ECC-Based Authenticated Key Agreement Protocol for Industrial Control System

Yanru Chen, *Member, IEEE*, Fengming Yin, Shunfang Hu, *Member, IEEE*, Limin Sun, *Member, IEEE*, Yang Li, Bin Xing, Liangyin Chen, *Member, IEEE*, and Bing Guo

***Abstract*—Nowadays, Industrial Internet of Things (IIoT) technology has made a great progress and the industrial control systems (ICSs) have been used extensively, which has brought more and more serious information security threats to the ICS at the same time. The authenticated key agreement (AKA) protocol is a common method to ensure the communication security. This work proposes a lightweight AKA protocol based on the elliptic curve cryptography (ECC) algorithm to adapt to the resource-constrained environment. We only employ hash operation, XOR operation, and ECC algorithm to encrypt the data in the authentication and key agreement phase, and avoid involving the register center while proceeding the key agreement, to give consideration to both performance and security. The security analyses indicate that our protocol can meet nine critical security requirements, more than all of the existing protocols, and the performance analysis carried out indicates that our protocol has less computational and communication overheads in contrast to other corelative protocols.**

***Index Terms*—Elliptic curve cryptography (ECC), identity authentication, Industrial Internet of Things (IIoT), integrity validation, key agreement.**

I. Introduction

With the rapid development and wide use of the industrial control system (ICS), the threats in the field of information security faced by ICS are becoming much more serious [1], [2]. The authenticated key agreement (AKA) protocol is a common method to ensure information security [3]. To adopt to the situation that Industrial Internet of Things (IIoT) devices are often resource constrained [4], [5], many works choose elliptic curve cryptography (ECC) because of its lower resource consumption. Therefore, in the field of IIoT communication security, the ECC-based AKA protocol has become a research hotspot.

In recent years, prolific related research on ECC-based AKA protocols has been conducted. Tsai and Lo [6] proposed an ECC-based anonymous AKA protocol using bilinear pairing, which can guarantee the secrecy of critical and private data. But because of the utilize of bilinear pairing, the protocol has a high computational overhead. Furthermore, the protocol cannot ensure message integrity. In this occasion, Debiao et al. [7] improved this protocol in 2016, reduced the computational and communication overheads. But the new protocol cannot guarantee anonymity or resist known session-specific temporary information attacks. Han et al. [8] proposed an ECC-based effective AKA protocol using RFID technology, which aims at the IoT system. But this protocol also cannot ensure messages integrity. In 2018, a provably secure AKA protocol based on ECC was proposed by Odelu et al. [9], but it can neither guarantee anonymity and message integrity nor resist known session-specific temporary information attacks. In the same year, Mahmood et al. [10] proposed a lightweight ECC-based AKA protocol with lower computational and communication overheads, but it cannot guarantee anonymity, message integrity, and perfect forward secrecy, and cannot resist known session-specific temporary information attacks either. Also in 2018, Saeed et al. [11] proposed an ECC-based AKA protocol using the ID-PKC authentication scheme. But they made assumptions that the channel that the PKG and WSN nodes use to transfer data was always secure, and the channel used to transfer global secret key would never be attacked, which cannot be satisfied in practice. Before long, an ECC-based security enhanced lightweight AKA protocol was proposed by Abbasinezhad-Mood and Nikoughadam [12], which can neither ensure message integrity and anonymity nor resist replay attacks. Lohachab and Karambir [13] proposed an ECC-based AKA protocol, which employed the message queue telemetry transport technology and the access control mechanism. But this one also cannot ensure perfect forward secrecy. Poomagal and Kumar [14] proposed a lightweight secure information transport protocol based on ECC, which cannot resist known session-specific temporary information attacks either. Singh et al. [15] proposed a secure IoT-based mutual

Manuscript received 23 July 2022; revised 29 September 2022; accepted 18 October 2022. Date of publication 3 November 2022; date of current version 7 March 2023. This work was supported in part by the National Natural Science Foundation Program of China under Grant 62072319 and Grant 62172061; in part by the National Key Research and Development Program of China under Grant 2020YFB1711800 and Grant 2020YFB1707900; in part by the Science and Technology Project of Sichuan Province under Grant 2022YFG0041; and in part by the Science and Technology Innovation Project of Luzhou under Grant 2021CDLZ-11. *(Corresponding authors: Liangyin Chen; Bing Guo.)*

Yanru Chen, Fengming Yin, Shunfang Hu, Liangyin Chen, and Bing Guo are with the College of Computer Science and Institute for Industrial Internet Research, Sichuan University, Chengdu 610065, China (e-mail: yanruchen@stu.scu.edu.cn; yinfengming@stu.scu.edu.cn; hushunfang@stu.scu.edu.cn; chenliangyin@scu.edu.cn; guobing@scu.edu.cn).

Limin Sun is with the Internet of Things Security Laboratory, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: sunlimin@iie.ac.cn).

Yang Li is with the Science and Technology on Security Communication Laboratory, Institute of Southwestern Communication, Chengdu 610041, China (e-mail: yishuihanly@gmail.com).

Bin Xing is with the National Engineering Laboratory for Industrial Big-Data Application Technology, CASICloud-Tech Company Ltd., Beijing 100040, China, and also with the Chongqing Innovation Center, Industrial Big-Data Company Ltd., Chongqing 400707, China (e-mail: xingbin@casic.com).

Digital Object Identifier 10.1109/JIOT.2022.3219233

authentication scheme for healthcare applications in WSNs, which is also based on ECC. The secret key can be shared between users, gateways, and sensor nodes in this scheme. But the scheme also cannot resist known session-specific temporary message attacks. In order to achieve the goal of communication security, an AKA protocol has to meet critical security requirements [16] like anonymity, message integrity, perfect forward security, mutual authentication, and resistance to various attacks, including known session-specific temporary information attack, replay attack, impersonation attack, man-in-the-middle attack, and unknown key-share attack. But none of the protocols mentioned above can both meet critical security requirements and resist all types of attacks. Moreover, the computational and communication overheads of some protocols above are not low enough to be applicable in ICS.

To solve the problems mentioned above, we carried out further researches and improvements, and proposed a new lightweight AKA protocol based on ECC. This protocol only employs hash operation, XOR operation, and point addition and multiplication based on elliptic curve to implement a set of operations, including sending hash value of identities instead of real identities via public channel, regenerating and using a new random number to update the temporary parameters each time before sending a message via a public channel, and using the hash operation instead of encryption and decryption on those parameters whose actual value are not needed, in the authentication and key agreement phase, to ensure the security and reduce the resource consumption at the same time. Besides, the protocol avoids involving the register center in the authentication and key agreement phase to improve the performance further. Moreover, the principals in the protocol can register and sign out dynamically, which improves the scalability of the protocol. The formal and informal security analyses conducted shows that our protocol can meet all the security requirements we mentioned above, which means our protocol has a much better security than existing protocols; and the performance evaluation indicates that the computational and communication overheads of the protocol are both lower than existing ones.

The contributions of the work can be listed into following four points.

1) A new AKA protocol employs some lightweight encryption and validation methods using only ECC, hash operation, and XOR operation in the authentication and key agreement phase is proposed. The protocol not only can meet all critical security requirements and resist all types of attack mentioned above but also has low resource consumption and good performance.
2) The protocol supports dynamic principal registration and signing out to keep a high scalability, while the register center is avoided to improve the performance in the authentication and key agreement phase.
3) Security analyses and simulation based on ProVerif have been carried out to prove that our protocol can meet all the critical security requirements we mentioned before.
4) Computational and communication overheads are evaluated for our protocol to prove that our protocol has better performance than existing ones. The computational and communication overheads have been reduced by 11.9% and 2.7%, respectively.

