Cryptanalysis and Improvement of a Mutual Authentication Scheme for Smart Grid Communications

Piyush Sharma, Garima Thakur, and Pankaj Kumar

1 Introduction

An advanced intelligent network that maximizes energy efficiency is the smart grid. A smart grid is an enhanced electrical power grid that aids in controlling power distribution and facilitating communication between customers and service providers. A bidirectional communication link is the main means by which suppliers and consumers can dynamically change the distribution of power in real-time. By doing this, it is feasible to reliably and effectively transmit electricity while preventing the development of excess electricity. The latter will help in upsurging the power operators profit. As a result, the smart grid mitigates the loopholes of our traditional power grid by utilizing bidirectional communication instead of one-way communication. Bidirectional connectivity, self-control, remote verifier, distributed management, and additional consumer options are just a few of the impressive features of the smart grid. Industry and academic researchers have both shown interest in it.

To deliver proficient, reliable, cost-effective, and sustainable power, the smart grid incorporates controls, automation of the framework, computerization, and new technologies. However, communication between legal organizations is vulnerable to cyberattacks since there are no reliable security measures in place. The complex nature of SG and its several security requirements pose challenges to its widespread use. Two imperative issues are the preservation of user privacy and sender authentication. A suitable authentication mechanism should be implemented to ensure that the sender's identity can be verified because the data transmitted by individual appliances impacts the measure of electricity a generator must produce. Keeping in view the aforementioned security prerequisites, we have proposed an authentication scheme in this paper.

P. Sharma · G. Thakur · P. Kumar (✉)
Srinivasa Ramanujan Department of Mathematics, Central University of Himachal Pradesh, Dharamshala, H.P. 176215, India

J. K. Mandal et al. (eds.), *Proceedings of International Conference on Network Security and Blockchain Technology*, Lecture Notes in Networks and Systems 738,
https://doi.org/10.1007/978-981-99-4433-0_3

1.1 Related Work

Ma et al. [1] presented work on smart grid communication in which they discussed the various challenges and opportunities in smart grid communication. Further, in Yan et al. [2] paper, we studied how smart grid communication helps balance power supply and demand. This paper also allows us to know why we need new infrastructure today, the shortcomings in our old infrastructure, and how we can overcome them through smart grid communication. Kabalci et al. [3] also presented a survey paper that examined the technology, applications, and difficulties associated with smart metering and smart grid communication methods. Faheem et al. [4] provide an overview of several smart grid applications, including their advantages, traits, and prerequisites. This study researches and examines several wired and wireless communication technologies, as well as a number of significant difficulties, unresolved problems, and potential future research topics. Chen et al. [5] presented a paper that describes smart attacks and their defenses in a communication network for the smart grid. Thakur et al. [6] highlighted that the scheme proposed by Son et al. is vulnerable to various security assaults including impersonation attacks, offline password-guessing attacks, etc., and proposed an enhanced authentication scheme that can withstands all security attacks. Maitra et al. [7] also proposed an enhanced authentication scheme employing Chebyshev Chaotic Map and demonstrated the robustness of their scheme utilizing the widely accepted random oracle model and AVISPA simulation tool. Numerous other authentication schemes for smart grid environment have been proposed by Ferrag et al. [8], Fouda et al. [9], Liu et al. [10], and Mahmood et al. [11]. Further, an authentication protocol for home and building area networks was put forth by Li et al. [12]; however, the protocol has substantially higher computational cost. Sule et al. [13] also proposed a variable length message authentication code scheme to ensure a safe interaction among AMI devices and collector nodes but could not provide user anonymity, session key agreement, and message authentication as mentioned by [14].

1.2 Motivation and Contribution

Technology today binds us to convenience and ease of living. We formerly lit our homes with fire, but today, we talk of replacing our outdated grid system with the smart grid. In doing so, there are numerous threats, namely man-in-the-middle attacks, password-guessing attacks, replay attacks, insider attacks, smart card loss attacks, impersonation attacks, etc., and security relevance, namely data confidentiality, non-traceability, message authentication, user anonymity, etc. As stated in the preceding section, despite the introduction of numerous authentication schemes [9, 11–14], none of them are entirely proficient of delivering the needed security attributes for the reference of a smart grid. Therefore, this paper discusses the design flaws and cryptanalysis of Khan et al.'s protocol [14]. We discovered that their scheme

needs an authenticated key agreement, and also, we have mentioned how this protocol is susceptible to offline password-guessing attacks, user impersonation attacks, and replay attacks. Finally, we proposed an enhanced scheme utilizing the fuzzy extractor function [15, 16] that has no security risks.

