

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

H. Qiao *et al.*, "Anonymous Lightweight Authenticated Key Agreement Protocol for Fog-Assisted Healthcare IoT System," in *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2023.3270300.

<https://doi.org/10.1109/JIOT.2023.3270300>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Anonymous Lightweight Authenticated Key Agreement Protocol for Fog-Assisted Healthcare IoT System

Hui Qiao, Xuewen Dong, Qi Jiang, Siqi Ma, Chao Liu, Ning Xi, Yulong Shen

Abstract—The impact of fog-assisted Healthcare IoT (H-IoT) system is immense. The smart H-IoT equipments can upload healthcare information to fog nodes with low latency and high mobility. To facilitate secure interactions among three parties, including smart H-IoT equipments, fog nodes and a cloud server, over the public and insecure channels, a few authenticated key agreement (AKA) protocols are proposed. However, existing works are constructed based on expensive cryptographic primitives (e.g., bilinear pairing), which lead to high computation costs. Besides, the anonymity of H-IoT users is failed to be provided. To tackle these issues, an anonymous and lightweight three-party AKA protocol ALAKAP is proposed, which leverages an efficient cryptographic primitive (i.e., Chebyshev chaotic map operation) to generate a shared session key among three parties and achieve security (anonymity and other six properties) and efficiency simultaneously. It then formally proves the security of ALAKAP under the broadly accepted Burrows-Abadi-Needham (BAN) logic model and demonstrates how the proposed protocol satisfies the desired requirements in the fog-assisted H-IoT system. Finally, the performance of ALAKAP is validated by conducting the experiments on Amazon EC2 and Raspberry Pi. The results show that our work can achieve at least 44% higher improvement than the state-of-the-art works.

Index Terms—Healthcare-IoT, Fog Nodes, Key Agreement Protocol, Anonymity, Lightweight.

I. INTRODUCTION

Healthcare Internet of Things (H-IoT)¹ system is increasingly pervasive in our rapidly developing society for bringing convenience to users' daily lives [1], [2]. For example, the IoT equipments connected to the Internet could monitor and collect daily healthcare data from user-side. Meanwhile, these IoT equipments use various sensing technologies to transfer the collected data to a data center (or a cloud server) via wireless gateways located at their home or outside expediently. The doctors offer accurate diagnoses and give the most

comprehensive medical treatments via users' vital signs stored on the cloud server. To some extent, H-IoT system is capable of breaking the obstacles of time and geographical location.

However, cloud computing has its own restricted properties for real-world demands targeting for high mobility and low-latency. These restricted properties can be mitigated in a fog computing architecture [3]. Fog computing² usually deploys fog nodes to the network edge to share the pressure of the remote data centers [4]. In the meantime, the introduction of the fog nodes in the H-IoT system allows some services to be distributed to the fog nodes. As a result, it decreases bandwidth consumption, computational costs, data transmission latency, and so on [5]. The fog-assisted H-IoT system exerts a dramatic impact because the data gathered from users' daily behaviors are processed in real-time or a short-period. It then offloads the data from the remote cloud server for further analysis.

Under this scenario, there are some foundational security risks that urgently require to be tackled [6] and how to deal with these security risks is a challenge. The messages transmitted among the IoT equipments, fog nodes and remote cloud server may leak the users' private information such as users' identities. Once the users' identities are compromised, the user anonymity could not be guaranteed. The other private information can be inferred and compromised such as users' passwords and diseases. Although the users can pre-process their data prior to transmission to improve data confidentiality, it is impractical and romantic to pre-share the session key among the participants. Therefore, authenticated key agreement (AKA) protocol is a significant and profound cryptographic solution, which enables two or more entities to negotiate a shared session key via the public channel. Moreover, in the existing AKA protocols, the participants may include three parties, but the shared session key is negotiated just between two of them with the help of the third party [7].

In the fog computing environment, the fog nodes usually take part in pre-processing the healthcare data from the IoT equipments of users and receive orders from the cloud server and users. Many classical two-party AKA protocols are not adaptive to fog-based environments [3]. For example, Kerberos and some provisions of the ISO/IEC 9798 standard are applicable to the client/server architecture. Although tripartite communication can be safely accomplished by using two-party key agreements twice, the cost is prohibitive. So, three-party

Hui Qiao, Xuewen Dong and Yulong Shen are with the School of Computer Science and Technology, Xidian University, Xi'an Shaanxi 710071, China (e-mail: qiaohui1007300405@163.com; xwdong@xidian.edu.cn; ylsen@mail.xidian.edu.cn).

Qi Jiang and Ning Xi are with the School of Cyber Engineering, Xidian University, Xi'an Shaanxi 710071, China (e-mail: jiangqixdu@gmail.com; nxi@xidian.edu.cn).

Siqi Ma is with the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia (e-mail: slivama@uq.edu.au).

Chao Liu is with the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD 21250, USA (e-mail: chaoliu717@umbc.edu).

Copyright (c) 2023 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

¹The IoT focused on the medical applications is termed as Healthcare Internet of Things (H-IoT).

²Fog computing usually provides storage, applications, and data to end-users and facilitates the operation of compute, storage, and networking services between end devices and cloud computing data centers.

AKA protocol has an absolute advantage in the fog computing scenario. However, the existing three-party AKA works [3] are constructed based on expensive cryptographic primitives (e.g., bilinear pairing), which are barely applied to practice because of the high computation costs. According to the works in [8], [9], the Chebyshev chaotic map is more efficient than the bilinear pairing. Apparently, in the resource-constrained H-IoT environment, lightweight and efficient Chebyshev chaotic map operation is indispensable. Moreover, anonymity, H-IoT users may not want to expose their private identities, is failed to be provided in prior works.

To address the issues and weaknesses mentioned above, our ALAKAP (for *Anonymous and Lightweight three-party Authenticated Key Agreement Protocol*) with lightweight cryptographic primitive (i.e., Chebyshev chaotic map operation) is put forward. Our ALAKAP is constructed for the fog-assisted H-IoT system to enhance its efficiency, security and privacy. Using the lightweight Chebyshev chaotic map operation, a shared session key is generated among three parties (i.e., smart H-IoT equipments, fog nodes and the cloud server). Our ALAKAP achieves the desired security requirements, including mutual authentication, anonymity, perfect forward secrecy, session key agreement, and so on.

To measure the performance of our ALAKAP, the extensive experiments are conducted on Amazon EC2 and single-board computer (i.e., Raspberry Pi 3 Model B). Our experimental results showed that the performance overhead incurred by ALAKAP is significantly reduced than that of previous works [3], [4], [10]. For example, in ALAKAP, the total execution cost is 1.1936ms. Compared to [3], where the total execution time is 66.8518ms, our proposal has reduced the performance overhead by a factor of 55. The decreased overhead is principally attributed to the fact that lightweight Chebyshev chaotic map operation introduced a smaller overhead than expensive cryptographic primitives.

The main contributions of our ALAKAP are listed as follows:

- In combination with Chebyshev polynomial operation, hash function and symmetric encryption, a three-party AKA protocol for fog-assisted H-IoT system called ALAKAP is presented and a shared session key among user, fog node and server is generated in this context.
- ALAKAP is shown to fulfill the desired properties (described in Section III-C) of the fog-assisted H-IoT system. By using Burrows-Abadi-Needham logic (BAN logic) model, the rigorous formal security analysis indicates that the ALAKAP protocol achieves mutual authentication.
- ALAKAP's performance is evaluated on Amazon EC2 and single-board computer (i.e., Raspberry Pi 3 Model B), respectively. The performance analysis reveals that ALAKAP can achieve at least 44% higher improvement than the state-of-the-art works [3], [4], [10].

The remainder of this paper is organized as follows. In Section II, a brief review of the existing related work is taken. Next, the system model, threat model and desired properties are presented in Section III and building blocks in Section IV. Section V gives a description of ALAKAP in detail, whereas Section VI bewrites the security analysis and formal

security proof employing BAN logic model. The performance of ALAKAP is evaluated in Section VII. Concluding remarks of this paper are found in the last Section.