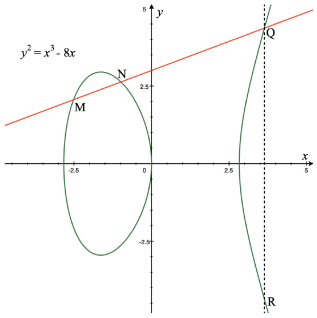

Fig. 1. Geometric addition.

The Organization of This Article: Section II presents some preliminaries applied to this work. Section III shows the detailed process of the proposed protocol. Section IV presents formal security analyses and simulation experiment based on ProVerif for our protocol. Section V presents the security and performance comparison. Section VI gives the conclusion of this work.

II. Preliminaries

A. Elliptic Curve Cryptography

The ECC algorithm is an algorithm based on basic theories of elliptic curves. It has higher proceeding speed and shorter secret key than RSA [17]. For example, a 256-bit secret key in the ECC algorithm is as secure as a 3072-bit secret key in the RSA algorithm [14]. In this section, some conceptions in the ECC algorithm will be brought out.

1) Elliptic Curve Equation: An elliptic curve equation usually looks like $E(a, b) : y^2 = (x^3 + ax + b) \mod p$ where p is a random big prime and $\forall a, b \in Z_p^*$ such that $(4a^3 + 27b^2) \mod p \neq 0$ [10]. A random point G in the Galois Field of the curve $E(a, b)$ will be chosen as the base point later.

2) Geometric Addition: Select two random points M, N on the elliptic curve $E(a, b)$, and draw a line passing through M, N. The line will intersect a third curve Q on the curve $E(a, b)$. Then take the inverse point R of the point Q, and the result of the geometric addition $M + N$ is R, just as shown in Fig. 1. The elliptic curve point multiplication can be transferred to the geometric addition, as $k \cdot P = \underbrace{P + P + \cdots + P}_{k}$, where k is a constant and P is a point on the elliptic curve.

3) Elliptic Curve Discrete Logarithm Problem: If there are two points M and N on the elliptic curve $E(a, b)$ and $M = k \cdot N(\forall k \in Z^*)$, the elliptic curve discrete logarithm problem is to find the value of the coefficient k while the points M and N are given. It is impossible to solve this problem because of its computational complexity.

4) Elliptic Curve Diffie–Hellman Problem: Choose a random point G in Galois Field of the elliptic curve $E(a, b)$. When $\forall a, b \in Z^*, c = a \cdot b$, and $M = a \cdot G, N = b \cdot G, R = c \cdot G$, the elliptic curve Diffie–Hellman problem is to find the point R while points M, N are given. It is impossible to solve the problem effectively in the polynomial time.

B. BAN Logic

BAN logic [18] is a logic of beliefs based on message transmissions and trust relations between protocol principals. It is usually used to analyze the security of protocols formally by many researchers [19].

Several steps need to be taken while using BAN logic: first, model the protocol to be analyzed; then set security goals; finally, try to verify the goals using basic rules. If the goals can be proved, the protocol is reliable; otherwise, there may be some loopholes in the protocol.

There are three basic objects in BAN logic: principals, secret keys, and statements. The notations used in BAN logic analysis are defined as follows.

1) P, Q: Principals.
2) M, N: Statements.
3) K: Secret key.
4) $P \mid\equiv M$: P believes M.
5) $P \triangleleft M$: P sees M.
6) $P \mid\sim M$: P once said M.
7) $P \Rightarrow M$: P has jurisdiction over M (P's belief about M should be trusted).
8) $< M >_N$: M combined with N.
9) $\#(M)$: M is fresh.
10) (M, N): M or N is one part of (M, N).
11) $\{M\}_K$: M is encrypted with the symmetric key K.
12) SSK: Session key used in the current session.
13) $P \overset{K}{\longleftrightarrow} Q$: P and Q may use the shared key K to communicate. K is good in that it will be known only by P and Q.

C. ProVeif

ProVerif [20], [21] is an automated tool to analyze the security of a protocol formally. It simulates the concurrent execution of the protocol to be analyzed and the actions of the attackers.

The input for *ProVerif* is a formalized model of the target protocol using Pi calculus or Horn clause.

There are three types of *ProVerif*'s outputs as follows.

1) *RESULT* [*Query*] *Is True*: There are not any potential risks.
2) *RESULT* [*Query*] *Is False*: There are some potential risks.
3) *RESULT* [*Query*] *Cannot Be Proved*: The existence of potential risks is unknown.

If some potential risks exist, *ProVerif* will also show the attack queues for further analyses and improvements.

III. Proposed Protocol

In this section, three phases of our protocol: 1) initialization phase; 2) registration phase; and 3) authentication and key agreement phase, will be detailed. The notations used to describe this protocol are listed in Table I.

In the example process in this section, there are three principals: 1) Alice; 2) Bob; and 3) RC. It is assumed that Bob has registered before Alice.

TABLE I
Symbols Used in the Protocol

Symbol	Meaning
RC	The register center.
ID_x	The identity of the principal x.
$H(.)$	One-way hash function.
sk_x/PK_x	A pair of asymmetric keys for x.
SSK_{ab}/SSK_{ba}	The session key shared between principals A and B.
PSK_{ab}/PSK_{ba}	A part of SSK_{ab}/SSK_{ba}.
$\|, \oplus$	Concatenation operation, bitwise XOR operation.
$\cdot, +$	Point multiplication operation, point geometric addition.

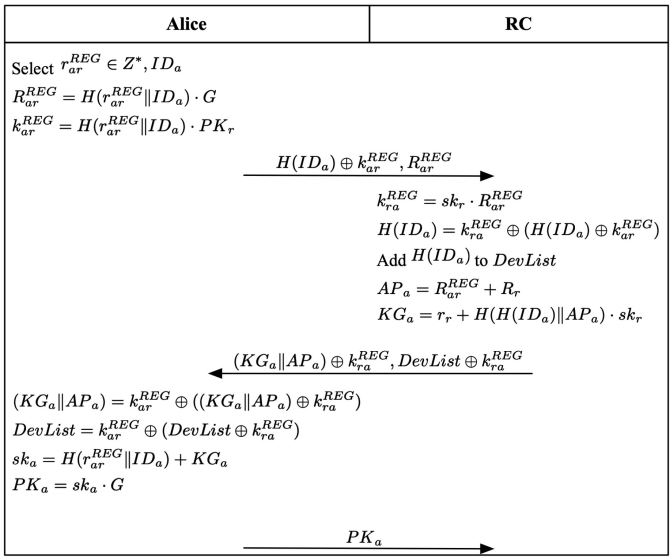

Fig. 2. Registration phase.

A. Initialization Phase

In this phase, RC selects and broadcasts initialization parameters.

Step I1: RC generates a random big prime p and parameters a, b where $4a^3 + 27b^2 (\text{mod } p) \neq 0$, then selects a random point G in the Galois Field F_p of the curve $E(a, b) : y^2 = x^3 + ax + b$ as the base point, and chooses a hash function $H(.)$.

Step I2: RC generates a random number $sk_r \in Z_a^*$ as the long-term private key of RC and stores, and calculates the long-term public key of TRC: $PK_r = sk_r \cdot G$.

Step I3: RC generates a random number r_r, calculates $R_r = r_r \cdot G$, and stores them.

Step I4: RC sends $\{PK_r, G, H(.), E(a, b), p\}$ to all principals via a public channel.

B. Registration Phase

In this phase, principals generates its identity, register itself on RC, and generates its long-term key pair. Following steps describe how Alice registers herself, as shown in Fig. 2.