2 Review of Khan et al.'s Scheme

This section reviews the Khan et al. [14] scheme. All the symbols used in this paper are given in Table 1.

2.1 Initialization Phase

1. S_g chooses q, E_q (a, b): $y^2 = x^3 + ax + b \bmod q$, where $a, b \epsilon G$ with $4a^3 + 27b^2 \bmod q \neq 0$.
2. S_g selects a base point $P \epsilon G$ and chooses their $h(\cdot)$.
3. S_g selects its private key as $s \epsilon Z_q^*$ and public key as $\text{PKs} = s \cdot P$.
4. $\{\text{Eq}(a, b), q, p, P, \text{PKs}, h(\cdot)\}$ are public parameters, and s is kept confidential.

Table 1 Symbols and their description

Symbol	Description
ECC	Elliptic curve cryptography
Q	Large prime number
G	Additive group
P	Generator of G
U	User
s_g	Smart grid server
ID_u	Identity of U
PW_u	Password of U
B_u	Biometric of U
Z_q^*	Multiplicative group of order $q - 1$
SK_{us}	Session key between u and s
Gen(.), Rep(.)	Fuzzy extractor, reproduction function
A	Adversary
δt	Time stamp
$h(\cdot)$	Hash function
$\|, \oplus$	Concatenation, bitwise XOR operators

2.2 Registration Phase

1. The user U chooses an identity ID_u and a password PW_u, imprints their B_u, and computes $(\sigma_u, \theta_u) = \mathrm{Gen}\ (B_u)$. A random nonce a is generated by U, thereafter computes $B_1 = h\ (\mathrm{PW}_u \parallel \sigma_u) \oplus \mathrm{a}$ and transfers the registration request $\{ID_u, B_1, t_{BG1}\}$ to S_g through a secure channel.
2. On receiving registration request from U, S_g verifies $t_{BG2} - t_{BG1} \leq \delta t$. Thereafter, computes $B_2 = \mathrm{h}\ (\mathrm{ID}_u \parallel s \parallel z)$ where s is the private key of S_g and z denotes the counter. Then, S_g computes $B_3 = B_2 \oplus B_1$ and stores $\{B_3, z, p, h(\cdot)\}$ in the database and forwards $\{B_3, z, p, h(\cdot)\}$ to U.
3. After receiving $\{B_3, z, p, h(\cdot)\}$, U computes $B_4 = B_3 \oplus \sigma_u$, $B_5 = h\ (\mathrm{ID}_u \parallel \mathrm{PW}_u \parallel B_4)$ and stores $\{B_3, B_4, B_5\}$ in the database.

2.3 Login and Authentication Phase

The subsequent steps are performed by U and S_g to accomplish mutual authentication:

1. Firstly, user U enters his/her $\mathrm{ID}'_u, \mathrm{PW}'_u, imprints\ B'_u$, computes $\sigma'_u = \mathrm{Rep}(B'_u, \theta'_u)$, $B'_4 = B'_3 \oplus \sigma'_u$, $B'_5 = h\big(\mathrm{ID}'_u \parallel \mathrm{PW}'_u \parallel B'_4\big)$ and verifies $B'^{?}_{5=} B_5$. If the validation holds, then U generates a random nonce $r \epsilon Z^*_q$, thereafter computes $M_1 = h\ (\mathrm{ID}_u \parallel B_1 \parallel t_1)$, $\mathrm{ID}_{U1} = \mathrm{ID}_u \oplus (B_1 \oplus t_1)$ and sends $\{M_1, \mathrm{ID}_{U1}, r \cdot p, t_1\}$ to S_g through a public channel.
2. On receiving $\{M_1, \mathrm{ID}_{U1}, r \cdot p, t_1\}$ from user, S_g verifies $t_2 - t_1 \leq \delta t$. Thereafter, S_g computes $\mathrm{ID}^*_u = \mathrm{ID}_{U1} \oplus (B_1 \oplus t_1)$, $M^*_1 = \mathrm{h}\ (\mathrm{ID}^*_u \parallel B_1 \parallel t_1)$ and verifies $M^{*?}_{1=} M_1$. If the validation holds, then $\mathrm{S_g}$ generates a random nonce $b \epsilon Z^*_q$, computes $M_2 = h\ (\mathrm{ID_s} \parallel B_3 \parallel t_2)$ and session key as $\mathrm{SK_{su}} = h(\mathrm{ID}^*_u \parallel \mathrm{ID}_s \parallel M^*_1 \parallel M_2 \parallel B_3 \cdot p \parallel r \cdot b \cdot p \parallel s \cdot p \parallel t_3)$, $\mathrm{ID_{S1}} = \mathrm{ID_s} \oplus (B_3 \oplus t_3)$. Finally, the message $\{M_2, \mathrm{ID}_{S1}, b \cdot p, t_3\}$ is sent to the user U.
3. On receiving data from S_g, U verifies $t_4 - t_3 \leq \delta t$. If yes then, the user computes $\mathrm{ID}^*_s = \mathrm{ID}_{S1} \oplus (B_3 \oplus t_3)$, $M^*_2 = h\big(\mathrm{ID}^*_s \parallel B_3 \parallel t_3\big)$ and verifies $M^{*?}_{2=} M_2$. If the validation holds, then the session key is computed by $\mathrm{SK}_{us} = h\ (\mathrm{ID}_u \parallel \mathrm{ID}^*_s \parallel M_1 \parallel M^*_2 \parallel B_3 \cdot p \parallel r \cdot b \cdot p \parallel \mathrm{PK}_s \parallel t_3)$.

3 Cryptanalysis of the Khan et al.'s Scheme

This segment shows the cryptanalysis of Khan et al.'s scheme [14].

3.1 Offline Password-Guessing Attack

Step 1. Suppose adversary A is a privileged—insider of the server S_g. Then, he/she can access the information of the registration phase, i.e., adversary A has the information $\{ID_u, B_1, t_{BG1}\}$, where $B_1 = h(PW_u \| \sigma_u) \oplus a$.

Step 2. If adversary A steals user's device, the database of U can be accessed. Therefore, the information $\{B_3, B_4, B_5\}$ is known to A.

Step 3. Now, the adversary guesses a password PW_u^*, computes $B_5^* = h(ID_u \| PW_u^* \| B_4)$, and verifies $B_5^* \stackrel{?}{=} B_5$. If B_5^* equals B_5, then the adversary successfully guessed the user's password.

3.2 User Impersonation Attack

A can produce a new forged login message in this attack, which is then sent to S_g. If S_g acknowledges this message, the attacker will be successful in user impersonation attack. In Khan et al.'s scheme, this attack is possible if the privileged user performs the task of A. The data $\{ID_u, B_1, t_{BG1}\}$ is accessible to the privileged user. This attack then takes place as follows:

Step 1. A generates its own random nonce a ϵZ_q^*, thereafter A computes $M_1 = h(ID_u \| B_1 \| t_1)$, $ID_{U1} = ID_u \oplus (B_1 \oplus t_1)$, where t_1 is the current time stamp. Then, A sends $\{M_1, ID_{U1}, a \cdot p, t_1\}$ to S_g through a public channel.

Step 2. On receiving $\{M_1, ID_{U1}, a \cdot p, t_1\}$ from user, S_g verifies $t_2 - t_1 \leq \delta t$. After verification S_g computes $ID_u^* = ID_{U1} \oplus (B_1 \oplus t_1)$, $M_1^* = h(ID_u^* \| B_1 \| t_1)$ and verifies $M_1^* \stackrel{?}{=} M_1$. This verification would be successful because of using correct identification factors. Thus, A is successful in performing a user impersonation attack.

3.3 Replay Attack

The message $\{M_1, ID_{U1}, r \cdot p, t_1\}$ transferred over the public channel is captured by A. Assume A is a privileged—insider of the server S_g. As previously mentioned, A can compose its message $\{M_1, ID_{U1}, r \cdot p, t_1\}$ and transmit it to S_g again. As a result, the attacker's replay attack is successful.