II. RELATED WORK

There are a large body of works on key agreement design of the medical areas in the past decade [11]–[33]. The work in [20] proposed a secure authentication protocol for telecare medicine information systems. In the same year, Wei *et al.* [21] testified that He *et al.*'s protocol [20] is incapable of resisting off-line password guessing attack and proposed an improved authentication scheme for telecare medicine information systems. However, their protocol is still incapable of withstanding password guessing attack [22]. Jiang *et al.* [23] proposed a privacy enhanced authentication protocol which achieves user anonymity and untraceability, but Kumari *et al.* [24] found their protocol has confronted with stolen verifier attack, online password guessing attack and impersonation attack. According to the above reasons, the work in [25] put forward a privacy authentication protocol on cloud computing to provide medical services. The authentication protocol used mobile device's characteristics, allowing people to use medical resources in the cloud environment to find medical advice conveniently. They declared their proposed protocol was secure against many ordinary and common attacks. Chiou *et al.* [26] testified that the protocol in [25] not only had a high degree of computational complexity but also failed to provide patient anonymity and message authentication. In addition, in their scheme, patients have to visit the hospital in person to let doctors obtain medical data, which means that Chen *et al.*'s protocol [25] is incapable of achieving real telemedicine. Later, the work in [27] claimed that the protocol presented in [26] still failed to ensure message confidentiality and patient anonymity. Li *et al.* [28] pointed out that the protocol [27] failed to protect patient anonymity and unlinkability. Liu and Ma [29] found that each valid patient had ability of acquiring the cloud server's private key easily in Mohit *et al.*'s protocol [27]. It is indisputable that known private key brings the system security risk. Continuously, a number of more advanced AKA protocols [30], [31] have been presented for many different purposes, such as protocol with user anonymity [32]. However, these key agreement protocols are only designed in the cloud computing environment, which is not suitable for the latency/delay sensitive applications and geographically distributed applications [3], while ignoring the low latency and high mobility.

Furthermore, fog computing has been extensively deployed in recent years. The work in [33], focusing on protecting the privacy of sensitive healthcare data using a fog computing facility, put forward a tri-party one-round AKA protocol based on the bilinear pairing cryptography that can generate a session key among the participants and communicate among them securely. Finally, the private healthcare data are accessed and stored securely by implementing a decoy technique. Jia *et al.* [3] certified that the session key produced by Hamid *et al.*'s protocol [33] was static and had no ability to provide forward privacy. Besides, Hamid *et al.*'s protocol also suffered from man-in-the-middle attacks launched by an active adversary

since their key exchange mechanism was based on the tripartite Diffie-Hellman key exchange algorithm. Jia *et al.* then came up with a three-party AKA protocol employing bilinear pairings for the IoT health system under fog computing facility. Chen *et al.* [34] showed that the work in [3] is vulnerable to ephemeral secret leakage attacks and designed a secure AKA protocol based on fog computing. In 2021, Shamshad *et al.* [35] reported that the work in [3] is incapable of resisting impersonation attacks and cannot provide anonymity for users and fog nodes. Wu *et al.* [36] also found that Jia *et al.*'s protocol [3] exhibits security vulnerabilities, such as known session specific temporary information attacks and a lack of pre-verification. Ma *et al.* [4] pointed out that the execution of a bilinear pairing was expensive in many basic operations. A novel AKA protocol without the use of bilinear pairing was proposed to improve the efficiency. However, the efficiencies of the work in [3] using bilinear pairing and the works in [4], [10] using Elliptic Curve Cryptography (ECC) scalar multiplication was still to be improved because the Chebyshev chaotic map operation utilized in this paper is more efficient than bilinear pairing and ECC scalar multiplication in terms of computation according to the works in [8], [9], which we also display in Table V of Section VII.

III. MODEL AND DESIRED PROPERTIES

A. System Model

Unlike a centralized cloud-based system, in our system model, a fog-assisted healthcare IoT (H-IoT) system is depicted minutely (as shown in Fig. 1), in which four components (Users U , Fog Nodes FN , Cloud Server S and Doctors D) are involved. Since ALAKAP protocol is done among users U , fog nodes FN and cloud server S , we mainly introduce three components inside the red dotted box. In Fig. 1, the dotted arrows indicate a secure channel and the solid arrows indicate a public channel. The concrete descriptions of them are as follows:

- **Users:** These entities are the main participants of an H-IoT system, who own their respective identities. They regulate one or more smart IoT equipments which are joined to the fog nodes and put in their personal medical health data to a fog node through wireless terminals at their home or outside expediently. U_i represents the i th user and must input the correct password and insert the matched smart card to pass the system authentication.
- **Fog Nodes:** These entities have certain computing and storage capacity, who also own their respective identities. Each fog node, e.g., the j th fog node FN_j , processes, delivers, and stores the received authentication messages and the other commands between the user U_i and the cloud server S . The introduction of the fog nodes in the H-IoT system is capable of providing real-time responses on time-sensitive tasks. Fog nodes also send comprehensive data to the cloud server for long-term storage. And then the doctors can conduct further analysis.
- **Cloud Server:** Cloud server, which possesses the high-performance and significantly powerful storage capacity, provides the corresponding registration service for users,

fog nodes and doctors according to their registration requests. This paper assumes that the cloud server is a trusted and trustworthy component [4].

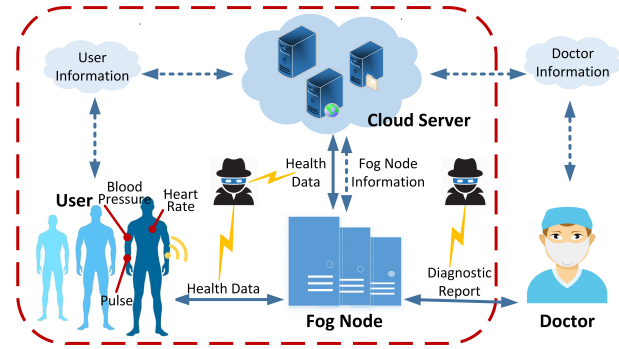


Fig. 1. The system model of fog-assisted H-IoT system

B. Threat Model

Generally speaking, the adversary possesses several advantages or capabilities because the authentication protocol is completed in the public channel. The AKA protocol in the fog-assisted H-IoT system should accord with the following assumptions [32]:

- Using reverse engineering techniques, an adversary *Malice* is capable of detecting energy consumption to extract any user's personal information saved in the smart card [37], [38]. Furthermore, the adversary *Malice* listens in on all messages transmitted in the public channel. Concurrently, the malicious adversary *Malice* has no ability to intercept messages in the secure channel.
- Similar to [8], the adversary *Malice* can guess unique identity ID and low entropy password PW of the enrolled users and fog nodes individually easily, but, in polynomial time, guessing two secret arguments (e.g., ID , PW) is computationally infeasible.
- An adversary *Malice* can modify, delete, resend and reroute the messages derived from users, fog nodes, and cloud server in the fog-assisted H-IoT system.
- An adversary *Malice* may be a malicious but legitimate user or fog node in the fog-assisted H-IoT system.

C. Desired Properties

On the ground of the existing literature, such as those of [4], [39], a three-party AKA protocol for fog-assisted H-IoT system needs to fulfill the following fundamental security properties.

- **Mutual authentication.** The proposal should have capability of ensuring mutual authentication among all participants (i.e., user, fog node, and cloud server) to strive against common and usual attacks such as impersonation attacks.
- **User anonymity.** The proposal should have ability of providing the anonymity of users. From the view of the fog-assisted H-IoT system, users' real identities cannot

be determined even though an adversary intercepts the messages in the transmission process.

- **Session key agreement.** The proposal should support session key agreement or establishment to communicate with parties in the fog-assisted H-IoT system securely. That is to say, one can establish a secure connection with other parties to negotiate a shared session key for succeeding communications.
- **Perfect forward secrecy.** In the proposal, to pledge the security of messages exchanged in the fog-assisted H-IoT system before, perfect forward secrecy should be guaranteed where any adversary who acquires both participants' private/public key pair will not be capable of recovering prior session keys.
- **Resistance of known session key attack.** In some circumstances, the adversary can not compute another secure session key, even if he/she knows the session key generated in a certain protocol.
- **Resistance of password guessing attack.** The proposal in the fog-assisted H-IoT system has ability of withstanding the password guessing attack, even if the attacker uses the information that he/she has gained, including the data of smart cards, the messages of transmission, and the information of the cloud server.
- **Resistance of replay attack.** An attacker can not replay the old messages to attack the fog-assisted H-IoT system because of the timestamp's freshness.

IV. BUILDING BLOCKS

Before working with secure authentication protocol specifics, we review in detail the building blocks: Chebyshev polynomial and intractability problems. Due to the limitation of length, hash function and symmetric encryption refer to [40].

A. Chebyshev Polynomial

At the first, a review of several basic concepts about Chebyshev polynomial are given concisely. More details can be found in [41], [42].