Step R1: Alice selects a random number $r_{ar}^{\text{REG}} \in Z^*$ and ID_a, and calculates $R_{ar}^{\text{REG}} = H(r_{ar}^{\text{REG}} \| \text{ID}_a) \cdot G$, and then generates the temporary key shared between Alice and RC in this phase: $k_{ar}^{\text{REG}} = H(r_{ar}^{\text{REG}} \| \text{ID}_a) \cdot PK_r$.

Step R2: Alice sends $\{H(\mathrm{ID}_a) \oplus k_{ar}^{\mathrm{REG}}, R_{ar}^{\mathrm{REG}}\}$ to RC via a public channel.

Step R3: RC calculates the temporary key: $k_{ra}^{\mathrm{REG}} = sk_r \cdot R_{ar}^{\mathrm{REG}}$, and extracts $H(\mathrm{ID}_a)' = k_{ra}^{\mathrm{REG}} \oplus (H(\mathrm{ID}_a) \cdot k_{ar}^{\mathrm{REG}})$ and stores $H(\mathrm{ID}_a)$ to DevList (a local datafile to store the hash values of the identities of all registered principals).

Step R4: RC calculates Alice's agreement parameter $AP_a = R_{ar}^{\mathrm{REG}} + R_r$ and key generator $KG_a = r_r + H(H(\mathrm{ID}_a)\|AP_a) \cdot sk_r$.

Step R5: RC generates $r_{rx}^{\mathrm{ADD}} \in Z^*$, calculates $R_{rx}^{\mathrm{ADD}} = r_{rx}^{\mathrm{ADD}} \cdot G$ and a new temporary key $k_{rx}^{\mathrm{ADD}} = r_{rx}^{\mathrm{ADD}} \cdot PK_x$, and sends $\{(KG_a\|AP_a) \oplus k_{ra}^{\mathrm{REG}}, \mathrm{DevList} \oplus k_{ra}^{\mathrm{REG}}\}$ to Alice and $\{H(\mathrm{ID}_a) \oplus k_{rx}^{\mathrm{ADD}}, R_{rx}^{\mathrm{ADD}}\}$ to all registered principals X (except Alice) via public channel.

Step R6: Alice extracts and stores $(KG_a\|AP_a) = k_{ar}^{\mathrm{REG}} \oplus ((KG_a\|AP_a) \oplus k_{ra}^{\mathrm{REG}})$, $\mathrm{DevList} = k_{ar}^{\mathrm{REG}} \oplus (\mathrm{DevList} \oplus k_{ra}^{\mathrm{REG}})$, then generates her long-term asymmetric key pair $sk_a = H(r_{ar}^{\mathrm{REG}}\|\mathrm{ID}_a) + KG_a$, $PK_a = sk_a \cdot G$, and sends $\{PK_a\}$ to RC via public channel.

Step R7: RC stores PK_a.

Step R8: Registered principal X generates the temporary key $k_{xr}^{\mathrm{ADD}} = sk_x \cdot R_{rx}^{\mathrm{ADD}}$, and then extracts and stores the hash value of Alice's identity $H(\mathrm{ID}_a) = k_{xr}^{\mathrm{ADD}} \oplus (H(\mathrm{ID}_a) \oplus k_{rx}^{\mathrm{ADD}})$.

C. Authentication and Key Agreement Phase

In this section, we will show how Alice and Bob negotiate their secret session key. The process of this phase is shown in Fig. 3.

Step AK1: Alice selects random numbers $r_{ab}^{AKA}, r_{ab} \in Z^*$ and fetches $H(\mathrm{ID}_a), H(\mathrm{ID}_b)$ from local storage.

Step AK2: Alice calculates $R_{ab}^{AKA} = r_{ab}^{AKA} \cdot G, R_{ab} = r_{ab} \cdot G$, then calculates the temporary key $k_{ab}^{A} = H(\mathrm{ID}_b) \cdot R_{ab}$, then generates a validation value $V_{a1} = H(H(\mathrm{ID}_a)\|R_{ab}^{AKA}\|R_{ab}\|AP_a)$ for her next message.

Step AK3: Alice sends $\{H(\mathrm{ID}_a) \oplus k_{ab}^{A}, R_{ab}^{AKA}, R_{ab}, AP_a \oplus k_{ab}^{A}, V_{a1}\}$ to Bob via a public channel.

Step AK4: Bob fetches $H(\mathrm{ID}_a), H(\mathrm{ID}_b)$ from local storage.

Step AK5: Bob calculates $k_{ba}^{A} = H(\mathrm{ID}_b) \cdot R_{ab}$, then extracts $H(\mathrm{ID}_a)' = k_{ba}^{A} \oplus (H(\mathrm{ID}_a) \oplus k_{ab}^{A}), AP'_a = k_{ba}^{A} \oplus (AP_a \oplus k_{ab}^{A})$, then calculates $V'_{a1} = H(H(\mathrm{ID}_a)'\|R_{ab}^{AKA}\|R_{ab}\|AP'_a)$, and checks if $V'_{a1} = V_{a1}$, if not, Bob will abort the process.

Step AK6: Bob checks if $H(\mathrm{ID}_a)' = H(ID_a)$, if not, Bob will abort the process.

Step AK6: Bob selects random numbers $r_{ba}^{AKA}, r_{ba} \in Z^*$ and calculates $R_{ba}^{AKA} = r_{ba}^{AKA} \cdot G, R_{ba} = r_{ba} \cdot G$, then generates another temporary key $k_{ba}^{B} = H(\mathrm{ID}_a) \cdot R_{ba}$.

Step AK7: Bob calculates a part of the temporary session key $\mathrm{PSK}_{ba} = r_{ba}^{AKA} \cdot (AP_a + H(H(\mathrm{ID}_a)\|AP_a) \cdot PK_r) + sk_b \cdot R_{ab}^{AKA}$ and generates a validation value $V_b = H(\mathrm{PSK}_{ba}\|H(\mathrm{ID}_b)\|R_{ba}^{AKA}\|R_{ba}\|AP_b)$ for his next message.

Step AK8: Bob sends $\{H(\mathrm{ID}_b) \oplus k_{ba}^{B}, R_{ba}^{AKA}, R_{ba}, AP_b \oplus k_{ba}^{B}, V_b\}$ to Alice via public channel.

Step AK9: Alice calculates $k_{ab}^{B} = H(\mathrm{ID}_a) \cdot R_{ba}$, and extracts $H(\mathrm{ID}_b)' = k_{ab}^{B} \oplus (H(\mathrm{ID}_b) \oplus k_{ba}^{B}), AP'_b = k_{ab}^{B} \oplus (AP_b \oplus k_{ba}^{B})$, calculates a part of temporary session secret key $\mathrm{PSK}_{ab} = r_{ab}^{AKA} \cdot (AP'_b + H(H(\mathrm{ID}_b)'\|AP'_b) \cdot PK_r) + sk_a \cdot R_{ba}^{AKA}$, then calculates $V'_b = H(\mathrm{PSK}_{ab}\|H(\mathrm{ID}'_b)\|R_{ba}^{AKA}\|R_{ba}\|AP'_b)$, and checks if $V'_b = V_b$, if not, Alice will abort the process (we will explain why $\mathrm{PSK}_{ab} = \mathrm{PSK}_{ba}$ at the end of this section).

Step AK10: Alice checks if $H(\mathrm{ID}_b)' = H(\mathrm{ID}_b)$, if not, Alice will abort the process.

Step AK11: Alice calculates the secret session key $\mathrm{SSK}_{ab} = H(H(\mathrm{ID}_a\|H(\mathrm{ID}_b)\|\mathrm{PSK}_{ab})$, then generates a validation value $V_{a2} = H(\mathrm{PSK}_{ab}\|H(\mathrm{ID}_a))$ for her next message.