4 Design Flaws of Khan et al.'s Scheme

Khan et al. [14] have the following potential design problems:

- During the login and authentication phase of [14], in step-2 when the S_g generates a random nonce $b \epsilon Z_q^*$ and computes $M_2 = h\ (\mathrm{ID}_s \parallel B_3 \parallel t_2)$, the server used its identity ID_s, B_3 (stored in the database in the registration phase), and used a timestamp t_2. After that S_g compute session key as $\mathrm{SK}_{su} = \mathrm{h}\ (\mathrm{ID}_u{}^* \parallel \mathrm{ID}_S \parallel M_1{}^* \parallel M_2 \parallel B_3 \cdot p \parallel r \cdot b \cdot p \parallel s \cdot p \parallel t_3)$. But in step-3 of login and authentication phase, the user first compute $\mathrm{ID}_S{}^* = \mathrm{ID}_{S1} \oplus (B_3 \oplus t_3)$, then computes $M_2{}^* = h\ (\mathrm{ID}_s{}^* \parallel B_3 \parallel t_3)$ and verifies $M_2{}^* =^? M_2$. Here, U uses time stamp t_3 instead of t_2. So, by property of the hash function, $M_2{}^*$ does not equal M_2. Hence, mutual authentication does not hold. Also, U computes session key as $\mathrm{SK}_{us} = h\ (\mathrm{ID}_u \parallel \mathrm{ID}_s{}^* \parallel M_1 \parallel M_2{}^* \parallel B_3 \cdot p \parallel r \cdot b \cdot p \parallel \mathrm{PK}_S \parallel t_3)$.

Since $M_2{}^* \neq M_2$, session keys SK_{su} and SK_{us} are not equal. Therefore, there is no mutual authentication and session key agreement in [14].

- For a moment, if we assume S_g computes M_2 as $M_2 = h\ (\mathrm{ID}_s \parallel B_3 \parallel t_3)$, that is, S_g uses time stamp t_3 in M_2 so that $M_2{}^* = M_2$. But still, there is no session key agreement because the S_g computes the session key as $\mathrm{SK}_{su} = h\ (ID_u{}^* \parallel \mathrm{ID}_s \parallel M_1{}^* \parallel M_2 \parallel B_3 \cdot p \parallel r \cdot b \cdot p \parallel s \cdot p \parallel t_3)$ whereas U computes session key as $\mathrm{SK}_{us} = h\ (\mathrm{ID}_u \parallel \mathrm{ID}_s{}^* \parallel M_1 \parallel M_2{}^* \parallel B_3 \cdot p \parallel r \cdot b \cdot p \parallel \mathrm{PK}_S \parallel t_3)$. Then, U uses the server's public key PK_S instead of $s \cdot p$, but PK_S is computed as $\mathrm{PK}_S = s \cdot P$, where P is a base point belonging to elliptic curve group G, and p is the generator of G. So, if p and P are different, then PK_S does not equal $s \cdot p$. As a result, the hash function's attribute prevents the server-generated session key SK_{su} from being similar to the user-generated session key SK_{us}. Because of this, there is no session key agreement.
- The recommended scheme must include a password change phase if the user wants to change their password for whatever reason. However, the common authentication approach suggested in [14] does not include a step for changing passwords. The user must periodically change his password for security reasons. They will thus be protected from numerous attacks.

5 Discussion and Improvements

This section proposes an enhanced scheme that overcomes the security threats of Khan et al.'s scheme [14]. The proposed scheme has the following four phases:

5.1 Initialization Phase

In this phase, server S_g chooses q, E_q (a, b): $y^2 = x^3 + ax + b \bmod q$, where $a, b \epsilon Z_q$ with $4a^3 + 27b^2 \bmod q \neq 0$ and chooses his/her $h(\cdot)$. S_g generates its private key s ϵ Z_q^* and computes public key as $PK_S = s \cdot p$, where p is the generator of G. {Eq (a, b), q, p, PK_s, $h(\cdot)$} are public parameters and server keeps its private key s secretly.

5.2 Registration Phase

The following are the steps for user registration:

Step 1. The U selects an identity ID_u and a password PW_u, and he/she imprints his/her B_u and computes $(\sigma_u, \theta_u) = \text{Gen}(B_u)$. Then, a random nonce a is generated by user U and computes $B_1 = h(PW_u \parallel \sigma_u) \oplus a$, $HID_u = h(ID_u \parallel \sigma_u) \oplus a$ and transfers data $\{HID_u, B_1, \}$ toward S_g through a secure channel.

Step 2. On receiving data from U, S_g computes $B_2 = h(HID_u \parallel s \parallel z)$ where s is the private key of S_g and z is the counter. Then, S_g computes $B_3 = B_2 \oplus B_1$. Server S_g stores $\{B_3, z, p, h(\cdot)\}$ in the database and forwards it to U.