Definition 1. Chebyshev Polynomial:

Let n be an integer, and x as a variable taking value with the interval $x \in [-1, 1]$, then the Chebyshev polynomial $T_n(x)$: $[-1, 1] \rightarrow [-1, 1]$ above is written as below:

$$T_n(x) = \cos(n \arccos(x)) \quad (1)$$

A recurrent relation can be utilized for defining Chebyshev polynomial map $T_n : R \rightarrow R$ of degree n , by specifying the following equation.

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \quad (2)$$

Given $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

The first few Chebyshev polynomials are presented as follows:

$$T_2(x) = 2x^2 - 1 \quad (3)$$

$$T_3(x) = 4x^3 - 3x \quad (4)$$

$$T_4(x) = 8x^4 - 8x^2 + 1 \quad (5)$$

Definition 2. Semi-group Feature:

The semi-group feature of Chebyshev polynomial can be written on an interval $(-\infty, +\infty)$ as below:

$$T_r(T_l(x)) = T_{rl}(x) = T_{lr}(x) = T_l(T_r(x)) \quad (6)$$

The work in [41] states that the semi-group feature above-mentioned is also appropriate for the Chebyshev polynomial with the interval $(-\infty, +\infty)$.

Definition 3. Extended Chebyshev Polynomial:

Presume n as an integer, and x as a variable over the interval $(-\infty, +\infty)$, then the extended Chebyshev polynomial is written as below:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \pmod{p} \quad (7)$$

Given $n \geq 2$, $T_0(x) = 1$, $T_1(x) = x$, and p is a big prime. Distinctly, the semi-group feature is also appropriate for the extended Chebyshev polynomial.

$$T_r(T_l(x)) = T_{rl}(x) = T_{lr}(x) = T_l(T_r(x)) \pmod{p} \quad (8)$$

B. Intractability Problems

Definition 4. Chaotic Map Discrete Logarithm Problem (CMDLP): Given $\langle x, T_u(x) \rangle$, it is computationally infeasible to earn a proper integer u such that $T_u(x) = y$.

Definition 5. Chaotic Map Computational Diffie-Hellman Problem (CMCDHP): Given $\langle x, T_u(x), T_v(x) \rangle$, it is computationally infeasible to calculate $T_{uv}(x) = y$.

V. OUR PROPOSAL ALAKAP PROTOCOL

In this section, ALAKAP which combines the extended Chebyshev chaotic map operation, hash function, and symmetric encryption to deal with issues and drawbacks mentioned in Section I is detailedly depicted. Our new proposal contains six phases, i.e., user registration phase, fog node registration phase, authentication and key agreement phase, password update phase, user revocation and re-registration phase, and fog node revocation phase. All the phases of ALAKAP are detailed as follows:

Several of the employed notations are presented in Table I.

TABLE I
NOTATION DESCRIPTION

Symbol	Description
U_i	User U_i
FN_j	Fog node FN_j
S	Cloud server S
ID_i, ID_j	The identities of the user U_i and fog node FN_j
PW_i	The password of the user U_i
SC	Smart Card
s	The master key of cloud server S
$E_s(\cdot)$	Symmetric key encryption algorithm using s
$D_s(\cdot)$	Symmetric key decryption algorithm using s
$h(\cdot)$	Secure one-way hash function
\oplus	Exclusive-or operation
\parallel	String concatenation operation
$T_u(x)$	Chebyshev chaotic map operation

A. User Registration Phase

This phase is executed between the user U_i and the cloud server S via a secure and secret channel when the user U_i wants to be registered as a legitimate user and gains the cloud server S 's service. As shown in Fig. 2, it implements the following steps for registration.

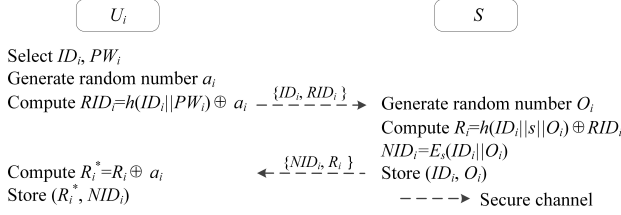


Fig. 2. User registration phase

Step UR1: User U_i first chooses its identity ID_i and secret password PW_i and generates a random number a_i . Then, the user U_i computes $RID_i = h(ID_i || PW_i) \oplus a_i$ and sends a registration request message (ID_i, RID_i) to the cloud server S .

Step UR2: After that, the cloud server S generates a random number O_i , computes $R_i = h(ID_i || s || O_i) \oplus RID_i$ and $NID_i = E_s(ID_i || O_i)$, and stores the message $\{R_i, NID_i\}$ in the smart card SC . Finally, the smart card SC is sent to user U_i by the cloud server S . The cloud server S also records (ID_i, O_i) into its own database.

Step UR3: Once the smart card SC is received, the user U_i calculates $R_i^* = R_i \oplus a_i$ and takes place of the parameter R_i on the smart card SC with R_i^* .

B. Fog Node Registration Phase

This phase is executed between the fog node FN_j and the cloud server S via a secure and secret channel when the fog node FN_j wants to deliver the authentication messages between the user U_i and the cloud server S . As shown in Fig. 3, it completes the following steps for registration.

Step FR1: Fog node FN_j first chooses its unique identity ID_j and sends its unique identity to the cloud server S .

Step FR2: After that, the cloud server S generates a random number N_j , computes $R_j = h(ID_j || s || N_j)$ and $NID_j = E_s(ID_j || N_j)$, and sends the message $\{R_j, NID_j\}$ to the fog node FN_j . The cloud server S also records (ID_j, N_j) into its own database.

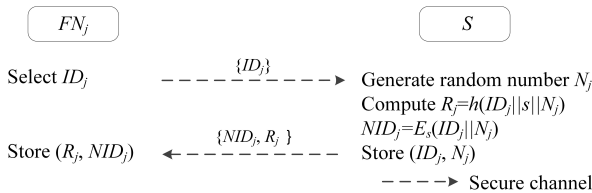


Fig. 3. Fog node registration phase

C. Authentication and Key Agreement Phase

This phase is executed among user U_i , fog node FN_j , and cloud server S to realize mutual authentication and generate

a shared session key for the subsequent communication. As shown in Fig. 4, it performs the following steps for authentication and key agreement.

Step A1: The user U_i computes $T_u(x)$ after choosing a random number u and calculates $PID_i = h(ID_i || PW_i || NID_i)$, $M_i = h(ID_i || PW_i) \oplus R_i^*$, $W_i = h(ID_i || T_u(x) || M_i || NID_i || T_1)$, where T_1 is the current timestamp. Finally, the user U_i sends message $MS_1 = \{NID_i, PID_i, W_i, T_u(x), T_1\}$ to the fog node FN_j .

Step A2: After that, the fog node FN_j first checks the freshness of the timestamp T_1 , generates a random number v , then calculates $T_v(x)$, $A = T_v(T_u(x))$ and $W_j = h(PID_i || ID_j || T_u(x) || A || R_j || NID_j || T_2)$, where T_2 is the current timestamp. Finally, the fog node FN_j sends message $MS_2 = \{NID_i, NID_j, PID_i, A, W_i, W_j, T_u(x), T_v(x), T_1, T_2\}$ to the cloud server S .

Step A3: Upon receiving the authentication message MS_2 sent by the fog node FN_j , the cloud server S first checks the freshness of two timestamps T_1 and T_2 . If the timestamps T_1 and T_2 are fresh, the cloud server S executes following operations.

- 1) The cloud server S decrypts the parameter NID_i and NID_j to retrieve the user U_i 's identity $ID_i' || O_i = D_s(NID_i)$ and the fog node FN_j 's identity $ID_j' || N_j = D_s(NID_j)$.
- 2) The cloud server S queries its database to find (ID_i', O_i) and (ID_j', N_j) . If (ID_i', O_i) and (ID_j', N_j) are not found, then the cloud server S stops the session.
- 3) The cloud server S continues to compute $M_i' = h(ID_i' || s || O_i)$, $R_j' = h(ID_j' || s || N_j)$, $W_i' = h(ID_i' || T_u(x) || M_i' || NID_i || T_1)$ and $W_j' = h(PID_i || ID_j' || T_u(x) || A || R_j' || NID_j || T_2)$.
- 4) The cloud server S verifies whether $W_i \stackrel{?}{=} W_i'$ and $W_j \stackrel{?}{=} W_j'$ hold or not. If either one does not hold, then the cloud server S rejects the session.
- 5) The cloud server S generates three random numbers b_i , b_j and w , computes $HID_i = h(ID_i' || b_i)$, $HID_j = h(ID_j' || b_j)$, $V_i = h(M_i' || ID_i') \oplus HID_i$, $V_j = h(R_j' || ID_j') \oplus HID_j$, $B = T_w(T_u(x))$, $C = T_w(T_v(x))$, $sk_w = h(A || B || C || T_w(A))$, $Au_i = h(ID_i' || NID_i || HID_i || C || sk_w || T_3)$ and $Au_j = h(PID_i || ID_j' || NID_j || HID_j || B || sk_w || T_3)$, where T_3 is the current timestamp.
- 6) Finally, the cloud server S sends the message $MS_3 = \{V_i, V_j, Au_i, Au_j, B, C, T_3\}$ to the fog node FN_j .