Step AK12: Alice sends $\{V_{a2}\}$ to Bob via public channel.

Step AK13: Bob calculates $V'_{a2} = H(\mathrm{PSK}_{ba}\|H(\mathrm{ID}_a))$, and checks if $V'_{a2} = V_{a2}$, if not, Bob will abort the process.

Step AK14: Bob calculates the secret session key $\mathrm{SSK}_{ba} = H(H(\mathrm{ID}_a)\|H(\mathrm{ID}_b)\|\mathrm{PSK}_{ba})$.

Now, Alice and Bob have finished negotiating and agreed to use $\mathrm{SSK}_{ab}/\mathrm{SSK}_{ba}$ as the secret session key.

Proof:

$$
\begin{aligned}
&\mathrm{PSK}_{ab} = \mathrm{PSK}_{ba}\\
&\mathrm{PSK}_{ba}\\
&= r_{ba}^{AKA} \cdot (AP_a + H(H(\mathrm{ID}_a)\|AP_a) \cdot PK_r) + sk_b \cdot R_{ab}^{AKA}\\
&= r_{ba}^{AKA} \cdot ((H(r_{ab}^{\mathrm{REG}}\|\mathrm{ID}_a) + KG_a) \cdot G) + sk_b \cdot R_{ab}^{AKA}\\
&= r_{ba}^{AKA} \cdot PK_{ab} + sk_{ba} \cdot R_{ab}^{AKA}\\
&= r_{ba}^{AKA} \cdot sk_{ab} \cdot G + sk_{ba} \cdot r_{ab}^{AKA} \cdot G\\
&= sk_a \cdot R_{ba}^{AKA} + PK_b \cdot r_{ab}^{AKA}\\
&= sk_a \cdot R_{ba}^{AKA} + r_{ab}^{AKA} \cdot (AP_b + H(H(\mathrm{ID}_b)\|AP_b) \cdot PK_r)\\
&= \mathrm{PSK}_{ab}.
\end{aligned}
$$

■

D. Sign Out Phase

If a principal wants to sign out, it only need to notify RC, and RC will notify other principals. The following steps describe how Alice signs herself out.

Step S1: Alice selects a random number $r_{ar}^{SO} \in Z^*$, calculates $R_{ar}^{SO} = H(r_{ar}^{SO}\|ID_a) \cdot G$ and the temporary key $k_{ar}^{SO} = H(r_{ar}^{SO}\|\mathrm{ID}_a) \cdot PK_r$, then sends $\{H(\mathrm{ID}_a) \oplus k_{ar}^{SO}, R_{ar}^{SO}\}$ to RC via a public channel.

Step S2: RC calculates $k_{ra}^{SO} = sk_r \oplus R_{ar}^{SO}$, then extracts $H(\mathrm{ID}_a) = k_{ra}^{SO} \oplus (H(\mathrm{ID}_a \oplus k_{ar}^{SO}))$.

Step S3: RC checks if $H(\mathrm{ID}_a)$ has been stored in its local storage, if not, it means that has never registered before, RC will abort the process.

Step S4: RC selects a random number $r_{rx}^{SO} \in Z^*$, calculates $R_{rx}^{SO} = r_{rx}^{SO} \cdot G$ and the temporary key $k_{rx}^{SO} = r_{rx}^{SO} \cdot PK_x$, then sends $\{H(\mathrm{ID}_a \oplus k_{rx}^{SO}), R_{rx}^{SO}\}$ to all registered principals X (except Alice), and delete $H(\mathrm{ID}_a), PK_a$ from its local storage.

Step S5: Registered principal X calculates $k_{xr}^{SO} = sk_x \cdot R_{rx}^{SO}$, and extracts $H(\mathrm{ID}_a) = k_{xr}^{SO} \oplus (H(\mathrm{ID}_a) \oplus k_{rx}^{SO})$, then removes $H(\mathrm{ID}_a)$ from local *DevList*.

IV. Security Analyses

In this section, we will formally prove that our protocol has unknown secret key-share attack resistance, and informally prove that our protocol also meets other security requirements. Furthermore, we will use *ProVerif* to verify the security of our protocol.

Fig. 3. Authentication and key agreement phase.

A. Formal Security Analysis

In this section, we will use BAN logic to analyze our protocol formally. The following BAN logic basic rules will be inferred in this section.

1) *Rule1*: $((P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft \{M\}_K)/(P \mid\equiv Q \mid\sim M))$ (If P believes the secret key K is only shared between P and Q, and P sees the statement $\{M\}_K$, then P believes Q once said the statement M).
2) *Rule2*: $((P \mid\equiv \#(M), P \mid\equiv Q \mid\sim M)/(P \mid\equiv Q \mid\equiv M))$ (If P believes the statement M is fresh, and P believes Q once said M, then P believes Q believes M).
3) *Rule3*: $((P \mid\equiv \#(M), P \mid\equiv Q \mid\equiv M)/(P \mid\equiv P \overset{K}{\longleftrightarrow} Q))$ (If P believes the statement M is fresh, and P believes Q believes M, then P believes the secret key K is only shared between P and Q).
4) *Rule4*: $((P \mid\equiv Q \Rightarrow M, P \mid\equiv Q \mid\equiv M)/(P \mid\equiv M))$ (If P believes Q has jurisdiction over the statement M, and P believes Q believes M, then P believes M).
5) *Rule5*: $((P \mid\equiv \#(M))/(P \mid\equiv \#(M, N)))$ (If P believes the statement M is fresh, then P believes the statement (M, N) is also fresh).
6) *Rule6*: $((P \mid\equiv (M, N))/(P \mid\equiv M))$ (If P believes the statement (M, N), then P believes the statement M), $((P \mid\equiv Q \mid\equiv (M, N))/(P \mid\equiv Q \mid\equiv M))$ (If P believes Q believes the statement (M, N), then P believes Q believes the statement M).
7) *Rule7*: $((P \triangleleft (M, N))/(P \triangleleft M))$, $((P \triangleleft < M >_N)/(P \triangleleft M))$ (If P sees the statement (M, N) or the statement $< M >_N$, then P sees the statement M).
8) *Rule8*: $((P \mid\equiv Q \mid\sim (M, N))/(P \mid\equiv Q \mid\sim M))$ (If P believes Q once said the statement (M, N), then P believes Q once said the statement M).

For convenience, let SSK_{ab} and SSK_{ba} be *SSK* here, then we can model the protocol as follows.

1) *Message1:* $A \rightarrow B : H(\mathrm{ID}_a), R_{ab}^{AKA}, AP_a, R_{ab}, V_{a1}$: $\{H(\mathrm{ID}_a), R_{ab}^{AKA}, < r_{ar}^{\mathrm{REG}}, \mathrm{ID}_a >_{r_r}, R_{ab}, (H(\mathrm{ID}_a), R_{ab}^{AKA}, R_{ab}, AP_a)\}$.
2) *Message2:* $B \rightarrow A : H(\mathrm{ID}_b), R_{ba}^{AKA}, AP_b, R_{ba}, V_b$: $\{H(\mathrm{ID}_b), R_{ba}^{AKA}, < r_{br}^{\mathrm{REG}}, \mathrm{ID}_B >_{r_r}, R_{ba}, (\mathrm{PSK}_{ba}, H(\mathrm{ID}_b), R_{ba}^{AKA}, R_{ba}, AP_b)\}$.
3) *Message3:* $A \rightarrow B : V_a : \{(\mathrm{PSK}_{ab}, H(\mathrm{ID}_a))\}$.

Set the following security goals.