Step 3. After receiving $\{B_3, z, p, h(\cdot)\}$, U computes $B_4 = B_3 \oplus \sigma_u$, $B_5 = h(ID_u \parallel PW_u \parallel B_4) \oplus a$ and stores $\{B_3, B_4, B_5\}$ in database of U.

5.3 Login and Authentication Phase

The subsequent steps are performed by U and S_g to accomplish mutual authentication:

Step 1. Firstly, U enters their ID'_u, PW'_u and imprints B'_u then computes $\sigma'_u = \text{Rep}(B'_u, \theta'_u)$, $B'_4 = B_3 \oplus \sigma'_u$, $a' = HID_u \oplus h(ID'_u \parallel \sigma'_u)$, $B'_5 = h(ID'_u \parallel PW'_u \parallel B'_4) \oplus a'$ and checks $B'_{5=}^{?} B_5$. If the verification holds, then U computes $M_1 = h(ID_u \parallel B_1 \parallel a \cdot p)$ and encrypts the message $M_{U1} = E_{K_U}(M_1, ID_u, t_1)$ with the help of key $K_U = h(HID_u \| a \cdot s \cdot p)$. Finally, U sends $\{M_{U1}, a \cdot p, t_1\}$ to S_g through a public channel.

Step 2. On receiving $\{M_{U1}, a \cdot p, t_1\}$ from U, S_g verifies $t_2 - t_1 \leq \delta t$. After verification S_g decrypts $(M_1, ID_u, t_1) = D_{K_S}(M_{U1})$ with the help of key $K_S = h(HID_u \| a \cdot s \cdot p)$, computes $M_1^* = h(ID_u \| B_1 \| a \cdot p)$ and verifies $M_{1=}^{*?} M_1$. Thereafter, S_g generates a random number $b \epsilon Z_q^*$, computes $M_2 = h(ID_s \parallel B_3 \| b \cdot p)$, calculates the session key as $SK_{su} = h(ID_u \| ID_s \| M_1^* \| M_2 \| B_3 \cdot p \| a \cdot b \cdot p \| s \cdot p \| t_3)$, and encrypts the message $M_{S1} = E_{K_{S1}}(M_2, ID_S, t_3)$with the help of key $K_{S1} = h(HID_u \| a \cdot b \cdot p)$. Then, the message $\{M_{S1}, b \cdot p, t_3\}$ t is sent to U.

Step 3. On receiving message from S_g, the user verifies $t_4 - t_3 \leq \delta t$, if yes then, user decrypts $(M_2, \mathrm{ID}_s, t_3) = D_{K_{U1}}(M_{S1})$with the help of key $K_{U1} = h(\mathrm{HID}_u \| a \cdot b \cdot p)$ and computes $M_2^* = h(\mathrm{ID}_s \| B_3 \| t_3)$, verifies $M_{2=}^{*?} M_2$. If the verification holds, the user computes the session key as $\mathrm{SK}_{us} = h(\mathrm{ID}_u \| \mathrm{ID}_s \| M_1 \| M_2^* \| B_3 \cdot p \| a \cdot b \cdot p \| \mathrm{PK}_s \| t_3)$.

5.4 Change Password and Biometric Phase

If U wants to change their biometric and password, then the following steps are to be followed by U:

Step 1. U inputs $\mathrm{ID}'_u, \mathrm{PW}'_u, B'_u$ and computes $\sigma'_u = \mathrm{Rep}(B'_u, \theta'_u)$, $B'_4 = B_3 \oplus \sigma'_u$, $a' = \mathrm{HID}_u \oplus h\left(\mathrm{ID}'_u \| \sigma'_u\right)$, $B'_5 = h\left(\mathrm{ID}'_u \| \mathrm{PW}'_u \| B'_4\right) \oplus a'$ and verifies $B'^{?}_{5=} B_5$. If verification does not hold, then the session is terminated. Otherwise, U selects a new biometric B_u^* and password PW_u^*. The user computes $\left(\sigma_u^{\mathrm{new}}, \theta_u^{\mathrm{new}}\right) = \mathrm{Gen}(B_u^*)$ and $B_1^{\mathrm{new}} = h\left(\mathrm{PW}_u^* \| \sigma_u^{\mathrm{new}}\right) \oplus$ a. User sends $\left\{\mathrm{HID}_u, B_1^{\mathrm{new}}, t_{BG1}\right\}$ toward S_g.