Step A4: After receiving the message MS_3 , the fog node FN_j first examines the validity of the timestamp T_3 , then computes $HID_j' = V_j \oplus h(R_j || ID_j)$, $sk_v = h(A || B || C || T_v(B))$, and $Au_j' = h(PID_i || ID_j || NID_j || HID_j' || B || sk_v || T_3)$. After that, the fog node FN_j verifies whether $Au_j \stackrel{?}{=} Au_j'$ holds or not. If it is unequal, the fog node FN_j stops the session. Otherwise, the fog node FN_j sends the message $MS_4 = \{V_i, Au_i, A, B, C, T_3\}$ to the user U_i .

Step A5: Upon receiving the message MS_4 sent by the fog node FN_j , the user U_i first examines the freshness of the timestamp T_3 , then computes $sk_u = h(A || B || C || T_u(C))$, $HID_i' = V_i \oplus h(M_i || ID_i)$, $Au_i' = h(ID_i || NID_i || HID_i' || C || sk_u || T_3)$ and verifies whether $Au_i \stackrel{?}{=} Au_i'$ holds or not. If it is not

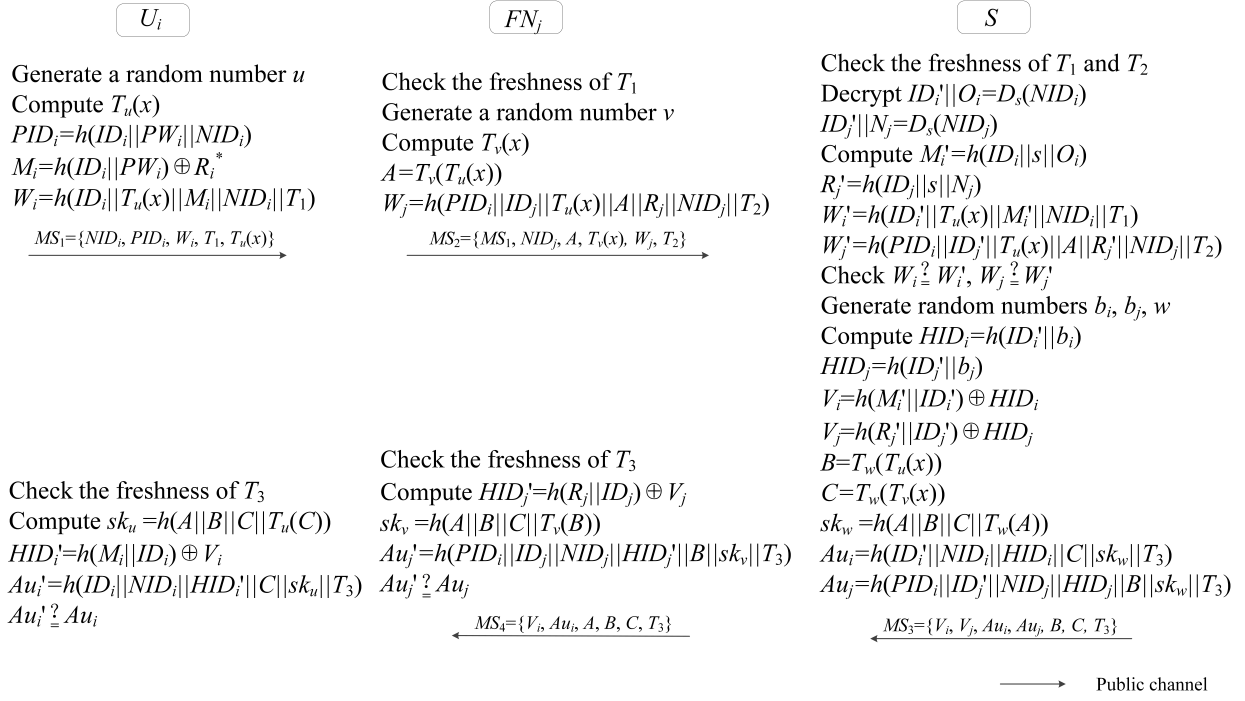


Fig. 4. Key agreement phase

equal, the user U_i stops the session. Otherwise, because of the semi-group feature, $sk = sk_u = sk_v = sk_w$ is the shared session key among user U_i , fog node FN_j and cloud server S .

D. Password Update Phase

In this phase, the following procedures will be executed when the user U_i cares to change his/her password.

Step PU1: The user U_i inserts the smart card SC into the card reader and issues an instruction to update the password.

Step PU2: The user U_i types in its identity ID_i , secret password PW_i and the new password PW_i^{new} .

Step PU3: The smart card SC calculates $R_i^{*new} = h(ID_i || PW_i) \oplus h(ID_i || PW_i^{new}) \oplus R_i^*$ and replaces the old parameter R_i^* with R_i^{*new} .

E. User Revocation and Re-registration Phase

In some situation, users have to revoke their account information from the H-IoT system. For example, if a thief steals a user U_i 's smart card SC or a user U_i 's smart card is lost, then the user U_i should revoke the old account information in such an event. The user U_i sends a revocation request message to the cloud server S . After verifying the user U_i 's identity ID_i (based on the old password or other identity information), the cloud server S deletes the entry (ID_i, O_i) from its database and thereafter the login request message issued by the old smart card will be terminated. The user U_i can re-register with the cloud server S using the same identity and a new password following the registration procedure described in Section V-A. The cloud server S then generates a new random number O_i^{new} and records (ID_i, O_i^{new}) in its database.

F. Fog Node Revocation Phase

If an adversary *Malice* has ability of controlling the fog node FN_j , that is to say, the fog node FN_j is damaged or compromised, then the cloud server S will revoke the node by deleting the record (ID_j, N_j) from its database. Whereafter, any access to the fog node will be rejected, since all authentication request messages issued by FN_j can not be successfully authenticated without the random number N_j .

VI. THEORETICAL ANALYSIS

This section makes a detailed description of the correctness analysis and the security analysis of ALAKAP. The security analysis includes the informal security analysis proving that ALAKAP can satisfy all the security properties mentioned in Section III-C and formal security proof under Burrows-Abadi-Needham logic (BAN logic) model.

A. Correctness Analysis

This subsection displays the correctness analysis of ALAKAP.

$$\begin{aligned} T_w(A) &= T_w(T_v(T_u(x))) \\ &= T_v(T_w(T_u(x))) \\ &= T_v(B) \end{aligned} \quad (9)$$

$$\begin{aligned} T_w(A) &= T_w(T_v(T_u(x))) \\ &= T_w(T_u(T_v(x))) \\ &= T_u(T_w(T_v(x))) \\ &= T_u(C) \end{aligned} \quad (10)$$

In Step A3 of the authentication and key agreement phase, the cloud server S computes the session key as $sk_w = h(A \parallel B \parallel C \parallel T_w(A))$. Similarly, as defined in Step A4 and Step A5 of the authentication and key agreement phase, the fog node FN_j calculates the session key as $sk_v = h(A \parallel B \parallel C \parallel T_v(B))$ and the user U_i calculates the session key as $sk_u = h(A \parallel B \parallel C \parallel T_u(C))$. Due to the semi-group feature of Chebyshev chaotic maps defined in Section IV-A, we have $T_w(A) = T_v(B) = T_u(C)$. Therefore, the user U_i , the fog node FN_j and the cloud server S hold the same session key sk .