1) *Goal1:* $B \mid\equiv A \overset{\mathrm{SSK}}{\longleftrightarrow} B$.
2) *Goal2:* $B \mid\equiv A \mid\equiv B \overset{\mathrm{SSK}}{\longleftrightarrow} A$.
3) *Goal3:* $A \mid\equiv B \overset{\mathrm{SSK}}{\longleftrightarrow} A$.
4) *Goal4:* $A \mid\equiv B \mid\equiv B \overset{\mathrm{SSK}}{\longleftrightarrow} A$.

The preliminary assumptions are as follows.

1) *Pre1:* $A \mid\equiv \mathrm{PSK}_{ba}, AP_b, r_{ba}^{AKA}$.
2) *Pre2:* $B \mid\equiv \mathrm{PSK}_{ab}, AP_a, r_{ab}^{AKA}$.
3) *Pre3:* $A \mid\equiv B \Rightarrow AP_b$.
4) *Pre4:* $B \mid\equiv A \Rightarrow AP_a$.
5) *Pre5:* $A \mid\equiv \#(r_{ba}^{AKA})$.
6) *Pre6:* $B \mid\equiv \#(r_{ab}^{AKA})$.

Now, to analyze the security of our protocol, the following steps need to be taken.

Step 1: With *Rule*7 and *Message*1, we have $B \triangleleft \{H(\mathrm{ID}_a), R_{ab}^{AKA}, <\ r_{ar}^{\mathrm{REG}}, \mathrm{ID}_a\ >_{(r_r)}, R_{ab}, (H(\mathrm{ID}_a), R_{ab}^{AKA}, R_{ab}, AP_a)\}$.

Step 2: With *Rule*1, step 1, and *Pre*1, we have $B \mid\equiv A \mid\sim \{H(\mathrm{ID}_a), R_{ab}^{AKA}, <\ r_{ar}^{\mathrm{REG}}, \mathrm{ID}_a\ >_{(r_r)}, R_{ab}, (H(\mathrm{ID}_a), R_{ab}^{AKA}, R_{ab}, AP_a)\}$.

Step 3: With *Pre*4, step 2, and *Rule*2, we have $B \mid\equiv A \mid\equiv \{H(\mathrm{ID}_a), R_{ab}^{AKA}, <\ r_{ar}^{\mathrm{REG}}, \mathrm{ID}_a\ >_{(r_r)}, R_{ab}, (H(\mathrm{ID}_a), R_{ab}^{AKA}, R_{ab}, AP_a)\}$.

Step 4: With step 3 and *Rule*4, we can prove *Goal*2: $B \mid\equiv A \mid\equiv B \overset{\mathrm{SSK}}{\longleftrightarrow} A$

Step 5: With *Pre*6, step 4, and *Rule*3, we can prove *Goal*1: $B \mid\equiv A \overset{\mathrm{SSK}}{\longleftrightarrow} B$.

Step 6: With *Rule*7 and *Message*2, we have $A \triangleleft \{H(\mathrm{ID}_b), R_{ba}^{AKA}, <\ r_{br}^{\mathrm{REG}}, \mathrm{ID}_b\ >_(\ r_r), R_{ba}, (\mathrm{PSK}_{ba}, H(\mathrm{ID}_b), R_{ba}^{AKA}, R_{ba}, AP_b)\}$.

Step 7: With *Rule*1, step 6, and *Pre*2, we have $A \mid\equiv B \mid\sim \{H(\mathrm{ID}_b), R_{ba}^{AKA}, <\ r_{br}^{\mathrm{REG}}, \mathrm{ID}_b\ >_{(r_r)}, R_{ba}, (\mathrm{PSK}_{ba}, H(\mathrm{ID}_b), R_{ba}^{AKA}, R_{ba}, AP_b)\}$.

Step 8: With *Pre*3, step 7, and *Rule*2, we have $B \mid\equiv A \mid\sim \{H(\mathrm{ID}_b), R_{ba}^{AKA}, <\ r_{br}^{\mathrm{REG}}, \mathrm{ID}_b\ >_{(r_r)}, R_{ba}, (\mathrm{PSK}_{ba}, H(\mathrm{ID}_b), R_{ba}^{AKA}, R_{ba}, AP_b)\}$.

Step 9: With step 8 and *Rule*4, we can prove *Goal*4: $A \mid\equiv B \mid\equiv B \overset{\mathrm{SSK}}{\longleftrightarrow} A$.

Step 10: With *Pre*5, step 9, and *Rule*3, we can prove *Goal*3: $A \mid\equiv B \overset{\mathrm{SSK}}{\longleftrightarrow} A$.

Now, we have proved all the security goals we set before, both Alice and Bob trust the temporary secret key SSK, and our protocol has unknown key-share attack resistance.

B. Descriptive Security Analyses

In this section, we will analyze the proposed security informally, and prove that our protocol meets other critical security requirements.

1) Anonymity: In our protocol, any message transmitted through public channels will never include principals' identities directly. The identity of a principal sent via the public channel will only be a hash value. If an attacker $\mathcal{A}$ grabs a message including the hash value $H(\mathrm{ID}_x)$ of an identity ID_x, it is impossible for $\mathcal{A}$ to extract ID_x, as the hash operation is irreversible. So, our protocol can keep the principals anonymous.

2) Message Integrity: If the attacker $\mathcal{A}$ modified the message transferred via public channel, the receiver will find it at once, for the hash value of the modified message will not equals to the original one. At the same time, $\mathcal{A}$ also cannot recalculate the hash value, for it cannot obtain the secret key to decrypt the message. So, our protocol can guarantee the message integrity.

3) Impersonation Attack Resistance: If an attacker $\mathcal{A}$ wants to impersonate Alice, it must change the message from $\{H(\mathrm{ID}_A) \oplus k_{ab}^A, R_{ab}^{AKA}, R_{ab}, AP_a \oplus k_{ab}^A, V_{a1}\}$ to the fake one $\{H(\mathrm{ID}_A) \oplus k_{ab}^A, R_{ab}^{AKA*}, R_{ab}^*, AP_a \oplus k_{ab}^A, V_{a1}^*\}$ ('*' means that the parameter has been modified by $\mathcal{A}$). But $\mathcal{A}$ cannot generate correct V_{a1} for it cannot extract $H(\mathrm{ID}_A)$ and AP_a from the message. Moreover, $\mathcal{A}$ also cannot calculate PSK_{ab} and V_{a2} correctly. When Bob receives the fake message generated by $\mathcal{A}$, he will check if $V_{a1}^* = V_{a1}$ and $V_{a2}^* = V_{a2}$, and this will of course fail, so he will abort the process, which means that $\mathcal{A}$ fails to impersonate Alice. Similarly, $\mathcal{A}$ also cannot impersonate Bob, as $V_b = H(\mathrm{PSK}_{ab} \| H(\mathrm{ID}_b) \| R_{ba}^{AKA} \| R_{ba} \| AP_b)$ will not equals to V_b^* generated by $\mathcal{A}$. So, our protocol has impersonation attack resistance.

4) Man-in-the-Middle Attack Resistance: The attacker $\mathcal{A}$ cannot obtain Alice's or Bob's private key, which means that it cannot generate correct PSK_{ab} or PSK_{ba}. In this situation, $\mathcal{A}$ cannot generate correct validation values V_{a1}, V_{a2}, and V_b, which are necessary if it wants to modify the message. Moreover, the protocol has mutual authentication. So, our protocol has man-in-the-middle attack resistance.

5) Replay Attack Resistance: In replay attacks, attackers steal valid messages from public channels, and then replays them ceaselessly. But in our protocol, every time when a new message is going to be sent, a new random will be generated and used to update temporary parameters. Therefore, the old validation value will become invalid, and $\mathcal{A}$ cannot generate the new one. So, our protocol can resist replay attack.

