Energy Efficient Authentication Scheme for Industrial Smart Grid Environments

IV. SECURITY ANALYSIS

In this section, we perform the mathematical analysis called informal security analysis to explain the security strength of the proposed scheme against various security attacks is protected under the concept of ECDHP, CDHP, and a hash function (h1(), h2()). Since, in the entire paper, we focused on a lightweight motive to get the benefit of performance and remove existing weaknesses. Therefore, in the next subsection, we learn how a proposed scheme is preventive against security attacks that affect the performance of SG devices. Then, we reverify our informal security analysis claim using widely accepted security analysis tools (i.e.., AVISPA, and ProVerif tool (formal security analysis tool)). Initially, we summarized the essential security requirement needed for designing a secure authentication scheme.

Then in the later subsection, we discuss the security advantage of the proposed scheme is as follows:

A. Essential security requirements

Many communications between nodes happens via an unreliable public channel. The result will mimic or modify communication over public channel thought intruder. Therefore, it is foremost important to study various security attacks and other security requirements that the proposed scheme must fulfill as follows [**]:

1. Replay attacks: In this attack, to get access to sensitive data, delay, replay and repeat the valid transmission of data between node and server or node, attacker keep on sending replay the previously captured information.

2. DoS attacks: An intruder sends out a huge number of authentication requests to deplete server resources, thereby crippling a server to accept and validate a fresh request for authentication.

3. Password guessing attacks: An intruder attempts any conceivable variation of passwords to figure out the right password. Thus, it is advised to use a high entropy password to minimize the risk of such a brute force attempt.

4. Node impersonation attacks: The intruder impersonates a legal device in this attack by introducing itself to legal authorities device, who's taking service from that organization in order to obtain any important, confidential data or information.

5. Server impersonation attacks: The intruder impersonates a legal entity in this attack by introducing itself to legal authorities in that organization in order to obtain any important, confidential data or information.

6. Server database stolen attacks: Throughout this attack, intruders represented themselves as a legal entity against legal authorities in that organization and steal server databases contain essential, confidential data or information.

7. Node spoofing attacks: An attacker will execute Domain Name System Attack (DNS) in this attack, where the attacker tries to exploit the DNS knowledge that is contained in the DNS resolver cache.

8. Many logged-in users attacks: In this attack, more than one user, including the attackers, may sign into the same account as the user service. Throughout this scenario, the legal client is abused by a number of users or attackers

9. Forward secrecy attacks: In this attack, the attacker impersonates a trustworthy entity or organization for some significant, confidential data to be accessed.

10. Insider attack: An authorized user operating inside a server company conducts a malicious attack on a server device or network for personal advantage.

11. Known session-specific temporary information attack: If the secrete credential (i.e., ephemeral secrets) of devices or trusted server TS leak out somehow, the adversary using those secretes can find out session key.

12. Attack on node anonymity: In this attack, an attacker can track users who exchange information through the public channel of communication.

13. Stolen verifier attack: During this attack, the server storage for device credential is compromised or stolen by an intruder. Then intruders used the same credentials to verify the false devices toward Sever.

14. A man-in-middle attack: In this attack, attackers interact and possibly change contact between two trustworthy parties. Those two-partisans mistakenly believe they are engaging with each other.

15. Brute force attack: Throughout this attack, the attacker tries to access the device credential by using automated software to validate the identity using millions of passwords generated by it.

In addition to the above, to make its design more secure, the proposed scheme must hold the following security criteria as follow:

1. Confidentiality: All messages shared between trustworthy parties keep it secret or confidential through encryption using the session key. Only with a legal device with whom another node verifies authentication, may decode it.

2. Mutual authentication: The two trusted parties mutually authenticate each other through mutual authentication or bidirectional authentication. It means the scheme can defend to spoofing attacks.

3. Key freshness: Each time a new key has produced, the freshness of the key would be retained. It means that the device is secure from attacks on transient details specific to the established session.