B. Informal Security Analysis

ALAKAP has capability of resisting a wide range of security attacks such as password guessing attacks and a succession of security features such as user anonymity. However, the works in [3], [4] are incapable of providing user anonymity. In [4], Ma *et al.* assume that a cloud server is a trusted cloud service provider. For comparison purposes under the same assumptions, we also assume that the cloud server in this paper is a trusted and trustworthy component, which we have described in Section III-A and cited the corresponding reference [4]. So, insider attack will not happen in the proposed protocol. In a word, the differences in the field of security properties between our ALAKAP and the existing protocols [3], [4], [10] have been presented in Table II. Evidently, ALAKAP has higher security in comparison with the other relevant protocols. The security properties of ALAKAP are specified as follows:

TABLE II
PROPERTIES COMPARISONS

Properties	[3]	[4]	[10]	Ours
Fog computing	✓	✓	✗	✓
H-IoT	✓	✗	✗	✓
Lightweight	✗	✗	✗	✓
Replay attack	✓	✓	✓	✓
Password guessing attack	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓
User anonymity	✗	✗	✓	✓
Session key agreement	✓	✓	✓	✓
Perfect forward secrecy	✓	✓	✓	✓
Known key session attack	✓	✓	✓	✓

• Mutual authentication.

Upon receiving the messages MS_1 and MS_2 from the user U_i and fog node FN_j , the cloud server S checks whether $W_i \stackrel{?}{=} h(ID_i' \parallel T_u(x) \parallel M_i' \parallel NID_i \parallel T_1)$ and $W_j \stackrel{?}{=} h(PID_i \parallel ID_j' \parallel T_u(x) \parallel A \parallel R_j' \parallel NID_j \parallel T_2)$ hold or not separately. If it is unequal, the cloud server S stops the session. Otherwise, the cloud server S believes that the user U_i is a legal user and the fog node FN_j is a legal fog node. In addition, upon receiving the messages MS_3 and MS_4 from the cloud server S , the user U_i verifies whether $Au_i' = h(ID_i \parallel NID_i \parallel HID_i' \parallel C \parallel sk_u \parallel T_3)$ and $Au_j' = h(PID_i \parallel ID_j \parallel NID_j \parallel HID_j' \parallel B \parallel sk_v \parallel T_3)$ hold or not. If equal, the user U_i and the fog node FN_j believe that the cloud server S is a legal server. In ALAKAP, the cloud server S , the fog node FN_j and the user U_i can be authenticated by checking W_i , W_j , Au_i and Au_j . There exists no probabilistic polynomial time adversary

to successfully forge the available W_i , W_j , Au_i and Au_j because solving CMCDHP problem is computationally hard. Visibly, our ALAKAP could achieve mutual authentication.

• User anonymity.

In ALAKAP, the real identity ID_i of user U_i is not included in the messages transmitted on the public channel. For the user U_i , because $NID_i = E_s(ID_i \parallel O_i)$ contains the random number O_i and $PID_i = h(ID_i \parallel PW_i \parallel NID_i)$ is protected by the hash function, the adversary *Malice* cannot obtain the real identity of the user U_i . Hence, our ALAKAP could guarantee user privacy without being compromised.

• Session key agreement.

In accordance with ALAKAP which is displayed in the Section V, the cloud server S computes the session key as $sk_w = h(A \parallel B \parallel C \parallel T_w(A))$, the fog node FN_j calculates the session key as $sk_v = h(A \parallel B \parallel C \parallel T_v(B))$ and the user U_i calculates the session key as $sk_u = h(A \parallel B \parallel C \parallel T_u(C))$ in the authentication and key agreement phase, separately. Due to the semi-group feature of Chebyshev chaotic maps defined in Section IV-A, we have $T_w(A) = T_v(B) = T_u(C)$. The user U_i , the fog node FN_j and the cloud server S hold the same session key sk . Therefore, ALAKAP could finish session key agreement.

• Perfect forward secrecy.

An authentication protocol is capable of providing perfect forward secrecy if the prior session keys fail to be recovered even both the user's and the server's secret keys are compromised [32]. In ALAKAP, a compromised password PW_i does not yield any prior session key $sk = h(A \parallel B \parallel C \parallel T_w(A))$ because the parameters u , v , w and x are casually chosen and independent among protocol executions. In addition, without the knowledge of the high entropy random integer u , v , w and x , the attacker *Malice* is incapable of figuring out the prior session key. So, our ALAKAP could achieve perfect forward secrecy.

• Resistance of known session key attack.

An authentication protocol could withstand known session key attack if, in some circumstances, its execution could generate a session key and the compromise of this key has no influence on other session keys. In ALAKAP, the shared session key $sk = h(A \parallel B \parallel C \parallel T_w(A))$ produced in different runs are independent of each other because the parameters u , v , w and x are chosen randomly and independently by the user U_i , the cloud server S and the fog node FN_j respectively, and are independent of each other among protocol executions. Consequently, ALAKAP could resist known session key attack.

• Resistance of password guessing attack.

An authentication protocol has ability of withstanding password guessing attack, although the attacker *Malice* uses the information that he/she has gained, including the data of smart card, the messages of transmission, and the information of the cloud server. The attacker *Malice* could intercept the messages $MS_1 = \{NID_i, PID_i, W_i, T_u(x), T_1\}$ relating to the user U_i , where $NID_i = E_s(ID_i \parallel O_i)$, $PID_i = h(ID_i \parallel PW_i \parallel NID_i)$ and $W_i = h(ID_i \parallel T_u(x) \parallel$

$M_i \parallel NID_i \parallel T_1$). He/She could also get $\{NID_i, R_i^*\}$ from the user U_i 's smart card through side channel attack. It is notable that an attacker *Malice* has ability of guessing low entropy password and finding the unique identity of the user individually easily but guessing two secret arguments (e.g., password, identity) is computationally infeasible in polynomial time. So, *Malice* cannot extract ID_i, PW_i from NID_i, PID_i, W_i and R_i^* . Therefore, our ALAKAP could resist the password guessing attack.

• Resistance of replay attack.

An attacker *Malice* can not replay the old messages to attack the fog-assisted H-IoT system successfully. In the replay attack, the adversary *Malice* can complete the purpose of identity authentication and key agreement by replaying the previous messages sent by the users. In each session of our ALAKAP, the user U_i , the fog node FN_j and cloud server S generate random numbers u, v and w respectively to keep the session fresh. When an attacker replays messages MS_1, MS_2, MS_3 and MS_4 sent by the user U_i , the fog node FN_j and cloud server S and attempts to calculate the session key using intercepted messages, he/she will face the CMCDHP problem. At the same time, three timestamps T_1, T_2 and T_3 are included in the messages MS_1, MS_2, MS_3 and MS_4 . Therefore, our ALAKAP could resist the replay attack due to the freshness of timestamp and CMCDHP problem.

C. Formal Security Analysis (BAN Logic)

This subsection incorporates the formal security proof of ALAKAP employing Burrows-Abadi-Needham logic (BAN logic) model. Several of the symbols utilized in BAN logic are depicted in Table III.

TABLE III
NOTATION DESCRIPTION

Symbol	Description
$P \models X$	The principal P believes X , or alternatively, P believes the statement X .
$P \triangleleft X$	P sees X , P receives some message X and may read or repeat it in any message.
$P \sim X$	P once said X , P had sent some message X and P believed that message when sent.
$\#(X)$	The message X may be treated as fresh.
$P \Rightarrow X$	P has got jurisdiction over X , or P has authority over X and could be trusted.
$P \xrightarrow{sk} Q$	P and Q can communicate with the shared session key sk .
$\langle X \rangle_Y$	The formulate X is combined with the formulate Y .
$\{X\}_{sk}$	X is encrypted with the key sk .

As such, several logical assumptions or rules are displayed as follows:

Rule 1. Message meaning rule: $\frac{P \models Q \xrightarrow{sk} P, P \triangleleft \{X\}_{sk}}{P \models Q \models X}$

Rule 2. Nonce verification rule: $\frac{P \models \#X, P \models Q \models X}{P \models Q \models X}$

Rule 3. Seeing rule: $\frac{P \models P \xrightarrow{sk} Q, P \triangleleft \{X\}_{sk}}{P \models \#X}$

Rule 4. Freshness rule: $\frac{P \models \#X}{P \models \#XY}$

Rule 5. Session key rule: $\frac{P \models \#X, P \models Q \models X}{P \models P \xrightarrow{sk} Q}$

Rule 6. Jurisdiction rule: $\frac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X}$

It is essential and necessary for ALAKAP to meet the goals below to guarantee its security under BAN logic model, taking full advantage of the postulates and rules mentioned above.

Goal 1: $U_i \models S \models U_i \xrightarrow{sk} S$

Goal 2: $U_i \models U_i \xrightarrow{sk} S$

Goal 3: $S \models U_i \models U_i \xrightarrow{sk} S$

Goal 4: $S \models U_i \xrightarrow{sk} S$

Goal 5: $S \models U_i \sim \#(T_u(x))$

Goal 6: $S \models FN_j \sim \#(T_v(x))$

Firstly, the messages commuted in ALAKAP can be transformed into idealized form in the undermentioned manner.