Step 2. Firstly, S_g verifies $t_{\mathrm{BG1}} - t_{\mathrm{BG2}} \leq \delta t$ then computes $B_3^{\mathrm{new}} = B_2 \oplus B_1^{\mathrm{new}}$. Thereafter, S_g replaces $B_3\, with\, B_3^{\mathrm{new}}$ in the database and sends B_3^{new} to U.

Step 3. After receiving B_3^{new}, the user U computes $B_4^{\mathrm{new}} = B_3^{\mathrm{new}} \oplus \sigma_u^{\mathrm{new}}$, $B_5^{\mathrm{new}} = h(\mathrm{ID}_u \| \mathrm{PW}_u^* \| B_4^{\mathrm{new}}) \oplus a$. Further, U replaces PW_u by PW_u^*, B_u by B_u^*, σ_u by σ_u^{new} and $\theta_u\, by\, \theta_u^{\mathrm{new}}$. Finally, user U stores $\left\{B_1^{\mathrm{new}}, B_3^{\mathrm{new}}, B_4^{\mathrm{new}}, B_5^{\mathrm{new}}\right\}$ in database replacing $\{B_1, B_3, B_4, B_5\}$, respectively.

6 Informal Security Analysis

In this section, an informal security analysis of the improved scheme has been discussed.

6.1 Replay Attack

The most frequent defenses against this attack are the timestamp and random numbers. The user and server generate random integers (a and b) and a time stamp condition $t_i - t_j \leq \delta t$ at each stage of the proposed scheme. They are used to ensure the message's freshness. By checking the timestamp of received messages, the user and the server can determine the nature of the assault. As a result, the proposed scheme maintains a replay attack.

6.2 Man-in-the-Middle Attack

Any attacker A may attempt to log in to the server using the previous message. A replay $\{M_{U1}, a \cdot p, t_1\}$, where $M_1 = h(\mathrm{ID}_u \| B_1 \| a \cdot p)$, $K_U = h(\mathrm{HID}_u \| a \cdot s \cdot p) and\ a \epsilon Z_q^*$ and t_1 is the time stamp that prohibits the replay attack. S_g checks the two verifying conditions $M_{1=}^{*?} M_1$ and $t_2 - t_1 \leq \delta t$ after receiving the message. Similarly, when the user receives the message $\{M_{S1}, b \cdot p, t_3\}$, the user verifies $t_4 - t_3 \leq \delta t$, and $M_{2=}^{*?} M_2$. A cannot access the user's or server's private keys and therefore cannot determine a real verifier. As a result, A cannot modify a parameter since the verifiers need to be suitably modified. In light of this cryptography attack, the proposed framework is secure.

6.3 Mutual Authentication

Here is a description of message authentication:

- S_g confirms the time stamp conditions $t_2 - t_1 \leq \delta t$ after receiving the message $\{M_{U1}, a \cdot p, t_1\}$. Then, verifies $M_{1=}^{*?} M_1$.
- U confirms the time stamp conditions $t_4 - t_3 \leq \delta t$ after receiving the message $\{M_{S1}, b \cdot p, t_3\}$. Then, verifies $M_{2=}^{*?} M_2$.

Message security is ensured by checking parameters, and hash values are difficult for an attacker to guess. Therefore, the recommended framework allows for mutual authentication.

6.4 Impersonation Attack

A can get $\{M_{U1}, a \cdot p, t_1\}$ and try to compute M_{U1}, it is very difficult to compute for any attacker because M_{U1} is encrypted with symmetric key $K_U = h(\mathrm{HID}_u \| a \cdot s \cdot p)$. Similarly, M_{S1} is encrypted with key $K_{S1} = h(\mathrm{HID}_u \| a \cdot b \cdot p)$ and protected with secret parameters using the elliptic curve computational Diffie Hellman problem. As a result, A cannot impersonate anyone in correspondence between these forwarded messages. Therefore, our protocol is protected from user impersonation assault.

6.5 Key Freshness

Every step in the suggested scheme uses a new key, such as a random number or a time stamp, so the key freshness criterion holds true throughout each session.

