil the desired security properties. As an example, most of the existing communication schemes cannot provide the anonymity, a feature that helps to better preserve the privacy of consumers.

1.1 Related Work

In 2011, Fouda *et al.* [10] addressed the traffic analysis, denial of service (DoS), *DoS* buffer overflow, spoofing, reconfigure, man in the middle, and replay attacks as the security threats that exist in *SG* communications. Further, they presented a lightweight message authentication protocol and indicated that their scheme can provide semantic-secure shared key, mutual authentication, and an encrypted channel for successive communications. In 2013, Li *et al.* [19] put forward an authenticated communication scheme called *AC* using the Merkle hash tree. Although the presented scheme by Li *et al.* has a proper level of performance, it requires lots of space for storing the generated parameters. In addition, in [19], the authors have indicated that their scheme can thwart the replay, message injection, message analysis, and message

modification attacks. Nonetheless, their scheme cannot resist the pollution or *DoS* attack. In 2016, Liu *et al.* [20] have proposed another lightweight authenticated communication protocol named *LAC* that has a better performance than *AC* in terms of storage space, communication overhead, and computational cost. However, same as *AC*, *LAC* fails to withstand the pollution attack and needs lots of space for storing generated parameters. At the same year, two other schemes have been proposed by Mahmood *et al.* [21] and Uludag *et al.* [28]. Mahmood *et al.* have mainly concentrated on the authentication and key agreement and Uludag *et al.* have proposed a holistic scheme consisting of both key establishment and data collection. Quite recently, to remedy the challenges of Li *et al.*'s scheme [19] and Liu *et al.*'s scheme [20], we have proposed an ultra-lightweight scheme for communications of *SM*s and *NG*s [4]. However, none of these schemes can offer the anonymity. Therefore, in this paper, we try to propose an anonymity-preserving protocol that can withstand the well-known attacks and has a proper level of performance to be executed on the resource-constrained *SM*s.

1.2 Motivation

First, as stated earlier, Li *et al.*'s [19] and Liu *et al.*'s [20] schemes suffer from the pollution attack. Second, as mentioned in [4, 19], the next generation of *SM*s should be able to send the consumption reports in one-minute or less time intervals. In this case, the presented schemes by Li *et al.* [19] and Liu *et al.* [20] would be impractical as the required storage space will exceed the flash storage of most popular low-cost microcontrollers. Third, most of the existing communication protocols are not anonymous or fail to withstand the well-known attacks like the *SM* memory modification attack. Fourth, more efficient the presented communication protocol, more its suitability to be performed on *SM*s. Finally yet significantly, the *ARM* microcontrollers are one of the most cost-effective and energy-efficient *MCU*s in the market and it will be useful to test the performance of different cryptographic operations on them in comparison to their counterparts in *AVR*. These facts motivated us to propose a communication protocol that can assuage the mentioned necessities.

1.3 Contribution

The fourfold contribution of this paper is as follows.

- Presenting a communication scheme, which (a) is secure against the well-known attacks, (b) provides the anonymity, and (c) is much more efficient than several recently published schemes.
- Presenting the details of shared key storage and retrieval both in the *SM* flash storage and the *NG* database.
- Reducing the number of parameters need to be stored in the *SM* flash storage to only two parameters.

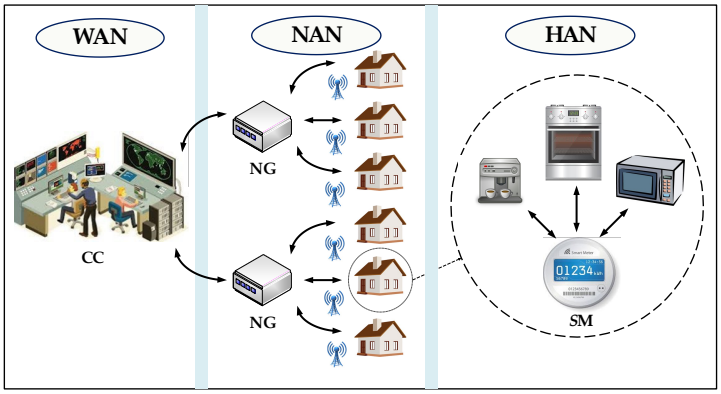

Figure 1: The network model of SG

- Implementation of different cryptographic operations on both 8-bit *AVR* and 32-bit *ARM*.

The remainder of this paper is structured as follows. The network model, attack model, and design objectives are explained in Section 2. In Section 3, the proposed anonymous communication scheme is delineated. In Sections 4 and 5, the descriptive security analysis and automatic formal verification of security using ProVerif are presented, respectively. The efficiency analysis and experimental study are presented in Section 6. Finally, Section 7 concludes this paper.

2 Models and Design Objectives

2.1 Network Model

The communication model of *SG* has been presented in several papers [4,7,20]. As shown in Figure 1, a smart metering communication system consists of *SM*s, which are responsible for energy usage measurement and collection; home area network (HAN), which is a network formed by an *SM* and its controlling smart appliances, neighborhood area network (NAN), which is composed of hundreds of *SM*s and collects information from multiple *HAN*s; wide area network (WAN), which is referred to as the backhaul and carries metering data to the control center (CC); and gateways, which gather energy consumption reports from several *HAN*s. The focus of this paper is on the secure bidirectional communications of *SM*s and *NG*s.

2.2 Attack Model

In this paper, we assume three kinds of adversary as follows.

1) **The external adversary.** The external adversary ε can eavesdrop or alter the exchanging messages between *SM*s and *NG*s and can perform replay, *DoS*, message injection, message modification, and message analysis attacks.