6) Known Session-Specific Temporary Information Attack Resistance: The attacker $\mathcal{A}$ can intercept messages $\{H(\mathrm{ID}_a) \oplus k_{ab}^A, R_{ab}^{AKA}, R_{ab}, AP_a \oplus k_{ab}^A, V_{a1}\}$ and $\{H(\mathrm{ID}_b) \oplus k_{ba}^B, R_{ba}^{AKA}, R_{ba}, AP_b \oplus k_{ba}^B, V_b\}$ from the public channel in the authentication and key agreement phase of the protocol, and extract $H(\mathrm{ID}_a) \oplus k_{ab}^A, AP_a \oplus k_{ab}^A$ and $H(\mathrm{ID}_b) \oplus k_{ba}^B, AP_b \oplus k_{ba}^B$ from the messages intercepted. But $\mathcal{A}$ cannot obtain the $H(\mathrm{ID}_a), AP_a$ or $H(\mathrm{ID}_b), AP_b$, or other temporary session information; hence, it still cannot calculate the temporary secret key SSK. So, our protocol has known session-specific temporary message attack resistance.

7) Perfect Forward Secrecy: The attacker $\mathcal{A}$ can collect temporary secret keys generated in the past from the public channel, but it still cannot calculate the current temporary secret key *SSK*, for each temporary secret key only relates to the corresponding temporary session information. Furthermore, even if A obtains PSK_{ab} and PSK_{ba}, and intercepts the messages $\{H(ID_a) \oplus k_{ab}^A, R_{ab}^{AKA}, R_{ab}, AP_a \oplus k_{ab}^A, V_{a1}\}$ and $\{H(\mathrm{ID}_b) \oplus k_{ba}^B, R_{ba}^{AKA}, R_{ba}, AP_b \oplus k_{ba}^B, V_b\}$, it is still unable to extract temporary session information from them to calculate the temporary secret key. So, our protocol has perfect forward secrecy.

C. Simulation for Formal Security Analysis Using ProVerif

In this section, the automated tool ProVerif is employed to analyze the security of our protocol formally.

TABLE II
SIMULATION RESULTS

Assertion	Target	Result
query inj-event($endAuthA$)==>inj-event($startAuthA$)	Consistence	RESULT inj-event($endAuthA$)==>inj-event($startAuthA$) is TRUE
query inj-event($endAuthB$)==>inj-event($startAuthB$)	Consistence	RESULT inj-event($endAuthB$)==>inj-event($startAuthB$) is TRUE
query attacker(SSK_{ab})	Temporary Secret Key Security	RESULT NOT attacker(SSK_{ab}) is TRUE
query attacker(SSK_{ba})	Temporary Secret Key Security	RESULT NOT attacker(SSK_{ba}) is TRUE
query attacker(ID_a)	Anonymity	RESULT NOT attacker(ID_a) is TRUE
query attacker(ID_b)	Anonymity	RESULT NOT attacker(ID_b) is TRUE

Algorithm 1 Formalized Model of Alice

$R_{ab}^{AKA} = r_{ab}^{AKA} \cdot G$
$R_{ab} = r_{ab} \cdot G$
$k_{ab}^{A} = hash(ID_b) \cdot r_{ab} \cdot G$
$V_{a1} = hash(hash(ID_a) \| R_{ab}^{AKA} \| R_{ab} \| AP_a)$
$(hash(ID_a) \oplus k_{ab}^{A}, R_{ab}^{AKA}, R_{ab}, AP_a, V_{a1}) \rightarrow$ **channel**
event *startAuthB*
$(SHid_b, xR_{ba}^{AKA}, xR_{ba}, xAP_b, xV_b) \leftarrow$ **channel**
$k_{ab}^{B} = hash(ID_a) \cdot xR_{ba}$
$xHid_b = SHid_b \oplus k_{ab}^{B}$
if $(xHid_b = hash(ID_b))$ **then**
  $PSK_{ab} = r_{ab} \cdot (AP_a + hash(xHid_b \| xAP_b) \cdot PK_r) + sk_a \cdot xR_{ba}^{AKA}$
  $xxV_b = hash(PSK_{ab} \| xHid_b \| xR_{ba}^{AKA} \| xR_{ba} \| xAP_b)$
  if $(xxV_b = xV_b)$ **then**
    $SSK_{ab} = hash(hash(ID_a) \| xHid_b \| PSK_{ab})$
    $V_{a2} = hash(PSK_{ab} \| hash(ID_a))$
    $(V_{a2}) \rightarrow$ **channel**
    event *endAuthA*

Algorithm 2 Formalized Model of Bob

event *startAuthA*
$(SHid_a, xR_{ab}^{AKA}, xR_{ab}, xAP_a, xV_{a1}) \leftarrow$ **channel**
$k_{ba}^{A} = hash(ID_b) \cdot xR_{ab}$
$xHid_a = SHid_a \oplus k_{ba}^{A}$
if $(xHid_a = hash(ID_a))$ **then**
  $xxV_{a1} = hash(xHid_a \| xR_{ab}^{AKA} \| xR_{ab} \| xAP_a)$
  if $(xxV_{a1} = xV_{a1})$ **then**
    $PSK_{ba} = r_{ba} \cdot (AP_b + hash(xHid_a \| xAP_a) \cdot PK_r) + sk_b \cdot xR_{ab}^{AKA}$
    $R_{ba}^{AKA} = r_{ba}^{AKA} \cdot G$
    $R_{ba} = r_{ba} \cdot G$
    $k_{ba}^{B} = xHid_a \cdot r_{ba} \cdot G$
    $V_b = hash(PSK_{ba} \| hash(ID_b) \| R_{ba}^{AKA} \| R_{ba} \| AP_b)$
    $(hash(ID_b) \oplus k_{ba}^{B}, R_{ba}^{AKA}, R_{ba}, AP_b, V_b)) \rightarrow$ **channel**
    $(xV_{a2}) \leftarrow$ **channel**
    $xxV_{a2} = hash(PSK_{ba} \| xHid_a)$
    if $(xxV_{a2} = xV_{a2})$ **then**
      $SSK_{ba} = hash(xHid_a \| hash(ID_b) \| PSK_{ba})$
      event *endAuthB*

1) Formalized Model: Before analyzing the protocol, formalized models needs to be built. The sign **channel** in the models represents the channel that Alice and Bob use to communicate with each other.

Alice's model of the AKA phase is shown in Algorithm 1, and Bob's model of the AKA phase is shown in Algorithm 2.

2) Security Assertions: To analyze the security of the protocol, we will define the assertions of event consistence, secrecy, and anonymity in this section.

The event consistence assertion is defined as follows.

1) query inj-event(*endAuthA*)==>inj-event(*startAuthA*).
2) query inj-event(*endAuthB*)==>inj-event(*startAuthB*).

The first statement means that if Alice starts to authenticate Bob's identity, she will finally finish it in the same session. The meaning of the second statement is similar. This assertion can prove that the protocol has mutual authentication and can resist impersonation attacks.

The secrecy assertion is defined as follows.

1) query attacker(SSK_{ab}).
2) query attacker(SSK_{ba}).

This assertion checks if attackers can obtain the session key SSK. It can also prove that the proposed protocol meets the security requirement of known session-specific temporary information attack resistance.

The anonymity assertion is defined in the following.

1) query attacker(ID_a).
2) query attacker(ID_b).

This assertion checks if attackers can obtain the real identity messages ID_a and ID_b.

3) Simulation Results: The result of simulation is shown in Table II, corresponding to the event consistence assertion, secrecy assertion, and anonymity assertion.

The first and second rows of the table show that each time before event *endAuthA* (or *endAuthB*) executes, event *startAuthA* (or *startAuthB*) must have already finished. This means the authentication of Alice (or Bob) always starts and finishes in the same session. It shows that our protocol has mutual authentication and can resist impersonation attacks.