6.6 User Anonymity

The suggested protocol is free from the problem of anonymity, since the hash function and biometrics protect user identity, and random number a, as $\mathrm{HID}_u = h(\mathrm{ID}_u \| \sigma_u) \oplus a$.

6.7 Offline Password-Guessing Attack

During the registration phase, the user selects their password PW_u, which is then used to compute $B_1 = h(\mathrm{PW}_u \| \sigma_u) \oplus a$, which is secured by a secure hash value and protected by private parameters like a biometric and a random value. Also, assume that A gets B_5 from the user database, but they can't guess the password from here because the random value a is used in B_5. In a secure medium, it isn't easy to guess the user's password in this way. As a result, the proposed scheme defends against the offline password-guessing attack.

6.8 Session Key Agreement

According to the proposed scheme, the session keys for the user and server are computed as $\mathrm{SK_{su}} = h(\mathrm{ID}_u \| \mathrm{ID}_s \| M_1^* \| M_2 \| B_3 \cdot p \| a \cdot b \cdot p \| s \cdot p \| t_3)$ and $\mathrm{SK}_{us} = h(\mathrm{ID}_u \| \mathrm{ID}_s \| M_1 \| M_2^* \| B_3 \cdot p \| a \cdot b \cdot p \| \mathrm{PK}_s \| t_3)$. Clearly, $\mathrm{SK}_{su} = \mathrm{SK}_{us}$. As a result, a session key agreement exists.

6.9 Data Confidentiality

- The user encrypts the message $M_{U1} = E_{K_U}(M_1, \mathrm{ID}_u, t_1)$ with the help of key $K_U = h(\mathrm{HID}_u \| a \cdot s \cdot p)$. Thereafter, S_g decrypts $(M_1, \mathrm{ID}_u, t_1) = D_{K_S}(M_{U1})$ with the help of key $K_S = h(\mathrm{HID}_u \| a \cdot s \cdot p)$ and verifies $M_{1=}^{*?} M_1$.
- $\mathrm{S_g}$ encrypts the message $M_{S1} = E_{K_{S1}}(M_2, \mathrm{ID}_S, t_3)$ with the help of key $K_{S1} = h(\mathrm{HID}_u \| a \cdot b \cdot p)$. Thereafter, U decrypts $(M_2, \mathrm{ID}_s, t_3) = D_{K_{U1}}(M_{S1})$ with the help of key $K_{U1} = h(\mathrm{HID}_u \| a \cdot b \cdot p)$ and verifies $M_{2=}^{*?} M_2$. Thus, the proposed scheme secures data privacy.

Table 2 Security features

Security features	Khan et al. [14]	Fouda et al. [9]	Li et al. [12]	Sule et al. [13]	Proposed
MM	Yes	Yes	Yes	Yes	Yes
RP	No	Yes	Yes	Yes	Yes
UA	No	No	Yes	No	Yes
KF	Yes	Yes	Yes	Yes	Yes
MA	No	No	Yes	No	Yes
IM	No	Yes	Yes	Yes	Yes
SK	No	Yes	Yes	No	Yes

Yes: Prevent the attack
No: Does not prevent the attack

7 Performance Analysis

7.1 Security Features

In this section, we compare the security features of proposed scheme with related previous schemes such as Khan et al. [14], Fouda et al. [9], Li et al. [12], and Sule et al. [13]. Table 2 shows that the proposed scheme resists all malicious attacks, namely man-in-the-middle attack (MM), key freshness (KF), replay attack (RP), message authentication (MA), session key agreement (SK), user anonymity (UA), etc. As a result, compared to the other existing schemes, the proposed scheme offers a wider range of security features.

7.2 Computational Cost

This section compares the computational cost of various authentication protocols [9, 12–14] for the login and authentication phase. Relying on [14], the execution time for point addition (PA), point multiplication (PM), symmetric encryption/decryption (ESED), modular exponentiation (ME), public key encryption/decryption (PKED), hash-based message authentication (HMAC), and hash operation (HO) is 0.0288, 2.226, 0.0046, 3.85, 3.85, 0.0046, and 0.0023 ms. Table 3 demonstrates that the computational cost of our scheme is higher than [14]; however, the cost is lesser than that of [9, 12, 13]. Therefore, the proposed protocol offers higher security and efficiency.