2) **The internal adversary.** The internal adversary $\jmath$ not only has the power of the external adversary, but

also can gain access to the stored records of the *NG* database.

3) **The global adversary.** The global adversary φ not only has the power of the internal adversary, but also can read the content of *SM*s flash storage.

2.3 Design Objectives

In this paper, we aim to propose a two-way anonymous communication scheme which can fulfil the following goals.

1) **Anonymity.** The proposed communication protocol should be designed such that an adversary cannot distinguish the real identifier of *SM*s.

2) **Near real-time authentication.** Each *SM* must be able to check that the received message has been sent from the authorized *NG* and nobody has impersonated *NG*. Similarly, *NG* must be able to check that the received message has been sent from the intended *SM*. Both of these actions should be done in a short amount of time.

3) **Confidentiality.** The exchanging messages between *SM*s and *NG*s must only be accessed by the intended party. That is to say, except authorized *SM*s and *NG*s, nobody else must be able to gain access to confidential messages.

4) **Message modification attack resistance.** Both *SM*s and *NG*s must be able to check whether or not a received message has been altered by an adversary during the transfer.

5) **Message injection attack resistance.** Both *SM*s and *NG*s must be able to filter the fabricated messages that may be sent by an attacker.

6) ***DoS* attack resistance.** Both *SM*s and *NG*s must be able to detect the modifications that make services unavailable.

7) **Message analysis attack resistance.** The adversary must not be able to recover the consumption reports or control messages by just eavesdropping the exchanging messages.

8) **Replay attack resistance.** Both *SM*s and *NG*s must be able to verify that a valid message is not a repeated one.

9) **Insider attack resistance.** In this paper, we assume the insider as a person who can easily gain access to the *NG* database and an insider attacker as an adversary who is an insider. The proposed scheme must be able to resist the attacks that may be performed by an insider attacker.

10) ***SM* memory modification attack resistance.** The protocol must be designed such that the stored data in the flash memory of *SM*s be kept confidential and even if an adversary alters them, the tampering could be revealed so soon.

11) **Low storage and computational costs.** Due to the constrained resources of *SM*s, the proposed protocol must be as lightweight as possible.

3 Proposed Lightweight Scheme

In this section, a complete description of the proposed anonymous lightweight communication scheme is given. Our scheme can be employed effectively for secure bidirectional communications of *SM*s and *NG*s in *SG*. In our scheme, each day, every 15 minutes, the i^{th} *SM*, SM_i, measures consumption report D_j^i, where $j = 1, 2, \ldots, 96$, and sends it to *NG*. Meanwhile, *NG* may send four control messages CM_k^i, where $k = 1, 2, 3, 4$, in order to be performed by SM_i. Here, same as the other related schemes, we consider 15 minutes time intervals for consumption reports collection. However, in the "efficiency analysis and experimental study" section of this paper, we will evaluate the schemes according to different time intervals.

Our scheme is composed of three phases, namely "initialization," "shared key generation and storage," and "secure message transmission." In the following subsections, we elaborate each phase. The notations used in our scheme together with their definitions have been listed in Table 1.

Table 1: Notations and their definitions

Notation	Definition
SM_i	i^{th} smart meter
NG	neighborhood gateway
ID_i	identifier of SM_i
ID_{NG}	identifier of NG
m_i	secret key of SM_i
s	secret key of NG
K_i^{NG},K_i^{SM}	shared key between NG and SM_i
D_j^i	j^{th} usage report of SM_i
CM_k^i	k^{th} control message for SM_i
T_i	j^{th} timestamp of data collection
CT_i	current time of SM_i
CT_{NG}	current time of NG
Enc	symmetric encryption
Dec	symmetric decryption
Δt	predefined maximum transmission delay

3.1 Initialization

In this phase, for each *SM*, *NG* first generates a random number r_i, then, computes encrypted identifier EID_i as

Equation (1), where Enc_s is the symmetric encryption using the key s, s is the private key of NG, and ID_i is the identifier of SM_i. Finally, NG sends the generated EID_i to SMs through a reliable medium.

$$EID_i = Enc_s(ID_i \,\|\, r_i). \tag{1}$$

3.2 Shared Key Generation and Storage

In this phase, SM_i and NG share a key K_i by running the proposed protocol in [2]. The following steps are done after the shared key generation.

Step 1. NG first calculates H_i^{NG} and E_i^{NG} as Equations (2) and (3), next, it adds (ID_i, E_i^{NG}, H_i^{NG}) record to its database. Here, K_i^{NG} is the NG side K_i.

$$H_i^{NG} = h(K_i^{NG} \,\|\, ID_i), \tag{2}$$
$$E_i^{NG} = K_i^{NG} \oplus h(s). \tag{3}$$

Step 2. SM_i first computes E_i^{SM} as Equation (4), where K_i^{SM} and m_i are the SM_i side K_i and SM_i's private key, respectively. Afterwards, it stores E_i^{SM} and EID_i in its flash storage. It should be noted that K_i^{NG} and K_i^{SM} are identical and we have named them differently to make them distinguishable.

$$E_i^{SM} = K_i^{SM} \oplus h(m_i). \tag{4}$$

Step 3. According to the security policies of system, the shared key K_i can be updated by rerunning the presented protocol in [2].

3.3 Secure Message Transmission

In this phase, every 15 minutes, SM_i sends usage report D_j^i to NG and meanwhile, NG may send 4 control messages CM_k^i to SM_i. The following steps are done in this phase. An illustration of this phase is depicted in Figure 2.