B. Informal security analysis

In this subsection, we mathematically verify the benefits of ECDHP, CDHP, and a hash function (h1(), h2()) against security attack, which in inherited in the proposed scheme. Then we

compare the security benefits of the proposed scheme with recent existing schemes. As mentioned, we are interested in those attack, especially replay attack and DoS attacks that can affect their performance of the SG devices as follows:

1. Replay attacks: In this attack, an attacker may impersonate legitimated node (U_i) by reusing previously send a challenge $< I_i, X_i, Y_i, P_{ke_j} >$ and send it to another node (U_j). Node U_j recomputes and checks: (i.e., $Y_i.G =? X_i \oplus (P_x \oplus P_{ks}h2(I_i, P_{ke_i}))h1(I_j.X_i.Psw_j)$). If the deduction is true, then it computes another challenge $< X_j, Y_j, P_{xj} >$ and forward it to U_i. However, after impersonating challenge $< X_j, Y_j, P_{xj} >$, an attacker unable to verify the challenge equation (i.e., $Y_j.G =? X_j \oplus (P_{xj} \oplus P_{ks}h2\left(I_j, P_{ke_j}\right))h1(I_i.X_j.Psw_i)$) and compute a session key $\left(S_K = x_{i1}.X_j.Psw_i.P_{ke_j}\right)$. This is due to the attacker unaware of secrete private information Psw_i of node U_i. In addition, security parameters $<Psw_i$, Psw_j, $x_{j1}$and $x_{i1} >$ are nither sent through any messages over public channel nor can be acquired from the U_i and U_j. In addition, the security of Psw_i (or Psw_j) and x_{i1} (or x_{j1}) is protected inside S_K, $P_{ke_i} = Psw_i.G$ (or $P_{ke_j} = Psw_j.G$) and $X_i = x_{i1}.G$ (or $X_j = x_{j1}.G$) by the concept of ECDHP and CDHP. Therefore, attacker unable to compute a valid session key (S_K). Hence, the proposed scheme is preventive against a replay attack.

2. DoS attacks: The proposed scheme terminates the communication if the number of incorrect attempts of entering I_i' exceed a limit value. However, the authentication request continues to proceed as long as correct I_i provided. Even if the adversary replace challenge $< I_i, X_i, Y_i, P_{ke_i} >$ with $< I_i', X_i', Y_i', P_{kei}' >$ by randomly choosing ephemeral secrete x_{i1}' and P_{yi}' and send it to the node U_j; However, node U_j unable to match the deduction (i.e., $Y_i'.\mathrm{G}?\,(\mathrm{Received}) = Y_i'.\mathrm{G}\,(\mathrm{computed})$) and terminates the protocol with a message of failure. This is because of parameters that set for I_i', are altered, that make deduction become unmatched with what was received with what was we computed. Therefore, the proposed scheme preventive against a DoS attack.

SG devises resources performance (i.e., optimization, and management) is greatly affected by those attacks (i.e., replay attack and DOS) [22]. However, the proposed scheme is resistant to both these attacks. The proposed scheme thus assures the efficiency reliability and performance of the node's communication in SG. In addition, we have listed a few security attacks which can affect the performance of the proposed scheme as follows:

3.Password-guessing attacks: In the proposed scheme, the original password (Psw_i) wrapped in the form of $I_i = H(ID_i||Psw_i)$ and ($P_{kei} = Psw_i.G$)). This password is stored in I_i using one-way collision resist function, and it stored in P_{ke_i} using the concept of ECDHP and CDHP, which is impossible to break. In addition, neither I_i and P_{ke_i} sent thought any open public channel. Therefore, attacker unable to guess psw_i without knowing completely I_i and P_{ke_i}. Hence, the proposed scheme is preventive against a password-guessing attack.

4. Node Impersonation attacks: Consider the scenario, where the phase of authentication of node U_i, impersonate by an adversary node. Adversary node impersonate parameters $< I_i, X_i, Y_i, P_{ke_i} >$ of U_i by realized U_i that an adversary node is an authenticated node in the SG network. The adversary node randomly choose parameters $< X_j', Y_j', P_{xj}' >$ responded to the node U_i. Then, node U_i compute deduction (i.e., $Y_j.G =? X_j \oplus (P_{xj} \oplus P_{ks}h2\left(I_j, P_{ke_j}\right))h1(I_i, X_j.Psw_i)$). However, deduction comes wrong, and the session gets terminated at the same moment. This is because of the reason that attacker randomly compute $< X_j, Y_j, P_{xj} >$ as $< X_j', Y_j', P_{xj}' >$ and is entirely ignorant of parameters $<x_{j1}$ (i.e., ephemeral parameter), P_{kej} , P_{xj} , $P_{yj}>$ which used to compute actual $< X_j, Y_j, P_{xj} >$. Since node U_j is a tamper-proof device, therefore, the security of $< P_{yj}, P_{xj} >$ in protected inside U_j. Even, attacker if tried to get those parameters from U_j by breaking its security, U_j automatically erase those parameters from Table and turn to reset state, which indicates that U_j tried to harm by attackers. If an attacker succeeds somehow, then it is impossible for an adversary to compute a session key. Since session key need secrete parameters $< Psw_j, x_{j1} >$ which is completely unknown to the adversary. In addition, the security of $< Psw_j, x_{j1} >$ protected inside $P_{ke_j} = Psw_j.G$, $X_j = x_{j1}.G$ and past session key ($S_K = x_{j1}.X_i.Psw_j.P_{kei}$) by the concept of ECDHP and CDHP. Therefore, the proposed scheme helps registered nodes (U_i, U_j) to preventive against node impersonation attacks.