Table 3 Comparison of the computational costs

Scheme	Operations	Computational cost (ms)
Khan et al. [14]	$4T_{PM} + 7T_{HO}$	$\cong$ 8.9201
Fouda et al. [9]	$4T_{ME} + 4T_{PKED} + 2T_{HO}$	$\cong$ 30.8046
Li et al. [12]	$7T_{ME} + 6T_{HO}$	$\cong$ 26.9638
Sule et al. [13]	$4T_{ME} + 4T_{PKED} + 2T_{HMAC}$	$\cong$ 30.8092
Proposed	$11T_{PM} + 12T_{HO}$	$\cong$ 24.5136

8 Conclusion

Smart grid technology is gaining popularity and is becoming a new area of interest. Sensitive data stored in the smart grid has upsurged the requirement of security of bidirectional communication. In this study, we have examined the various design flaws and vulnerability of scheme suggested by [14] in opposition of numerous cryptographic attacks like user impersonation attacks, replay attacks, and offline password-guessing attacks. We have also proposed an enhanced authentication framework for smart grid environment. The informal security analysis of the proposed scheme shows the efficiency and security against the various attacks. Further, the proposed scheme is compared to the related protocols in terms of computational efficiency and security features. The results demonstrate the elevated security of the proposed protocol. Therefore, the proposed work is suitable for smart grid environment.

References

1. Ma R, Chen HH, Huang YR, Meng W (2013) Smart grid communication: its challenges and opportunities. IEEE Trans Smart Grid 4(1):36–46
2. Yan Y, Qian Y, Sharif H, Tipper D (2012) A survey on smart grid communication infrastructures: motivations, requirements and challenges. IEEE Commun Surv Tutorials 15(1):5–20
3. Kabalci Y (2016) A survey on smart metering and smart grid communication. Renew Sustain Energy Rev 57:302–318
4. Faheem M, Shah SBH, Butt RA, Raza B, Anwar M, Ashraf MW, Gungor VC (2018) Smart grid communication and information technologies in the perspective of Industry 4.0: opportunities and challenges. Comput Sci Rev 30:1–30
5. Chen PY, Cheng SM, Chen KC (2012) Smart attacks in smart grid communication networks. IEEE Commun Mag 50(8):24–29
6. Thakur G, Kumar P, Jangirala S, Das AK, Park Y (2023) An effective privacy-preserving blockchain-assisted security protocol for cloud-based digital twin environment. IEEE Access 11:26877–26892
7. Maitra T, Singh S, Saurabh R, Giri D (2021) Analysis and enhancement of secure three-factor user authentication using Chebyshev Chaotic Map. J Inform Sec Appl 61:102915
8. Ferrag MA, Maglaras LA, Janicke H, Jiang J, Shu L (2018) A systematic review of data protection and privacy preservation schemes for smart grid communications. Sustain Cities Soc 38:8–835
9. Fouda MM, Fadlullah ZM, Kato N, Lu R, Shen XS (2011) A lightweight message authentication scheme for smart grid communications. IEEE Trans Smart grid 2(4):675–685

10. Liu Y, Cheng C, Gu T, Jiang T, Li X (2015) A lightweight authenticated communication scheme for smart grid. IEEE Sens J 16(3):836–842
11. Mahmood K, Chaudhry SA, Naqvi H, Shon T, Ahmad HF (2016) A lightweight message authentication scheme for smart grid communications in power sector. Comput Electr Eng 52:114–124
12. Li X, Wu F, Kumari S, Xu L, Sangaiah AK, Choo KKR (2019) A provably secure and anonymous message authentication scheme for smart grids. J Parallel Distrib Comput 132:242–249
13. Sule R, Katti RS, Kavasseri RG (2012) A variable length fast message authentication code for secure communication in smart grids. In: 2012 IEEE power and energy society general meeting. IEEE, pp 1–6
14. Khan AA, Kumar V, Ahmad M (2022) An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach. J King Saud Univ Comput Inform Sci 34(3):698–705
15. Maurya AK, Das AK, Jamal SS, Giri D (2021) Secure user authentication mechanism for IoT-enabled Wireless Sensor Networks based on multiple Bloom filters. J Syst Architect 120:102296
16. Dodis Y, Reyzin L, Smith A (2004) Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Advances in cryptology-EUROCRYPT 2004: international conference on the theory and applications of cryptographic techniques, Interlaken, Switzerland, May 2–6, 2004. Proceedings 23. Springer Berlin Heidelberg, pp 523–540