1) SM_i **to** ***NG*** **message transmission**

Step 1. SM_i first retrieves E_i^{SM} from its flash memory, then, computes K_i^{SM} as Equation (5).

$$K_i^{SM} = E_i^{SM} \oplus h(m_i). \tag{5}$$

Step 2. SM_i computes verifier V_j^i as Equation (6). This verifier will be used by NG for checking the message integrity, the SM_i authentication, and SM_i memory modification attack check. Here, T_j is the j^{th} timestamp of data collection.

$$V_j^i = h(D_j^i \oplus T_j \oplus ID_i). \tag{6}$$

Step 3. SM_i computes M_j^i as Equation (7).

$$M_j^i = Enc_{K_i^{SM}}(D_j^i). \tag{7}$$

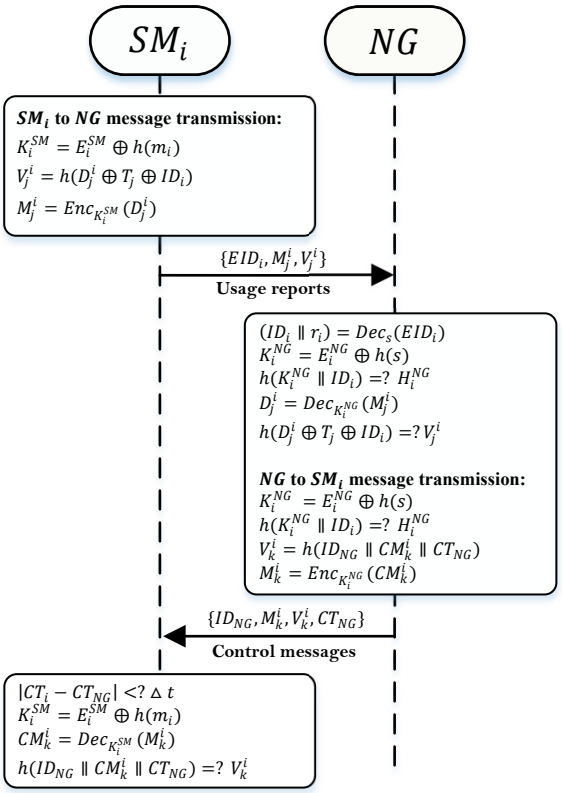

Figure 2: Secure message transmission phase of the proposed scheme

Step 4. SM_i sends $\{EID_i, M_j^i, V_j^i\}$ to NG. Here, there is no need to send timestamp T_j since NG can obtain an identical one. We refer readers to [4] for the detailed discussion.

Step 5. Upon receiving the message $\{EID_i, M_j^i, V_j^i\}$, NG first decrypts EID_i to achieve ID_i. Then, according to the obtained ID_i, retrieves H_i^{NG} and E_i^{NG} from its database. Finally, achieves K_i^{NG} as Equation (8) and checks whether $h(K_i^{NG} \,\|\, ID_i) = H_i^{NG}$ holds to ensure the integrity of ID_i, E_i^{NG}, and H_i^{NG}.

$$K_i^{NG} = E_i^{NG} \oplus h(s). \tag{8}$$

Step 6. NG obtains usage report D_j^i as Equation (9).

$$D_j^i = Dec_{K_i^{NG}}(M_j^i). \tag{9}$$

Step 7. Using its current time, NG first achieves timestamp T_j, then checks whether $h(D_j^i \oplus T_j \oplus ID_i) = V_j^i$ holds or not in order to ensure the message has been sent from the intended SM_i, the message has not been altered during its transfer, the message is not a repeated one, and the SM_i memory has not been changed.

Step 8. NG compares D_j^i with predefined format and if it conforms, accepts D_j^i from SM_i.

2) ***NG*** **to** SM_i **message transmission**

Step 1. NG picks intended control message CM_k^i that is needed to be performed by SM_i, where k = 1,2,3,4.

Step 2. NG retrieves H_i^{NG} and E_i^{NG} corresponding to ID_i from its database and achieves K_i^{NG} as Equation (8).

Step 3. NG checks the equality of $h(K_i^{NG} \| ID_i) = H_i^{NG}$ to ensure the integrity of ID_i, E_i^{NG}, and H_i^{NG}.

Step 4. NG computes verifier V_k^i as Equation (10) that will be used by SM_i for the message integrity check, the NG authentication, and SM_i memory modification attack check. Here, CT_{NG} is the current time of NG.

$$V_k^i = h(ID_{NG} \| CM_k^i \| CT_{NG}) \quad (10)$$

Step 5. NG computes M_k^i as Equation (11) in order to ensure that the control message CM_k^i can only be accessed by SM_i.

$$M_k^i = Enc_{K_i^{NG}}(CM_k^i). \quad (11)$$

Step 6. NG sends $\{ID_{NG}, M_k^i, V_k^i, CT_{NG}\}$ to SM_i.

Step 7. Upon receipt of the message, SM_i checks whether $|CT_i - CT_{NG}| < \Delta t$ holds or not to ensure that the received message is not a repeated one. CT_i is the current timestamp of SM_i and Δt is a predefined maximum transmission delay.

Step 8. SM_i first retrieves E_i^{SM} from its flash memory, then, computes K_i^{SM} as Equation (5).

Step 9. SM_i achieves control message CM_k^i as Equation (12).

$$CM_k^i = Dec_{K_i^{SM}}(M_k^i). \quad (12)$$

Step 10. SM_i checks the equality of Equation (10) to ensure that the message has been sent from the authentic NG, it has not been changed during the transfer, and its memory has not been altered.

Step 11. SM_i checks the CM_k^i format, then executes it.