5. Server Impersonation attacks: Consider the scenario, where the phase of authentication of node U_i, impersonate by an adversary server. Adversary server impersonates parameters $< I_i, X_i, Y_i, P_{ke_i} >$ of U_i by realized U_i that an adversary server is an authenticated server in the SG network. The adversary server randomly choose parameters $< X_j', Y_j' >$ responded to the node U_i. Then, node U_i compute deduction (i.e., $Y_i.G =? X_j \oplus P_{yi}\, h1(I_i, X_j.Psw_i)$). However, deduction comes wrong, and the session gets terminated at the same moment. This is because of the reason that the attacker server randomly compute $< X_j, Y_j >$ as $< X_j', Y_j' >$ and is entirely ignorant of parameters $<x_{j1}$ (i.e., ephemeral parameter), P_{ke_i} , $P_{yi}>$ which used to compute actual $< X_j, Y_j >$. Since node U_i is a tamper-proof device, therefore, the security of $< P_{yi} >$ in protected inside U_i. Even, attacker if tried to get those parameters from U_i by breaking its security, U_i automatically erase those parameters from there storage Table and turn to reset state, which indicates that U_i tried to harm by attackers. If an attacker succeeds somehow, then it is impossible for an adversary server to compute a session key. Since session key need secrete parameters $< s, x_{j1} >$ which is completely unknown to the adversary. In addition, the security of $< s, x_{j1} >$ protected inside $P_{ks} = s.G$, $X_j = x_{j1}.G$ and past session key ($S_K = x_{j1}.X_i.s.P_{kei}$) by the concept of ECDHP and CDHP. In addition, trusted server TS, the security of s is protected inside encryption key $E_{ks} = s.I_{TS}.G$, whose security is protected by the concept of ECDLP and CDHP. Therefore, without knowing secretes $< s, I_{TS} >$ adversary server unable to compute $< P_{xi}, P_{yi} >$. Hence, the proposed scheme is preventive against server impersonation attacks.

6. Sever database stolen attacks: In this case, if an attacker does stolen attack on a server database; however, an attacker is futile in breaching the server database. This is because of that server database security is protected by encryption key ($E_{ks} = s.I_{TS}.G$). An attacker unable to compute encryption key, without knowing s and I_{TS}. It is impossible for an attacker to extract s and I_{TS} from $E_{ks} = s.I_{TS}.G$ in polynomial time, whose security is protected by the concept of ECDLP and CDHP. If any of secrete parameter (i.e., either s or I_{TS}) is revealed to an attacker, however its impossible to know second secrete parameter from $E_{ks} = s.I_{TS}.G$. In addition, even if the encryption key is compromised by somehow, then attacker unable to determine $< P_{xi}, P_{yi} >$ from $<P_{xi}', P_{yi}'>$. This is due to the parameter $< P_{xi}, P_{yi} >$ security still protected by s and

I_{TS}. Therefore, the proposed scheme is preventive against server database stolen attacks.

7. Node-spoofing attacks: In this attack, an attacker may masquerade as a legitimated node U_j, to known secrete credential of node U_j. However, as mentioned in the password guessing attack and replay attack. It is impossible for an attacker to extract and guess password $< Psw_i \quad Psw_j >$ from $< P_{kei} = Psw_i.G$, $P_{kej} = Psw_j.G, >$ due to the concept of ECDLP and CDHP. Also, security parameters, $< x_{j1}$ and $x_{i1} >$ which is completely ephemeral and expires after every particular session. In additions, security parameters $< P_{xi}, \ P_{yi} >$ are completely stored in tamper-proof nodes $< U_j, U_j >$. Therefore, an adversary unable to launch successful node-spoofing attacks because of a lack of information. Hence, the proposed scheme is secure to a node-spoofing attack.

8. Many logged-in users attacks: The proposed scheme maintains a working bit W_{bit-i} for every communication. If many other nodes knew the valid credential of legitimated node U_i, then in this case only one legal node U_i, can communicate with another node U_j at a time. Whenever every communication successfully passes the verification step, both corresponding nodes set their $W_{bit-i\ or\ j}$ for a respective node as one. In that case, if received node U_j receive an authentication request from another node with similar kind of credential of U_i, then U_j check whether the status bit W_{bit-i} for received credential equal to one (i.e., represents the currently active communication of a node U_j with node U_i). If the status is one then node U_i reject those requests. In another term, U_j checks the status of W_{bit-i} for node U_i before establishing the communication. Therefore, the proposed scheme preventive against many logged-in users attack.