MS1: $U_i \rightarrow FN_j : \{NID_i, PID_i, W_i, T_u(x), T_1\}$

MS2: $FN_j \rightarrow S : \{NID_i, NID_j, PID_i, W_i, W_j, T_u(x), A, T_v(x), T_1, T_2\}$

MS3: $S \rightarrow FN_j : \{V_i, V_j, Au_i, Au_j, B, C, T_3\}$

MS4: $FN_j \rightarrow U_i : \{V_i, Au_i, A, B, C, T_3\}$

Next, the undermentioned assumptions have been constituted to give evidence of the security of ALAKAP.

AS1: $U_i \models S \xrightarrow{HID_i} U_i$

AS2: $S \models \#(b_i, T_3)$

AS3: $U_i \models S \models \#(w, T_3)$

AS4: $S \models U_i \xrightarrow{M_i} S$

AS5: $U_i \models \#(u, T_1)$

AS6: $S \models U_i \models \#(u, T_1)$

AS7: $U_i \xrightarrow{sk} S$

AS8: $U_i \models \#T_u(x)$

AS9: $FN_j \models \#(v, T_2)$

AS10: $FN_j \models \#T_v(x)$

On the ground of BAN logic rules and the postulates, a precise and detailed proof of the idealized form of ALAKAP has been done. The major proofs are stated as follows:

Proof 1 Because there exists a seeing rule $U_i \triangleleft \{h(M_i \parallel ID_i) \oplus HID_i, Au_i, A, B, C, T_3\}$, by applying message meaning rule and AS1 $U_i \models (U_i \xrightarrow{HID_i} S)$, it obtains the expression $U_i \models S \sim h(M_i \parallel ID_i) \oplus HID_i$. On the basis of the freshness rule and the assumption AS2, it could gain $U_i \models \#Au_i$. In accordance with the nonce verifications rule, it gets the expression $U_i \models S \models Au_i$. Based on the AS3 and jurisdiction rule, it gains $U_i \models Au_i$. According to the AS2 $U_i \models \#(b_i, T_3)$ and session key rule, it obtains the expression $U_i \models S \models U_i \xrightarrow{sk} S$ (**Goal 1**). In the light of the AS2 and jurisdiction rule, it gets the expression $U_i \models U_i \xrightarrow{sk} S$ (**Goal 2**). Eventually, it certifies that the user U_i believes that sk is the shared session key between the user U_i and the cloud server S .

Proof 2 Because there exists a seeing rule $S \triangleleft \{NID_i, h(ID_i \parallel T_u(x) \parallel M_i \parallel NID_i \parallel T_1), PID_i, T_u(x), T_1\}$, by applying message meaning rule and AS4 $S \models (U_i \xrightarrow{M_i} S)$, it gains the expression $S \models U_i \sim h(ID_i \parallel T_u(x) \parallel M_i \parallel NID_i \parallel T_1)$. On the basis of the freshness rule and the assumption AS5 $S \models \#(u, T_1)$, it could gain $S \models \#T_u(x)$. In accordance with the nonce verifications rule, it knows the expression $S \models U_i \models T_u(x)$. According to the AS6 and jurisdiction rule, it gains $S \models T_u(x)$. According to the AS5 and session key rule, it obtains the expression $S \models U_i \models U_i \xrightarrow{sk} S$ (**Goal 3**). According to the AS5 and jurisdiction rule, it knows $S \models U_i \xrightarrow{sk} S$

(Goal 4). Lastly, it testifies that the user U_i believes that sk is the shared session key between the user U_i and the cloud server S .

Proof 3 It has to validate that the user U_i has sent the fresh message to the cloud server S and testify that the user U_i is the legal user, that is to say, $S \models U_i \sim T_u(x)$. On the basis of the assumption AS7, it realizes that the cloud server S and the user U_i share the session key sk , and the server S has received the message W_i sent by the user U_i , that is to say, $\frac{S \models U_i \xleftrightarrow{sk} S, S \models W_i}{S \models U_i \sim W_i}$. In accordance with the message meaning rule, it realizes that the message exchanged between the cloud server S and the user U_i contains $T_u(x)$, and then, it obtains the expression $\frac{S \models U_i \xleftrightarrow{sk} S, S \models h(ID_i || T_u(x) || M_i || NID_i || T_1)}{S \models U_i \sim h(ID_i || T_u(x) || M_i || NID_i || T_1)}$. So, the cloud server S believes that the user U_i has sent the message comprised of $T_u(x)$. According to the assumptions AS5, AS6 and AS8, it gains $S \models U_i \sim \#T_u(x)$ (**Goal 5**). Eventually, it certifies that the user S believes the user U_i has sent the fresh message included $T_u(x)$, and validates that the user U_i is the legal user.

Proof 4 It has to validate that the fog node FN_j has sent the fresh message to the cloud server S and testify that the fog node FN_j is the legal node, that is to say, $S \models FN_j \sim T_v(x)$. In accordance with the assumption AS7, it realizes that the cloud server S and the fog node FN_j share the session key sk , and the server S has received the message W_j sent by the fog node FN_j , that is to say, $\frac{S \models FN_j \xleftrightarrow{sk} S, S \models W_j}{S \models FN_j \sim W_j}$. On the basis of the message meaning rule, it learns that the message exchanged between the cloud server S and the fog node FN_j involves $T_v(x)$, then, it obtains the expression $\frac{S \models FN_j \xleftrightarrow{sk} S, S \models h(PID_j || ID_j || T_u(x) || A || R_j || NID_j || T_2)}{S \models U_i \sim h(PID_j || ID_j || T_u(x) || A || R_j || NID_j || T_2)}$. So, the cloud server S believes that the fog node FN_j has sent the message comprised of $T_v(x)$. According to the assumptions AS9 and AS10, it gains $S \models FN_j \sim \#T_v(x)$ (**Goal 6**). Lastly, it demonstrates that the user S believes the fog node FN_j has sent the fresh message comprised of $T_v(x)$, and certifies that the fog node FN_j is the legal node.

By implementing the above-mentioned logic deduction process, the anticipated goal has been accomplished.

$U_i \models S \models U_i \xleftrightarrow{sk} S$, that is to say, the user U_i believes that S believes that sk is the shared session key between the user U_i and the cloud server S .

$U_i \models U_i \xleftrightarrow{sk} S$, that is to say, the user U_i believes that sk is the shared session key between the user U_i and the cloud server S .

$S \models U_i \models U_i \xleftrightarrow{sk} S$, that is to say, the cloud server S believes that U_i believes that sk is the shared session key between the cloud server S and the user U_i .

$S \models U_i \xleftrightarrow{sk} S$, that is to say, the cloud server S believes that sk is the shared session key between the cloud server S and the user U_i .

$S \models U_i \sim \#(T_u(x))$, that is to say, the cloud server S believes the user U_i has sent the fresh message comprised of $T_u(x)$.

$S \models FN_j \sim \#(T_v(x))$, that is to say, the cloud server S believes the fog node FN_j has sent the fresh message comprised of $T_v(x)$.

The aforementioned BAN logic proof certifies that ALAKAP fulfills mutual authentication. Besides, the session

key sk is created in consultation with the user U_i , the fog node FN_j and the cloud server S .

VII. PERFORMANCE EVALUATION

This section displays the performance of the proposed ALAKAP (which is concretely described in Section V) in the field of computation costs and communication costs.

TABLE IV
TEST PLATFORM

Devices	Raspberry Pi	Amazon EC2
Operating System	Raspberry Pi OS with desktop	Ubuntu 16.04
CPU	quad-core 1.2GHz	two vCPUs
Memory	1GB	4GB
Program Language	Python	Python

TABLE V
EXECUTION TIME OF BASIC OPERATIONS (MS)

Notation	Description	Raspberry Pi	Amazon EC2
T_h	hash function	0.0203	0.0043
T_s	symmetric encryption	0.1490	0.0250
T_c	chaotic map	0.3042	0.0450
T_b	bilinear pairing	52.7565	6.3910
T_m	scalar multiplication	0.4060	0.0630

A. Computation Costs Analysis

This subsection discloses the computation costs of ALAKAP. The configuration parameters of the two test platforms are shown in Table IV. One is the Amazon EC2 platform, where each VM is the t2.medium type with two vCPUs and 4GB memory, running Ubuntu 16.04. It was also utilized to simulate the cloud server and fog nodes. Another one is the single-board computer (i.e., Raspberry Pi 3 Model B), where the operating system is Raspberry Pi OS with desktop, the system configuration is an ARM Cortex-A53 1.2GHz quad-core and equipped with 1GB RAM. As such, it represents the IoT equipments in this paper. And the execution time of some essential operations which have been tested on the Raspberry Pi 3 Model B and Amazon EC2, respectively, is presented in Table V. The evaluation was realized by crypto library, hashlib library and NumPy library in Python. Notably, the XOR operation is ignored since its execution time is negligible [8].