It is worth noting that to guarantee the strong anonymity of SMs, NG needs to generate a new random number and update the EID_i. The new generated EID_i can be sent to SM via a control message.

A feature-based comparison with similar recently-published schemes is presented in Table 2.

4 Security Analysis

According to our objectives and attack model, in this section, we present the security analysis of the proposed scheme. We indicate that our scheme not only can provide confidentiality and anonymity, but also is secure against the

a. Message analysis;
b. Impersonation;
c. Modification;
d. Injection;
e. Replay;
f. DoS;
g. Insider;
h. SM memory modification attacks.

The details are as follows.

4.1 Providing Confidentiality, Preserving Anonymity, and Message Analysis Attack Resistance

In our scheme, an attacker $\mathscr{A}$ (either external, internal, or global), who is eavesdropping the communication channels, can get access to $\{EID_i, M_j^i, V_j^i\}$ and $\{ID_{NG}, M_k^i, V_k^i, CT_{NG}\}$ messages. In these messages, ID_{NG} and CT_{NG} are public parameters, V_j^i and V_k^i are two hash outputs, and M_j^i and M_k^i are two encrypted values using the shared key of SM_i and NG. Therefore, because of the one-way property of hash function, $\mathscr{A}$ cannot achieve D_j^i and CM_k^i from V_j^i and V_k^i. Moreover, having access to M_j^i or M_k^i, he/she cannot extract or recover the consumption reports D_j^i and control messages CM_k^i without knowing the K_i^{SM} or K_i^{NG}. The K_i^{SM} and K_i^{NG} are also kept secure using the secret keys of SM_i and NG. Hence, the proposed scheme provides confidentiality and is secure against the message analysis attack. In addition, since the EID_i is the encrypted value of identifier, $\mathscr{A}$ cannot identify the identity of SMs without knowing the private key of NG.

4.2 Impersonation, Modification, and Injection Attacks Resistance

In the proposed scheme, when the SM_i wants to send its usage report D_j^i, it first computes the $h(D_j^i \oplus T_j \oplus ID_i)$, and then sends $\{EID_i, M_j^i, V_j^i\}$ to NG. If the adversary $\mathscr{A}$, either external ε, internal $\jmath$, or global φ, tries to impersonate SM_i and send a forgery message by altering M_j^i, he/she will not be able to compute the proper V_j^i. Therefore, when NG checks the equality of $h(D_j^i \oplus T_j \oplus ID_i) = V_j^i$, it can detect any tampering. By checking this equation, NG becomes certain that the received message is from the real intended SM_i and nobody has modified the D_j^i. Same strategy is done when the NG sends a control message to SM_i. As soon as SM_i checks the equivalence of $h(ID_{NG} \| CM_k^i \| CT_{NG}) = V_k^i$, it not only ensures that the message has not been altered during the transfer, but also becomes sure that the message has

Table 2: Features comparison

Scheme	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}
[20]	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	Yes	No	No	No	Yes
[21]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No	Yes	No	No
[10]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No	Yes	No	No
[28]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	No	No	Yes	No	Yes
[19]	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	No	Yes
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes

F_1. Providing confidentiality F_2. Near real-time authentication F_3. Injection attack resistance F_4. Analysis attack resistance F_5. Modification attack resistance F_6. Replay attack resistance F_7. DoS attack detection capability F_8. Providing anonymity F_9. Considering *SM* memory modification attack resistance F_{10}. Formal verification/proof F_{11}. Extensive hardware implementation on different suitable testbeds F_{12}. Low communication overhead F_{13}. Low computational overhead F_{14}. Being extremely lightweight F_{15}. Presenting key agreement details F_{16}. Presenting the details of shared key storage and retrieval F_{17}. Presenting details of usage reports transmission.

been sent from the authorized *NG*. With the same reason, the scheme is secure against the message injection/pollution attack.

4.3 Replay Attack Resistance

By the employment of T_j in the computation of V_j^i, the messages from *SM*s to *NG* will be kept secure against the replay attack. Further, for the messages from *NG* to *SM*s, when a message is received by *SM*, it first generates a fresh timestamp, then, compares its generated timestamp with the received one. If the elapsed time is shorter than predefined maximum transmission delay, it will accept the message as a non-repeated one. Hence, the proposed scheme can properly withstand the replay attack.

4.4 DoS Attack Resistance

Since in the proposed scheme any tampering on (a) exchanging messages, (b) *SM* flash memory, and (c) *NG* database can be detected very soon, $\mathscr{A}$ cannot perform *DoS* attack. For the exchanging messages or *SM* flash memory, the tampering is revealed when equations $h(D_j^i \oplus T_j \oplus ID_i) = V_j^i$ and $h(ID_{NG} \| CM_k^i \| CT_{NG}) = V_k^i$ are checked. Additionally, as will be stated in the next part, if the internal adversary $\jmath$, who has access to the *NG* database, changes even one field of a record, *NG* will be informed very soon.

4.5 Insider Attack Resistance

In our proposed scheme, an insider adversary $\jmath$ cannot access the confidential data nor can perform the *DoS* attack. Since in the proposed scheme, the confidential data are saved as encrypted, the insider attacker $\jmath$ is not a privileged attacker. As a result, he/she cannot perform any special attack by having access to the *NG* database. Since in the *NG* database only the encrypted form of K_i is saved, $\jmath$ cannot get access to shared keys and if he/she tries to alter a field of a record, the tampering will be detected as soon as the equivalence of $h(K_i^{NG} \| ID_i) = H_i^{NG}$ is checked.