9. Forward secrecy: Even by some means, if the private key of both nodes U_j and U_j are compromised; the confidentiality of the session key must not be affected. In the proposed scheme, session key (S_K) security depends not only on the private key Psw_i or Psw_j but also on ephemeral secretes x_{i1} or x_{j1} which expire after every session termination. For an attacker, it computationally impossible to extract those security parameters $<Psw_i$ or Psw_j, x_{i1} or $x_{j1} >$ from session key $(S_K = x_{i1}.Psw_i.X_j.P_{ke_j} = x_{j1}.Psw_j.X_i.P_{ke_i})$ due to the complexity of the ECDLP and CDHP. Therefore, even if the current S_K leaked, it impossible for an attacker to guess the next session key due to the expiration of ephemeral secretes (x_i or x_j), after the termination of every session. Hence, the proposed scheme is secure to a forward secrecy attack.

10. Insider attacks: In the proposed scheme, password (Psw_i) send to the server (TS) or nodes (U_i) in the wrapped format $I_i = H(ID_i||Psw_i)$. Therefore, it is very computationally infeasible for TS, to extract Psw_i from I_i. This is due to the hardness of a one-way collision-resistant hash function *h ()*. In addition, it's impossible to extract password $< Psw_i$ or $Psw_j >$ from $<P_{kei} = Psw_i.G, \ P_{kej} = Psw_j.G, >$ and from session key $(S_K = x_{i1}.Psw_i.X_j.P_{ke_j} = x_{j1}.Psw_j.X_i.P_{ke_i})$ due to the concept of the ECDLP and CDHP. Therefore, privileged insider attacks fail to impersonate the secret information of a legitimate node U_i. Hence, the proposed scheme is preventive against an insider attack.

11. Known session-specific temporary information attacks: The session key $(S_K = x_{i1}.Psw_i.X_j.P_{ke_j} = x_{j1}.Psw_j.X_i.P_{ke_i})$ computation is performed after successful mutual authentication between node U_i and U_j. If adversary somehow successful in discovering ephemeral secretes (x_{i1}, x_{j1}), however, knowledge of a single ephemeral key is not enough to compute a session key. Since it is very hard for an attacker to guess a password, which is protected under the concept of a one-way collision-resistant hash function *h ()* and the ECDLP and CDHP. In addition, if either one of the parameters is revealed to the attacker, however it computationally infeasible to determines either (x_{i1}, x_{j1} or Psw_i or Psw_j) from the past session key (S_k). This is due to the difficulties in solving and comprehend the CDHP and ECDLP for pairs in S_K using a polynomial-time algorithm. Consequently, the proposed scheme withstands a known session-specific temporary information attack.

12. Attacks on node anonymity: Device anonymity ensures that the attacker cannot find the actual device identity (ID_i, ID_j) of node U_i and U_i. This identity is well protected with their password by concealing it inside a one-way collision-resistant hash function (i.e., $I_i = h1(ID_i \ ||Psw_i)$). Both ID_i and Psw_i neither send through any message. In addition, the security of the device is well protected because of the tamper-proof nature [18]. Therefore, the proposed scheme, ensure the safety of the device identity and password over the communication channel.

13. Stolen-verifier attack: If by some other means, attacker discover the information for node U_i and U_j, an attacker can launch a power-analysis attack for finding secrete information stored inside the devices. However, during the registration phase, for node U_i identity (I_i), the trusted server store following parameters: $P'_x = P_x \oplus \text{s} \oplus I_i$, and $P'_y = P_x \oplus P_y \oplus \text{s} \oplus I_i$ in the server database. Even if the attacker attempted and stole those records, the attacker failed to perform malicious activity. This is because of attacker unable to access actual security parameters (P_x, P_y), which is protected using secret server key (s) and I_{TS}. These security parameters (P_x, P_y) stored in an encrypted table of tamper-proof device U_i with encrypted secrete key $E_{ke} = psw_i.G$. In addition, even if the attacker made substantial login request with another node U_j, attacker fails to verify the challenge $(Y_i.G$=?) and compute a session key S_k. This is due to the attacker completely unaware of the psw_i of node U_i. This psw_i hard to extract from parameters $I_i \ and \ P_{ke_i}$, Because its security is protected by the property of the one way collision-resistant function, ECDHP, and device tamper-proof nature. Without knowing psw_i, an attacker can not compute $Y_i.G$ and verify the challenge and compute the session key. Consequently, the proposed scheme can resist stolen-verifier attacks.