The third and forth rows show that attackers cannot obtain neither Alice's temporary secret key SSK_{ab} nor Bob's temporary secret key SSK_{ba}, and proves that our protocol can resist known session-specific temporary information attacks.

The fifth and sixth rows show that attackers cannot know neither Alice's real identity ID_a nor Bob's real identity ID_b, which means that our protocol can always keep the principals anonymous.

V. PERFORMANCE EVALUATION

In this section, the security and performance, including computational and communication overheads, of our protocol will be compared with other existing protocols.

TABLE III
SECURITY COMPARISON

Security Features	Odelu *et al.* [9]	Khan *et al.* [22]	Garg *et al.* [23]	**Proposed**
MA	✓	✓	✓	✓
UKSA	✓	✓	✓	✓
PA	×	×	×	✓
MI	×	×	×	✓
PFS	✓	×	✓	✓
KKS	✓	✓	✓	✓
RA	×	✓	×	✓
IA	✓	×	✓	✓
KSTI	×	×	✓	✓

MA: Mutual Authentication;
UKSA: Unknown Key-Share Attack resistance;
PA: Principal Anonymity;
MI: Message Integrity;
PFS: Perfect Forward Secrecy;
KKS: Known Key Secrecy;
RA: Replay Attack resistance;
IA: Impersonation Attack resistance;
KSTI: Known Session-specific Temporary Information attack resistance

A. Security

The proposed protocol is compared with following protocols: the protocol proposed by Odelu et al. [9], a provably secure authenticated key distribution scheme based on ECC; the protocol proposed by Khan et al. [22], a password-based anonymous lightweight AKA scheme which is based on ECC and for smart grid; and the protocol proposed by Garg et al. [23], an ECC-based secure and lightweight authentication scheme for smart metering infrastructure in smart grid. The result in Table III indicates that the security of our protocol is better than others.

B. Performance

1) Computational Overhead: The computational overhead of our protocol will be compared with the protocol proposed by Odelu et al. [9], the protocol proposed by Khan et al. [22], and the protocol proposed by Garg et al. [23] in this section. It is assumed that T_{pm}, T_{pa}, T_h, T_{XOR}, T_{bp}, T_{em}, and T_{sec} denote the computational time required for a point multiplication, a point addition, a one-way hash, a bitwise XOR operation, a bilinear pairing, a modular exponentiation of large numbers, and a symmetric encryption. The operations are simulated on Windows 11 with Intel Core i5-9300H@2.4 GHz and 8-GB RAM. The computational overhead of T_{pm}, T_{pa}, T_h, T_{XOR}, T_{bp}, T_{em}, T_{sec} is 0.833, 0.041, 0.004, 0.003, 2.891, 0.065, and 0.053 ms, and the computational overheads of existing protocols and our protocol are described in Table IV. Fig. 4 shows the computational overhead of our protocol and other protocols.

The table and the figure indicate that our protocol has lower computer overhead than the protocol proposed by Odelu et al. [9], the protocol proposed by Khan et al. [22], and the protocol proposed by Garg et al. [23]

2) Communication Overhead: The communication overhead of our protocol will be compared with the protocol proposed by Odelu et al. [9], the protocol proposed by Khan et al. [22], and the protocol proposed by Garg et al. [23] in this section. It is assumed that

TABLE IV
COMPUTATIONAL OVERHEAD

Protocol	Operations	Computational Overhead (ms)
Odelu *et al.* [9]	$5T_{pm} + 2T_{pa} + 12T_h + 2T_{bp} + 2T_{XOR} + 2T_{em}$	10.313
Khan *et al.* [22]	$8T_{pm} + 19T_h + 9T_{XOR} + 4T_{sec}$	6.979
Garg *et al.* [23]	$8T_{pm} + 2T_{pa} + 8T_h$	6.778
Proposed	$7T_{pm} + 2T_{pa} + 10T_h + 6T_{XOR}$	5.971

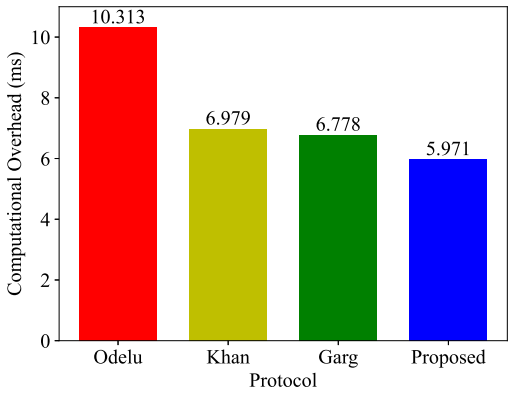

Fig. 4. Computational overhead comparison.

TABLE V
COMMUNICATION OVERHEAD

Protocol	Messages	Communication Overhead (bit)
Odelu *et al.* [9]	$2L_p + 3L_h + L_r + L_g + L_{id}$	2432
Khan *et al.* [22]	$2L_p + 2L_{sec} + 2L_t + 3L_{id} + 4L_h$	2336
Garg *et al.* [23]	$2L_p + 6L_h + 2L_{id} + 2L_r + 4L_t$	2304
Proposed	$4L_p + 6L_h$	2240

$L_h, L_{id}, L_p, L_t, L_r, L_g$, and L_{sec} denote the length of a 160-bit hash message [7], [9], [24], [25], a 160-bit identity message [7], [9], [24], [25], a 320-bit elliptic curve point message [7], [24], [25], a 32-bit timestamp message [7], [9], [24], a 128-bit random number message [9], [25], a 1024-bit bilinear pairing parameter [25], and a 256*n-bit AES symmetric block encryption message, where n is the count of the blocks to be encrypted [26] (here we assume $n = 1$). The communication overheads of existing protocols and our protocol are described in Table V. Fig. 5 shows the communication overhead of our protocol and other protocols.

The table and the figure indicate that our protocol has lower communication overhead than the protocol proposed by Odelu et al. [9], the protocol proposed by Khan et al. [22], and the protocol proposed by Garg et al. [23]

C. Performance in Practical Usage

The protocol we proposed in this article has been applied in a real ICS to enhance the communication security and reduce the resource consumption. After applying our protocol, the average decrease of the nodes' CPU loads is 10.3%, as well

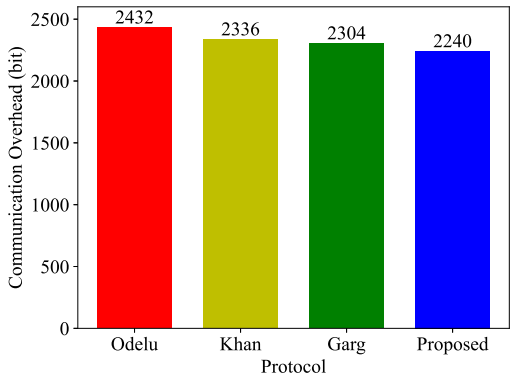

Fig. 5. Communication overhead comparison.

as the network load and the communication delay are also reduced significantly.

VI. Conclusion

This work proposed a new lightweight AKA protocol based on ECC for ICS. The proposed protocol employed the ECC algorithm instead of traditional encryption algorithms to reduce computational and communication overheads. After analyzing the security of our protocol formally and informally, it was proved that our protocol has met all of security requirements mentioned before. The comparison of security and computational and communication overheads indicated that our protocol has better security and performance than existing protocols, and performance of the protocol in the practical usage showed that our solution does benefit the ICS in the aspect of resource consumption significantly.

References

[1] A. Ometov, V. Petrov, S. Bezzateev, S. Andreev, Y. Koucheryavy, and M. Gerla, "Challenges of multi-factor authentication for securing advanced IoT applications," *IEEE Netw.*, vol. 33, no. 2, pp. 82–88, Mar./Apr. 2019.