4.6 Memory Modification Attack Resistance

In the proposed scheme, since K_i is obfuscated using the Exclusive-OR operation, it cannot be meaningfully modified without having the secret key of the *SM*. Therefore, this scheme can withstand the *SM* memory modification attack. As stated in part 4.4, at the worst case scenario, the adversary φ who has gained access to the *SM* memory, cannot even perform *DoS* attack.

5 Automatic Formal Verification

In order to ensure that none of the usage reports or control messages can be accessed by an adversary and the impersonation or replay attack cannot take place, we have used a well-known and popular automatic protocol verifier called ProVerif [8]. Figure 3 shows the obtained output from this tool.

The first two results are the results of two injective correspondence that assures SM_i has really executed the protocol with *NG* and vice versa and also the received messages by each of these two entities are fresh. Therefore, these two results prove the resistance of the protocol against impersonation and replay attacks. In the ProVerif, proving the reachability properties is among the most basic capabili-

```
Output: root@ubuntu:~/Proverif/proverif1.94pl1# ./proverif SMandNG.pv
-- Query inj-event(endNG) ==> inj-event(startNG)
Completing...
Starting query inj-event(endNG) ==> inj-event(startNG)
RESULT inj-event(endNG) ==> inj-event(startNG) is true.
-- Query inj-event(endSMi) ==> inj-event(startSMi)
Completing...
Starting query inj-event(endSMi) ==> inj-event(startSMi)
RESULT inj-event(endSMi) ==> inj-event(startSMi) is true.
-- Query not attacker(Ki[])
Completing...
Starting query not attacker(Ki[])
RESULT not attacker(Ki[]) is true.
-- Query not attacker(CMik[])
Completing...
Starting query not attacker(CMik[])
RESULT not attacker(CMik[]) is true.
-- Query not attacker(Dij[])
Completing...
Starting query not attacker(Dij[])
RESULT not attacker(Dij[]) is true
```

Figure 3: The results of analysing the proposed protocol using Proverif

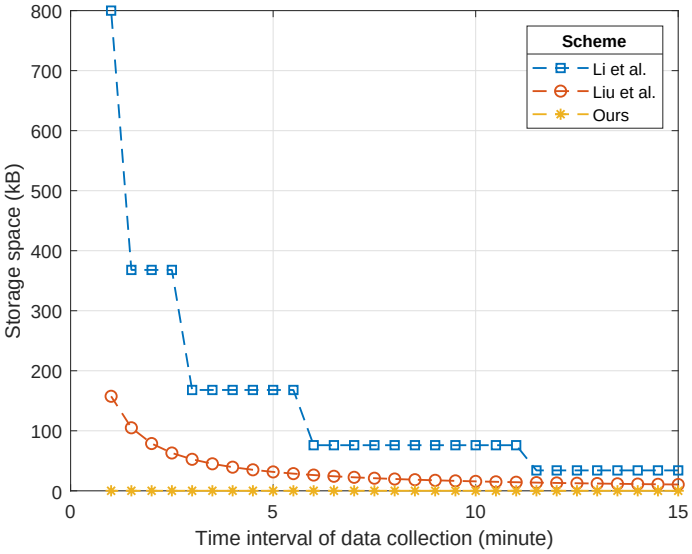

Figure 4: Storage space comparison for different time intervals

ties that lets us to check whether a term can be accessed by an attacker or not. The last three results are the results of such queries that indicate the attacker cannot obtain K_i, CM_k^i, and D_j^i. Therefore, the achieved results prove the secrecy of shared keys, control messages, and usage reports.

6 Efficiency Analysis and Experimental Study

In this section, we compare our proposed anonymous lightweight communication protocol with the related ones. Our comparative analysis shows better performance in terms of

a. Storage;
b. Communication;
c. Computational costs.

In the following, we present the detailed discussion.

6.1 Storage Space

Since only Liu *et al.* [20] and Li *et al.* [19] have discussed the storage space, in this section, we compare the proposed scheme with these two.

In Liu *et al.*'s scheme [20], *SM* needs to store r_j, R_j, and C_j, where $j = 1, 2, \ldots, 96$. The required storage space for r_j is 128×96 bits, the needed storage space for R_j is 256×96 bits, and the required storage space for C_j is 512×96 bits. Hence, the total required storage space is 10.5 kB.

In Li *et al.*'s scheme [19], the *SM* needs to store r_j, C_j, and API_j, where $j = 1, 2, \ldots, 128$. Considering the length of each random number to be 128 bits, the storage space of r_j is 128×128 bits, the storage space of $C_j = Enc_{k_i}(r_j \| TS_j)$ is $2 \times 128 \times 128$ bits, and the storage space of API_j, where each API_j contains seven hash values, is $7 \times 256 \times 128$ bits. Thus, the total required storage space is 34 kB.

In comparison to the previous schemes, in our scheme, the *SM* only needs to store E_i^{SM} and EID_i. Therefore, the required storage space is significantly reduced to only $256 + 256 = 512$ bits. Table 3 shows the comparison. Further, the required storage space for different time intervals of data collection, from 1 to 15 minutes, is illustrated in Figure 4.

Table 3: Storage space

[20]	[19]	Proposed
10.5 kB	34 kB	512 bits

6.2 Communication Cost

For Liu *et al.*'s scheme [20], the communication cost includes the encrypted r_{NG} which is 512 bits, 96 encrypted coefficients of the "f" function that are 128×96 bits, the reports $\{ID_i \| C_j \| S_j\}$ that are $(128 \times 96) + (512 \times 96) + (256 \times 96)$ bits, and the control messages $\{ID_{NG} \| M_k^1 \| M_k^2\}$ that are $(128 \times 4) + (256 \times 4) + (256 \times 4)$ bits. Therefore, the total communication cost is 12.375 kB.