Table VI and Fig. 5 reveal the total computation costs of user U_i , fog node FN_j and cloud server S , in different protocols. The computation costs of U_i in [3], [4], [10] and ALAKAP are $2T_m + 6T_h + T_b = 53.6903ms$, $3T_m + 4T_h = 1.2992ms$, $9T_m + 5T_h = 3.7555ms$ and $2T_c + 7T_h = 0.7505ms$, respectively. On the fog node side, the computation costs in [3], [4], [10] and ALAKAP are $2T_m + 4T_h + T_b = 6.5342ms$, $4T_m + 4T_h = 0.2692ms$, $8T_m + 10T_h = 0.5470ms$ and $3T_c + 4T_h = 0.1522ms$, separately. The cloud server side is $3T_m + 11T_h + T_b = 6.6273ms$, $8T_m + 11T_h = 0.5513ms$, $5T_m + 4T_s + 6T_h = 0.4408ms$ and $3T_c + 4T_s + 13T_h = 0.2909ms$, respectively. Clearly, the computation costs of different sides in our ALAKAP are superior to that of [3], [4], [10]. Especially, the total computation costs of ALAKAP is 55 times less than that of [3] and can achieve at least 44% higher improvement than the state-of-the-artwork [4].

TABLE VI
COMPARISONS OF COMPUTATION COSTS (MS)

Computation Costs	Jia <i>et al.</i> [3]	Ma <i>et al.</i> [4]	Liu <i>et al.</i> [10]	Ours
User U_i	$2T_m + 6T_h + T_b$ (53.6903)	$3T_m + 4T_h$ (1.2992)	$9T_m + 5T_h$ (3.7555)	$2T_c + 7T_h$ (0.7505)
Fog Node FN_j	$2T_m + 4T_h + T_b$ (6.5342)	$4T_m + 4T_h$ (0.2692)	$8T_m + 10T_h$ (0.5470)	$3T_c + 4T_h$ (0.1522)
Server S	$3T_m + 11T_h + T_b$ (6.6273)	$8T_m + 11T_h$ (0.5513)	$5T_m + 4T_s + 6T_h$ (0.4408)	$3T_c + 4T_s + 13T_h$ (0.2909)
Total	(66.8518)	(2.1197)	(4.7433)	(1.1936)

TABLE VII
COMPARISON OF COMMUNICATION COSTS (BITS)

Protocol	Communication cost of U_i	Communication cost of FN_j	Communication cost of S	Total
Jia <i>et al.</i> [3]	$ G + 4 q + T = 1696$	$4 G + 6 q + 3 T = 5152$	$ G + 4 q + T = 1696$	8544
Ma <i>et al.</i> [4]	$ G + 3 q + T = 1536$	$6 G + 6 q + 3 T = 7200$	$3 G + 4 q + T = 3744$	12480
Liu <i>et al.</i> [10]	$2 G + 3 q + T = 2560$	$5 G + 5 q + 2 T = 5984$	$2 G + 2 E + 2 q + T = 2912$	11456
Ours	$ E + 5 q + T = 1088$	$2 E + 12 q + 3 T = 2528$	$2 E + 8 q + T = 1824$	5440

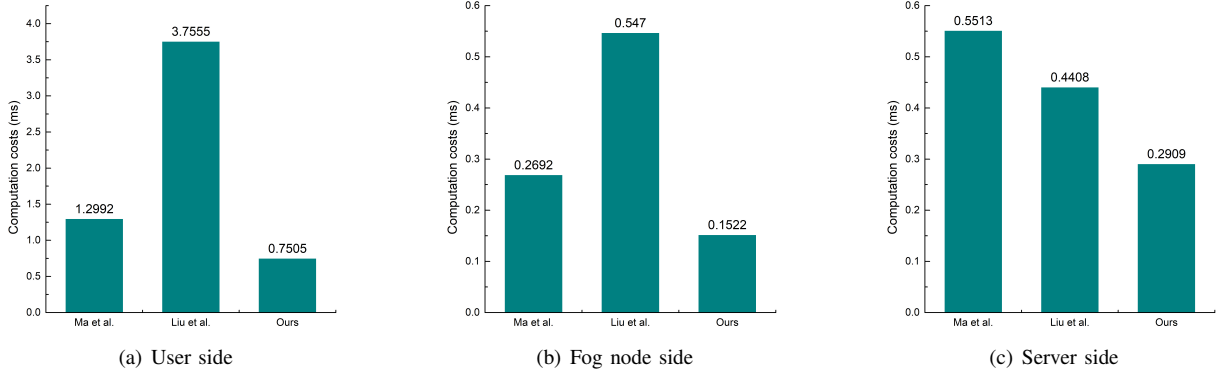


Fig. 5. Comparison of computation costs

B. Communication Costs and Storage Overhead Analysis

In this subsection, we compute the communication costs among user U_i , fog node FN_j and cloud server S in different protocols. To make a formal judgment about the communication costs, in this paper, we denote that the length of a point in group G as $|G|$, which is 1024 bits. The length of the key for symmetric encryption/decryption denoted as $|E|$ is 256 bits. The length of Chebyshev chaotic map operation denoted as $|C|$ is 160 bits. We assume that the identity digest of the user and fog node identity, hash function and nonce also have a length of 160 bits, which is indicated as $|q|$. The length of timestamp denoted as $|T|$ is 32 bits.

Table VII and Fig. 6 display the comparison of communication costs in different protocols. Based on the aforementioned assumptions, the communication costs of U_i in [3], [4], [10] and ALAKAP are $|G| + 4|q| + |T| = 1696$ bits, $|G| + 3|q| + |T| = 1536$ bits, $2|G| + 3|q| + |T| = 2560$ bits and $|E| + 5|q| + |T| = 1088$ bits, respectively. On the fog node side, the communication costs of [3], [4], [10] and ALAKAP are $4|G| + 6|q| + 3|T| = 5152$ bits, $6|G| + 6|q| + 3|T| = 7200$ bits, $5|G| + 5|q| + 2|T| = 5984$ bits and $2|E| + 12|q| + 3|T| = 2528$ bits, separately. The cloud server side are $|G| + 4|q| + |T| = 1696$ bits, $3|G| + 4|q| + |T| = 3744$ bits, $2|G| + 2|E| + 2|q| + |T| = 2912$ bits and $2|E| + 8|q| + |T| = 1824$ bits, respectively. In Fig. 6(c), although the communication costs of ALAKAP

is slightly higher than that of [3], the total communication costs of ALAKAP is well below that of [3]. Obviously, it is observed that the communication costs of ALAKAP is much less than that of [3], [4], [10].

VIII. CONCLUSION

Fog computing has been implemented in an extensive range of applications such as the context (H-IoT system) in this paper. Therefore, ensuring the security and privacy of a fog-assisted H-IoT system (focus on authentication and key agreement) is significantly necessary. In order to establish a secure mutual authentication for fog computing architecture, in this paper, a privacy-preserving AKA protocol for fog-assisted H-IoT system called ALAKAP is proposed by leveraging lightweight cryptographic primitive (i.e., Chebyshev chaotic map operation) to achieve security and efficiency simultaneously. Informal security analysis (security is reduced to CMCDHP problem) and formal security analysis of ALAKAP utilizing the broadly accepted BAN logic model are presented, respectively. We demonstrate the high-efficiency in terms of computation costs and communication costs of our design by comparing with the state-of-the-art works through the extensive experiments which are performed on Amazon EC2 and single-board computer Raspberry Pi 3 Model B.