14. Man-in-the-middle attacks: The successful mutual authentication, ensure the safety of an authentication scheme from man-in-the-middle attacks. The proposed scheme prevents Man-in-middle attack and supports successful mutual authentication, which we concluded from the result of the ProVerif tool mentioned in Subsection IV.*D*.

15. Brute-force attacks: To perform a successful brute-force attack, the attacker just need to extract security parameters $<I_i, X_i, Y_i,$ and $P_x >$ from an open transmitted channel. Even if the attacker succeeds in obtaining that security parameter. However, an attacker can't determine passwords $(psw_i \ and \ psw_j)$ of the U_i, then U_j which is entirely obscure. In addition, it's very hard to determine and speculate ephemeral secretes $(x_{i1}$ and $x_{j1})$ due to the complexity of ECDHP and ECDLP, which we explained in node impersonation attacks. Therefore, the proposed scheme can brute-force attacks.

Finally, we checked the proposed scheme security strength in comparison with the existing schemes [3, 10-15] by comparing some selected security attributes in Table II. The result of the comparison shows that the proposed scheme is free from security weaknesses within the existing schemes.

TABLE II
SECURITY ATTRIBUTES COMPARISON

Attacking scenarios	Chos et al [10]	Odelu et.al. [11]	Tsai et.al. [12]	Bhanse et.al. [13]	Abbasine Zhad-Mood et.al. [14]	Moghadam et.al. [15]	Kumar et.al. [3]	Proposed
Reply attack	✓	✓	-	✓	✓	✓	-	✓
Password guessing attack	-	✓	✓	-	-	-	✓	✓
Embedded device Impersonation attack	-	✓	-	✓	✓	✓	-	✓
Denial of service attack	-	-	-	-	-	-	-	✓
Many logged-in user attack	-	-	-	-	-	-	-	✓
Server Impersonation attack	-	-	-	-	-	-	-	✓
Forward secrecy	-	✓	✓	✓	✓	✓	✓	✓
Insider attack	-	-	-	✓	-	-	-	✓
Known session - specific temporary information attack	✓	✓	✓	-	✓	-	✓	✓
Attack on user anonymity	-	✓	✓	✓	-	✓	✓	✓
Server database stolen attack	✓	-	-	-	-	-	-	✓
Man-in-middle attack	✓	✓	-	✓	✓	✓	✓	✓
Brute Force attack	✓	✓	-	-	-	-	-	✓
Formal Security Analysis Proof (AVISPA+ProVerif)	-	-	-	-	✓	✓	✓	✓

REFERENCES

[10] Cho, S., Li, H. and Choi, B.J., 2014, November. PALDA: Efficient privacy-preserving authentication for lossless data aggregation in Smart Grids. In 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm) (pp. 914-919). IEEE.

[11] Odelu, V., Das, A.K., Wazid, M. and Conti, M., 2016. Provably secure authenticated key agreement scheme for smart grid. IEEE Transactions on Smart Grid, 9(3), pp.1900-1910.

[12] Tsai, J.L. and Lo, N.W., 2015. Secure anonymous key distribution scheme for smart grid. IEEE transactions on smart grid, 7(2), pp.906-914.

[13] Bhanse, P., Mishra, B. and Jena, D., 2019, October. A Novel Smart Meter Authentication Scheme for Secure Smart Grid Communication. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) (pp. 1275-1279). IEEE.

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

[15] Moghadam, M.F., Nikooghadam, M., Mohajerzadeh, A.H. and Movali, B., 2020. A lightweight key management protocol for secure communication in smart grids. Electric Power Systems Research, 178, p.106024.

[3] Kumar, P., Gurtov, A., Sain, M., Martin, A. and Ha, P.H., 2018. Lightweight authentication and key agreement for smart metering in smart energy networks. IEEE Transactions on Smart Grid, 10(4), pp.4349-4359.

[18] Evanczuk, Stephen. "Employing Tamper Detection and Protection in Smart Meters." DigiKey, Electronic Products, 17 June 2015, www.digikey.com/en/articles/employing-tamper-detection-and-protection-in-smart-meters.

[22] Darwish, M., Ouda, A. and Capretz, L.F., 2017. Formal Analysis of an Authentication Protocol against External Cloud-Based Denial-of-Service (DoS) Attack. arXiv preprint arXiv:1711.09985.