For Li *et al.*'s scheme [19], the communication cost includes the encrypted root node value which takes 256 bits and $\{U_i \| C_j \| S_j \| API_j\}$ reports which

Table 4: Daily Communication Cost

[20]	[21]	[10]	[28]	[19]	Proposed
12.37 kB	7.81 kB	7.81 kB	8.12 kB	27.03 kB	7.81 kB

are sent from *SM* to *NG*. As a result, the total communication cost is $(256) + (128 \times 96) + (2 \times 128 \times 96) + (128 \times 96) + (7 \times 256 \times 96)$ bits = 27.03125 kB.

For Uludag *et al.*'s scheme [28], the communication cost includes $\{ID_j, SKE(K_{MD_j}^{DC_i}, T \| PRODATA \| HASH(DK, PRODATA))\}$ and $\{ID_{DC_i}, SKE(GK_i, SIGN(PO, COMD) \| COMD)\}$ messages. Therefore, the total communication cost is $(128 \times 96) + (128 \times 96) + (128 \times 96) + (256 \times 96) + (128 \times 4) + (1024 \times 4) + (128 \times 4)$ bits = 8.125 kB.

For Mahmood *et al.*'s [21] and Fouda *et al.*'s [10] schemes, the communication cost includes $\{ID_i, E_{K_i}(M_i \| T_i \| HMAC_{K_i})\}$ and $\{ID_i, E_{K_{ij}}(M_i \| t_{if} \| HMAC_{K_{ij}}(M_i))\}$ messages (and same command messages), respectively. As a result, the total communication cost of both is $(128 \times 96) + (128 \times 96) + (128 \times 96) + (256 \times 96) + (128 \times 4) + (128 \times 4) + (128 \times 4) + (256 \times 4)$ bits = 7.8125 kB.

The communication cost of our scheme includes the reports $\{EID_i, M_j^i, V_j^i\}$ that are $(256 \times 96) + (128 \times 96) + (256 \times 96)$ bits and the control messages $\{ID_{NG}, M_k^i, V_k^i, CT_{NG}\}$ which are $(128 \times 4) + (128 \times 4) + (256 \times 4) + (128 \times 4)$ bits. Thus, the total communication cost of our scheme is 7.8125 kB. Note that if M_k^i contains the updated encrypted identifier, its size will be increased to 256 bits. Table 4 shows the communication cost comparison. In addition, the communication cost for different time intervals of data collection is illustrated in Figure 5.

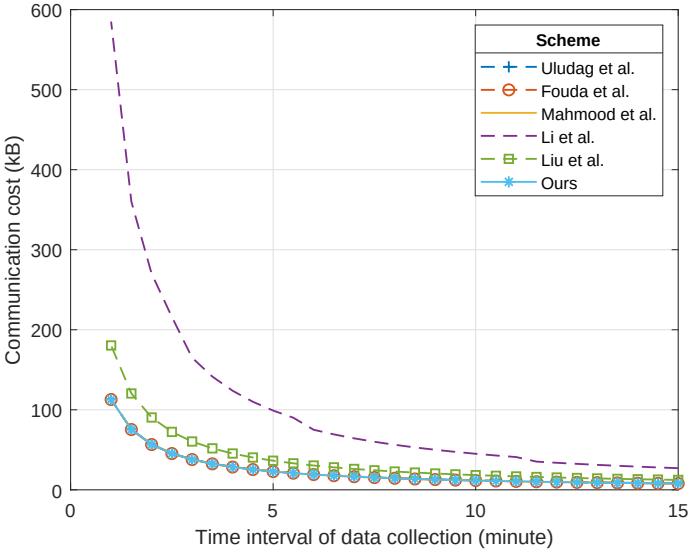

Figure 5: Communication cost comparison for different time intervals

6.3 Computational Cost

In this paper, in order to obtain the *SM* side computational cost of our scheme and the related ones, we have implemented different cryptographic operations on two testbeds. First is an *AVR* microcontroller called ATmega2560 which has 256 kB flash memory, 8 kB SRAM, 4 kB EEPROM, and clock speed of 16 MHz. Second is an *ARM* Cortex-M3 microcontroller called AT91SAM3X8E that has 512 kB flash memory, 96 kB SRAM, and clock speed of 84 MHz. The results have been achieved by utilization of the cryptographic library of ArduinoLibs [1]. On *AVR*, The *RSA* signature verification takes 670 ms; for the *AES*-256 encryption/decryption, 228.96 μs are spent for setting the key, 46.88 μs for encrypting each byte, and 90.05 μs for decrypting each byte; and for the hash operation, 43.89 μs are spent per byte. On *ARM*, The *RSA* signature verification takes 34 ms; for the *AES*-256 encryption/decryption, 46.97 μs are spent for setting the key, 8.04 μs for encrypting each byte, and 14.73 μs for decrypting each byte; and for the hash operation, 1.2 μs are spent per byte. Table 5 shows the execution time of cryptographic operations and Table 6 indicates the comparative computational cost. In Table 6, T_h, T_H, T_{Enc}, T_{Dec}, T_{Rnd}, T_{Pol}, and T_{Ver} are the execution time of one-way hash operation, *HMAC* operation, symmetric encryption, symmetric decryption, random generation, polynomial generation, and *RSA* signature verification, respectively. Moreover, Figures 6 and 7 depict the computational cost comparison for different time intervals of data collection, from 1 to 15 minutes, on *AVR* and *ARM*, respectively.