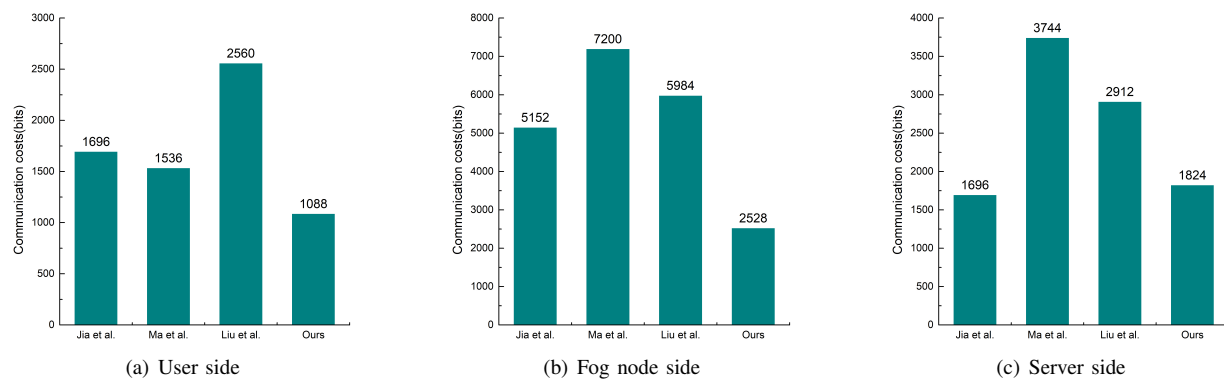


Fig. 6. Comparison of communication costs

As part of our future work, we will probe into how to build up the efficiency of our proposal such as reducing the times of the basic operations, to be more applicable for a variety of lightweight applications such as smart traffic. Moreover, we consider introducing blockchain technology to assist in accomplishing mutual authentication and key agreement. We also intend to collaborate with a health center's operator to accomplish and evaluate a prototype of the extended system for achieving higher utility in the real world.

ACKNOWLEDGEMENT

This work is supported in part by the National Key Research and Development Program of China (Grant No.2020YFB1005500), the National Natural Science Foundation of China (Grant No.61972310, 61972017, 61941114 and 62072487).

REFERENCES

- [1] C. Lin, D. He, N. Kumar, X. Huang, P. Vijaykumar, and K.-K. R. Choo, "Homechain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818–829, 2020.
- [2] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare internet of things: A survey of emerging technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020.
- [3] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven iot healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.
- [4] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8065–8075, 2019.
- [5] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.
- [6] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 546–556, 2014.
- [7] X. Li, F. Wu, L. Junguo, and Y. Liu, "Efficient three-party authenticated key agreement protocol based on chaotic map," *Chinese Journal of Network & Information Security*, vol. 2, no. 6, pp. 13–21, 2016.
- [8] L. Zhang, S. Zhu, and S. Tang, "Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme," *IEEE Journal of Biomedical & Health Informatics*, vol. 21, no. 2, pp. 465–475, 2017.
- [9] P. Roychoudhury, B. Roychoudhury, and D. K. Saikia, "Provably secure group authentication and key agreement for machine type communication using chebyshevs polynomial," *Computer Communications*, vol. 127, pp. 146–157, 2018.
- [10] X. Liu, W. Ma, and H. Cao, "Npma: A novel privacy-preserving mutual authentication in tmis for mobile edge-cloud architecture," *Journal of Medical Systems*, vol. 43, no. 10, pp. 318:1–318:16, 2019.
- [11] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE transactions on dependable and secure computing*, vol. 15, no. 4, pp. 708–722, 2016.
- [12] P. Mall, R. Amin, A. K. Das, M. T. Leung, and K.-K. R. Choo, "Puf-based authentication and key agreement protocols for iot, wsns, and smart grids: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8205–8228, 2022.
- [13] H. Amintoosi, M. Nikooghadam, M. Shojafar, S. Kumari, and M. Alazab, "Slight: A lightweight authentication scheme for smart healthcare services," *Computers and Electrical Engineering*, vol. 99, p. 107803, 2022.
- [14] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.
- [15] Y. Chen and J. Chen, "An efficient and privacy-preserving mutual authentication with key agreement scheme for telecare medicine information system," *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 516–528, 2022.
- [16] J. Srinivas, A. K. Das, N. Kumar, and J. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 942–956, 2020.
- [17] S. F. Aghili, H. Mala, M. Shojafar, and P. Peris-Lopez, "Laco Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in iot," *Future Generation Computer Systems*, vol. 96, pp. 410–424, 2019.
- [18] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, "User centric three-factor authentication protocol for cloud-assisted wearable devices," *International Journal of Communication Systems*, vol. 32, no. 6, pp. e3900.1–e3900.20, 2019.
- [19] V. Odelu, S. Saha, R. Prasath, L. Sadineni, M. Conti, and M. Jo, "Efficient privacy preserving device authentication in wbans for industrial e-health applications," *Computers & Security*, vol. 83, pp. 300–312, 2019.
- [20] H. Debiao, C. Jianhua, and Z. Rui, "A more secure authentication scheme for telecare medicine information systems," *Journal of medical systems*, vol. 36, no. 3, pp. 1989–1995, 2012.
- [21] J. Wei, X. Hu, and W. Liu, "An improved authentication scheme for telecare medicine information systems," *Journal of medical systems*, vol. 36, no. 6, pp. 3597–3604, 2012.
- [22] Z. Zhu, "An efficient authentication scheme for telecare medicine information systems," *Journal of medical systems*, vol. 36, no. 6, pp. 3833–3838, 2012.
- [23] Q. Jiang, J. Ma, Z. Ma, and G. Li, "A privacy enhanced authentication scheme for telecare medical information systems," *Journal of medical systems*, vol. 37, pp. 1–11, 2013.
- [24] S. Kumari, M. K. Khan, and R. Kumar, "Cryptanalysis and improvement of a privacy enhanced scheme for telecare medical information systems," *Journal of medical systems*, vol. 37, pp. 1–11, 2013.
- [25] C.-L. Chen, T.-T. Yang, M.-L. Chiang, and T.-F. Shih, "A privacy authentication scheme based on cloud for medical environment," *Journal of medical systems*, vol. 38, no. 11, p. 143, 2014.
- [26] S.-Y. Chiou, Z. Ying, and J. Liu, "Improvement of a privacy authentication scheme for telecare medical information systems," *Journal of medical systems*, vol. 38, no. 11, p. 143, 2014.

- tication scheme based on cloud for medical environment,” *Journal of medical systems*, vol. 40, no. 4, pp. 101:1–101:15, 2016.
- [27] P. Mohit, R. Amin, A. Karati, G. Biswas, and M. K. Khan, “A standard mutual authentication protocol for cloud computing based health care system,” *Journal of medical systems*, vol. 41, no. 4, pp. 50:1–50:13, 2017.
- [28] C.-T. Li, D.-H. Shih, and C.-C. Wang, “Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems,” *Computer methods and programs in biomedicine*, vol. 157, pp. 191–203, 2018.
- [29] X. Liu and W. Ma, “Etap: Energy-efficient and traceable authentication protocol in mobile medical cloud architecture,” *IEEE Access*, vol. 6, pp. 33 513–33 528, 2018.
- [30] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, “Secure remote user authenticated key establishment protocol for smart home environment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2020.
- [31] C.-C. Chang and H.-D. Le, “A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks,” *IEEE Transactions on wireless communications*, vol. 15, no. 1, pp. 357–366, 2015.
- [32] H. Qiao, X. Dong, and Y. Shen, “Authenticated key agreement scheme with strong anonymity for multi-server environment in tms,” *Journal of medical systems*, vol. 43, no. 11, pp. 321:1–321:13, 2019.
- [33] H. A. A. Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, “A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography,” *IEEE Access*, vol. 5, pp. 22 313–22 328, 2017.
- [34] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, “A secure authenticated and key exchange scheme for fog computing,” *Enterprise Information Systems*, vol. 15, no. 9, pp. 1200–1215, 2021.
- [35] S. Shamshad, M. S. Obaidat, U. Shamshad, S. Noor, K. Mahmood *et al.*, “On the security of authenticated key agreement scheme for fog-driven iot healthcare system,” in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 1760–1765.
- [36] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, “Improved authenticated key agreement scheme for fog-driven iot healthcare system,” *Security and Communication Networks*, vol. 2021, pp. 1–16, 2021.
- [37] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Advances in Cryptology – CRYPTO’99*, 1999, pp. 388–397.
- [38] T. Messerges, E. Dabbish, and R. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.
- [39] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, “Bsein: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0,” *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 2018.
- [40] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2014.
- [41] L. Zhang, “Cryptanalysis of the public key encryption based on multiple chaotic systems,” *Chaos, Solitons and Fractals*, vol. 37, no. 3, pp. 669–674, 2008.
- [42] L. Kocarev and S. Lian, *Chaos-based cryptography: Theory, algorithms and applications*. Springer Publishing Company, Incorporated, 2011.