[2] J. Wang, J. Li, H. Wang, L. Y. Zhang, L.-M. Cheng, and Q. Lin, "Dynamic scalable elliptic curve cryptographic scheme and its application to in-vehicle security," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 5892–5901, Aug. 2019.

[3] Y. Zheng et al., "Design and analysis of a security-enhanced three-party authenticated key agreement protocol based on chaotic maps," *IEEE Access*, vol. 8, pp. 66150–66162, 2020.

[4] H. Fang, A. Qi, and X. Wang, "Fast authentication and progressive authorization in large-scale IoT: How to leverage AI for security enhancement," *IEEE Netw.*, vol. 34, no. 3, pp. 24–29, May/Jun. 2020.

[5] P. Tedeschi, S. Sciancalepore, A. Eliyan, and R. Di Pietro, "LiKe: Lightweight certificateless key agreement for secure IoT communications," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 621–638, Jan. 2020.

[6] J.-L. Tsai and N.-W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 906–914, Mar. 2016.

[7] H. Debiao, W. Huaqun, K. M. Khurram, and L. Wang, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Commun.*, vol. 10, no. 14, pp. 1795–1802, 2016.

[8] S. Han, S. Jian, K. M. Khurram, and L. Jong-Hyouk, "Efficient RFID authentication using elliptic curve cryptography for the Internet of Things," *Wireless Pers. Commun.*, vol. 96, no. 4, pp. 5253–5266, 2017.

[9] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[10] 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 Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[11] M. E. S. Saeed, Q.-Y. Liu, and F. Li, "AKAIoTs: Authenticated key agreement for Internet of Things," *Wireless Netw.*, vol. 25, no. 6, pp. 3081–3101, 2019.

[12] D. Abbasinezhad-Mood and M. N. Nikoughadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Future Gener. Comput. Syst.*, vol. 84, pp. 47–57, Jul. 2018.

[13] A. Lohachab and Karambir, "ECC based inter-device authentication and authorization scheme using MQTT for IoT networks," *J. Inf. Security Appl.*, vol. 46, pp. 1–12, Jun. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212618306513

[14] C. T. Poomagal and G. A. S. Kumar, "ECC based lightweight secure message conveyance protocol for satellite communication in Internet of Vehicles (IoV)," *Wireless Pers. Commun.*, vol. 113, no. 2, pp. 1359–1377, 2020.

[15] D. Singh, B. Kumar, S. Singh, and S. Chand, "A secure IoT-based mutual authentication for healthcare applications in wireless sensor networks using ECC," *Int. J. Healthcare Inf. Syst. Inform.*, vol. 16, no. 2, pp. 21–48, 2021.

[16] Y. Yan, R. Q. Hu, S. K. Das, H. Sharif, and Y. Qian, "An efficient security protocol for advanced metering infrastructure in smart grid," *IEEE Netw.*, vol. 27, no. 4, pp. 64–71, Jul./Aug. 2013.

[17] D. B. Rawat and C. Bajracharya, "Cyber security for smart grid systems: Status, challenges and perspectives," in *Proc. SoutheastCon*, 2015, pp. 1–6.

[18] S. Shapsough, F. Qatan, R. Aburukba, F. Aloul, and A. R. Al Ali, "Smart grid cyber security: Challenges and solutions," in *Proc. Int. Conf. Smart Grid Clean Energy Technol. (ICSGCE)*, 2015, pp. 170–175.

[19] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.

[20] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 824–839, Sep./Oct. 2018.

[21] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1953–1966, 2015.

[22] A. A. Khan, V. Kumar, M. Ahmad, S. Rana, and D. Mishra, "PALK: Password-based anonymous lightweight key agreement framework for smart grid," *Int. J. Electr. Power Energy Syst.*, vol. 121, Oct. 2020, Art. no. 106121. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061519340621

[23] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3548–3557, May 2020.

[24] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for Industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1133–1146, Nov./Dec. 2020.

[25] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 7996–8004, Oct. 2018.

[26] J. Cui, Y. Wang, J. Zhang, Y. Xu, and H. Zhong, "Full session key agreement scheme based on chaotic map in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8914–8924, Aug. 2020.

Yanru Chen (Member, IEEE) received the master's degree in management (finance) from the University of Melbourne, Melbourne, VIC, Australia, in 2017. She is currently pursuing the Ph.D. degree with the Embedded Systems and Big Data Management Laboratory and the Internet of Things Laboratory, College of Computer Science, Sichuan University, Chengdu, China, under the supervision of Prof. B. Guo.

Her research interests include blockchains, wireless sensor networks, cognitive psychology, personal financial big data, computational finance, and financial data space.

Fengming Yin received the B.E. degree from Harbin Institute of Technology, Weihai, China, in 2021. He is currently pursuing the master's degree with the College of Computer Science, Sichuan University, Chengdu, China, under the supervision of Prof. L. Chen.

His research interests include Industrial Internet of Things and authenticated key agreement protocol.

Shunfang Hu (Member, IEEE) received the B.E. and M.E. degrees from Sichuan University, Chengdu, China, in 2002 and 2005, respectively, where she is currently pursuing the Ph.D. degree with the School of Computer Science.

She is also a Lecturer with the School of Mathematics and Computer Science, Yunnan Minzu University, Kunming, China. Her research interests include cyberspace security, embedded systems, wireless sensor networks, and Internet of Things and their applications.

Limin Sun (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the College of Computer, National University of Defense Technology, Changsha, China, in 1988, 1995, and 1998, respectively.

He is currently working as a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interests include IoT security and privacy, ICS security, wireless sensor networks, and Internet of Things.

Dr. Sun is a Senior Member of the China Computer Federation.

Yang Li received the B.S. degree in physics and the Ph.D. degree in radio physics from Peking University, Beijing, China, in 2007 and 2012, respectively.

He is currently a Senior Engineer with the Department of Science and Technology, Communication Security Laboratory, Institute of Southwestern Communication, Chengdu, China. His research interests include blockchains, quantum cryptography, quantum secure communication, and quantum information.

Bin Xing received the Ph.D. degree in data analysis and processing from the Central Polytechnic University of Paris (École Centrale de Paris), Paris, France, in 1998.

He worked with DASSULT Group, Paris, and ATOS France, Bezons, France, as a Senior Data Consultant, a Data Analysis Scientist, and is currently the Chief Scientist of National Engineering Laboratory of Industrial Big-Data Application Technology, CASICloud-Tech Company, Ltd., Beijing, China, and the Chief Scientist of Chongqing Innovation Center, Industrial Big-Data Company, Ltd., Chongqing, China.

Liangyin Chen (Member, IEEE) received the Ph.D. degree from the College of Computer Science, Sichuan University, Chengdu, China, in 2008.

He joined the School of Computer Science, Sichuan University and has been a Professor since 2014. He works as the Director of Sichuan Big Data Analysis and Fusion Application Technology Engineering Laboratory. He was a Visiting Researcher with the University of Minnesota, Minneapolis, MN, USA, from 2009 to 2010, under the supervision of Prof. T. He. He has authored or coauthored more than 100 papers, many of which were published in premier network journals and conferences, such as the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Knowledge-Based Systems*, IEEE International Conference on Computer Communications, ACM International Conference on Multimedia, IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, and ACM Conference on Embedded Networked Sensor Systems. His research interests include wireless sensor networks, Intelligent Internet of Things and IoT security, Industrial Internet, blockchains, and big data.

Bing Guo received the M.S. and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 1999 and 2002, respectively.

He is currently a Professor with Sichuan University, Chengdu. His current research interests include blockchains, green computing, and big personal data.