7 Conclusion

Recently, a number of lightweight communication schemes have been proposed to be employed in the context of smart grid. Nevertheless, most of them are not anonymous and some cannot resist the well-known attacks like the pollution attack. Therefore, in this paper, to remedy the existing challenges, we have proposed an efficient anonymous communication protocol that can properly withstand the common attacks. Moreover, we have implemented the

Table 5: Execution Time Cryptographic Operations on AVR and ARM

Operation	ATmega2560	AT91SAM3X8E
AES-256 ECB Setting the Key	228.96 μs	46.97 μs
AES-256 ECB Encryption (16 Bytes)	750.08 μs	128.64 μs
AES-256 ECB Decryption (16 Bytes)	1440.8 μs	235.68 μs
SHA256 (16 Bytes)	702.24 μs	19.2 μs
HMAC KEY Setup	2836 μs	81 μs
Polynomial Generation	160 ms	10 ms
Random Generation	12 ms	80 μs
RSA Signature Verification	670 ms	34 ms

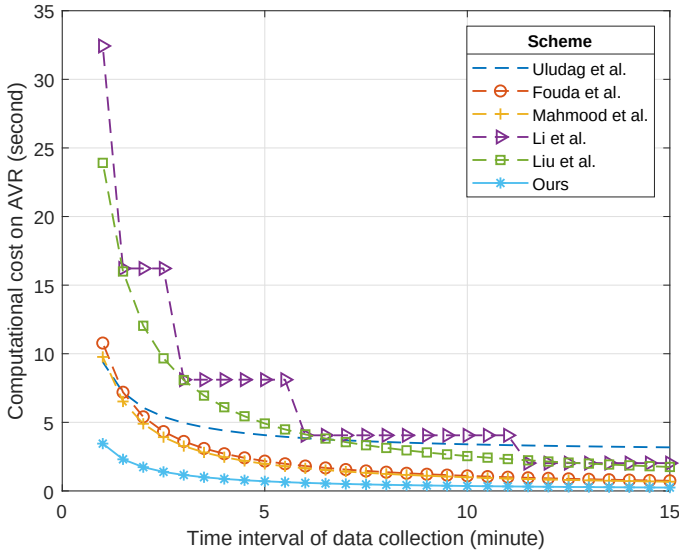

Figure 6: Computational cost comparison for different time intervals on AVR

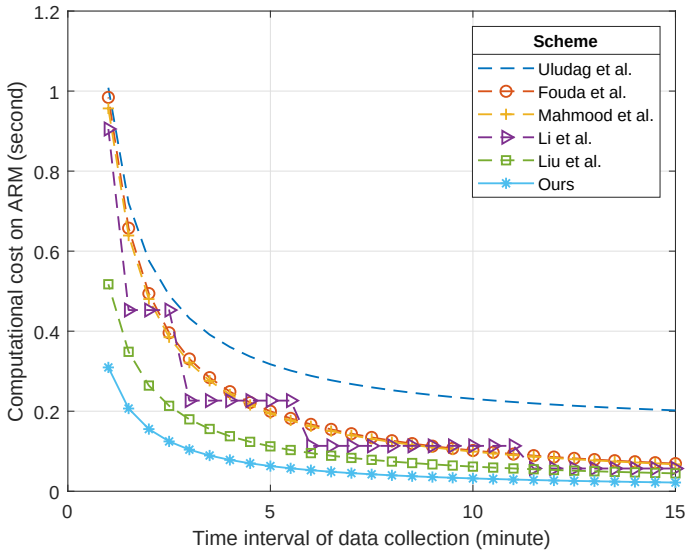

Figure 7: Computational cost comparison for different time intervals on ARM

cryptographic operations on both *AVR* and *ARM* and have compared our scheme with the related ones based on the obtained results on these two hardware. The achieved results indicate the superiority of the proposed scheme in terms of storage, communication, and computational costs. We hope that the presented results of this paper be useful for future researches in this field.

References

[1] "ArduinoLibs: Cryptographic library," 2018. (`http://rweather.github.io/arduinolibs/crypto.html`).

[2] D. Abbasinezhad-Mood, M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.

[3] D. Abbasinezhad-Mood, M. Nikooghadam, "Design and extensive hardware performance analysis of an efficient pairwise key generation scheme for smart grid," *International Journal of Communication Systems*, 2018. (`https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.3507`)

[4] D. Abbasinezhad-Mood, M. Nikooghadam, "An ultra-lightweight and secure scheme for communications of smart meters and neighborhood gateways by utilization of an ARM Cortex-M microcontroller," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6194-6205, 2017.

[5] D. Abbasinezhad-Mood, M. and Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended chebyshev chaotic maps," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4815-4828, 2018.

[6] D. Abbasinezhad-Mood, and M. Nikooghadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Future Generation Computer Systems*, vol. 84, pp. 47–57, 2018.

Table 6: Daily computational cost

Scheme	[20]	[21]	[10]	[28]	[19]	Proposed
Operations	$96\,T_h$ $96\,T_{Enc}$ $1\,T_{Dec}$ $96\,T_{Rnd}$ $1\,T_{Pol}$	$100\,T_H$ $96\,T_{Enc}$ $4\,T_{Dec}$	$100\,T_H$ $96\,T_{Enc}$ $4\,T_{Dec}$	$96\,T_h$ $96\,T_{Enc}$ $4\,T_{Dec}$ $4\,T_{Ver}$	$255\,T_h$ $129\,T_{Enc}$ $128\,T_{Rnd}$	$200\,T_h$ $96\,T_{Enc}$ $4\,T_{Dec}$
Execution Time on ARM	$\approx$ 45 ms	$\approx$ 67 ms	$\approx$ 69 ms	$\approx$ 202 ms	$\approx$ 57 ms	$\approx$ 22 ms
Execution Time on AVR	$\approx$ 1.75 s	$\approx$ 673 ms	$\approx$ 740 ms	$\approx$ 3.18 s	$\approx$ 2.03 s	$\approx$ 246 ms

[7] D. Abbasinezhad-Mood, and M. Nikooghadam, "Efficient design and extensive hardware evaluation of an anonymous data aggregation scheme for smart grid," *Security and Privacy*, 2018. (`https://onlinelibrary.wiley.com/doi/full/10.1002/spy2.24`)

[8] B. Blanchet, V. Cheval, X. Allamigeon, and B. Smyth, *ProVerif: Cryptographic Protocol Verifier in the Formal Model*, 2010. (`http://prosecco.gforge.inria.fr/personal/bblanche/proverif`)

[9] K. Chatterjee, and L. Priya, "HKDS: A hierarchical key distribution scheme for wireless ad hoc network," *International Journal of Network Security*, vol. 20, no. 2, pp. 243–255, 2018.

[10] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.

[11] M. S. Hwang and C. C. Lee, "Research issues and challenges for multiple digital signatures," *International Journal Network Security*, vol. 1, no. 1, pp. 1–7, 2005.

[12] M. S. Hwang and T. Y. Chang, T. Yi, "Threshold signatures: Current status and key issues," *International Journal Network Security*, vol. 1, no. 3, pp. 123–137, 2005.

[13] M. S. Hwang and C. Y. Liu, "Authenticated encryption schemes: Current status and key issues," *International Journal Network Security*, vol. 1, no. 2, pp. 61–73, 2005.

[14] M. S. Hwang, C. C. Lee, and W. P. Yang, "An improvement of mobile users authentication in the integration enviromnents," *AEU-International Journal of Electronics and Communications*, vol. 56, no. 5, pp. 293–297, 2002.

[15] Y. Kabalci, "A survey on smart metering and smart grid communication," *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 302–318, 2016.

[16] A. V. N. Krishna, A. H. Narayana, and K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[17] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[18] C. Li, H. Cheung, C. Yang, "Secure and efficient authentication protocol for power system computer networks," *International Journal of Network Security*, vol. 20, no. 2, pp. 337–344, 2018.

[19] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient Merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655–663, 2014.

[20] Y. Liu, C. Cheng, T. Gu, T. Jiang, X. Li, "A lightweight authenticated communication scheme for smart grid," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 836–842, 2016.

[21] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Computers & Electrical Engineering*, vol. 52, pp. 114–124, 2016.

[22] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.

[23] N. B. M. Mohan, A. S. N. Chakravarthy, and C. Ravindranath, "Cryptanalysis of design and analysis of a provably secure multi-server authentication scheme," *International Journal of Network Security*, vol.20, no. 2, pp. 217–224, 2018.

[24] N. T. Nguyen, H. D. Le, and C. C. Chang. "Provably secure and efficient three-factor authenticated key agreement scheme with untraceability," *International Journal of Network Security*, vol. 18, no. 2, pp. 335–344, 2016.

[25] R. Paspula, K. Chiranjeevi, and S. L. Kumar, "Hidden data transmission with variable DNA technology," *International Journal of Electronics and Information Engineering*, vol. 7, no. 2, pp. 96–106, 2017.

[26] S. K. Ravva, "Common private exponent attack on multi prime RSA," *International Journal of Electronics and Information Engineering*, vol. 7, no. 2, pp. 79–87, 2017.

[27] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status

and key issues," *International Journal Network Security*, vol. 3, no. 2, pp. 101–115, 2006.

[28] S. Uludag, K. S. Lui, W. Ren, and K. Nahrstedt, "Secure and scalable data collection with time minimization in the smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 43–54, 2016.

[29] M. Wazid, A. K. Das, N. Kumar, and J. J. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3144–3153, 2017.

Biography

Dariush Abbasinezhad-Mood received the B.Sc. degree in Computer Science from Payame Noor University, Mashhad, Iran, in 2009 and the M.Sc. degree in Secure Communications from Imam Reza International University, Mashhad, Iran, in 2016 with the first rank. Further, his M.Sc. thesis was elected as the top thesis. His research interests include Cryptography, Trust and Reputation based Systems, Authentication Protocols, Smart Grid Security, Wireless Sensor Networks, Internet of Things, and Embedded Systems. He is a reviewer for several well-known journals, such as IEEE Transactions on Smart Grid, IEEE Transactions on Industrial Informatics, IEEE Systems Journal, IEEE Internet of Things Journal, and IEEE Communications Letters.

Arezou Ostad-Sharif received the B.Sc. degree in Information Technology from Safahan University, Esfahan, Iran, in 2015 and M.Sc. degree in Information Security from Imam Reza International University, Mashhad, Iran, in 2017. Her M.Sc. thesis was elected as the top thesis in 2018. She hopes to advance her education in network security. Her research interest focuses on the security protocols for Wireless Sensor Networks, Internet of Things, Smart Grid, and Telecare Medical Information Systems.

Morteza Nikooghadam received the B.Sc. degree from university of Sadjad, Iran, in 2006, M.Sc. from the Shahid Beheshti University, Iran, in 2008, and Ph.D. from Shahid Beheshti University, Iran, in 2012. He is currently an assistant professor in the Department of Computer Engineering and Information Technology at Imam Reza International University, Mashhad, Iran. His research focuses on Data Security, Cryptography, and Sensor Network Security. His current research interests are Reconfigurable Architectures for multipliers under Galois Field $GF(2^m)$.