Journal of Medical Systems (2019) 43: 318
https://doi.org/10.1007/s10916-019-1444-9

MOBILE & WIRELESS HEALTH

NPMA: A Novel Privacy-Preserving Mutual Authentication in TMIS for Mobile Edge-Cloud Architecture

Xiaoxue Liu[1] · Wenping Ma[1] · Hao Cao[1,2]

Received: 27 June 2019 / Accepted: 28 August 2019 / Published online: 14 September 2019

Abstract
Mobile Edge-Cloud Network is a new network structure after fog-cloud computing, where service and data computing are scattered in the most logical, nearby and efficient place. It provides better services than fog-cloud computing with better performance in reasonably low cost way and allows users to eliminate numerous limitations inherent in fog-cloud computing, although it inherits those security-privacy issues from fog-cloud computing. A novel privacy-preserving mutual authentication in TMIS for mobile Edge-Cloud architecture (abbreviated to NPMA) is constructed in this paper. NPMA scheme not only mitigates some weaknesses of fog-cloud computing, but has other advantages. First, NPMA scheme supports patients(edge-servers) anonymity and forward-backward untraceability (traceability, when needed), since their identities are hidden in two distinct dynamic anonyms and a static one and only the trusted center can recover their real identities, when needed. Second, each edge-server shares a secret value, which realizes authentication with extremely low computional cost in authentication phase. Finally, NPMA scheme is proven safely against passive and active attacks under elliptic curve computable Diffie-Hellman problem (ECDHP) assumption in random oracle model. Hence, it achieves the required security properties and outperforms prior approaches in terms of energy and computational costs.

Keywords Mobile edge-cloud network · Privacy-preserving · Authentication · TMIS · Dynamic anonym

Introduction

Telecare Medical Information System (TMIS) for mobile edge-cloud computing (MEC) architecture is a new paradigm of blending medical care, edge-cloud computing and cloud storage together to provide high quality e-health care for patients [1]. MEC is an extension of fog-cloud computing, which is at the furthest edge of the network. Further, MEC not only offers data, storage, computing and services to the patients [2], but also mitigates the weaknesses of fog-cloud computing. In fog-cloud computing network, devices usually generate a large amount of data, which can cause tremendous pressure on fog-cloud server. Meanwhile, due to the centralized nature and physical location of cloud server, it can be challenging for remote cloud server to respond the real-time requirements.

However, MEC is a potential solution to solve the aforementioned weaknesses of fog-cloud computing. In order to share the pressure of the central cloud server, MEC servers can be responsible for the real-time and short-period data processing tasks as well as the real-time processing and execution within their own scopes, as depicted in Fig. 1. Hence, it can provide high-value data for the cloud. Cloud computing is responsible for computing tasks that MEC servers are incompetent. Further, through large data analysis, cloud servers can transfer some non-real-time, long-period data processing tasks to MEC servers. In this way, MEC can meet the local needs, and then complete the application life cycle management. Therefore, cloud computing and MEC form a complementary and

This article is part of the Topical Collection on *Mobile & Wireless Health*

✉ Xiaoxue Liu
862417756@qq.com

Wenping Ma
wp_ma@mail.xidian.edu.cn

Hao Cao
13655505689@163.com

[1] Xidian University, No.2, South Taibai Road, Yanta District, Xi'an, China

[2] Anhui Science and Technology University, Chuzhou, 233100, China

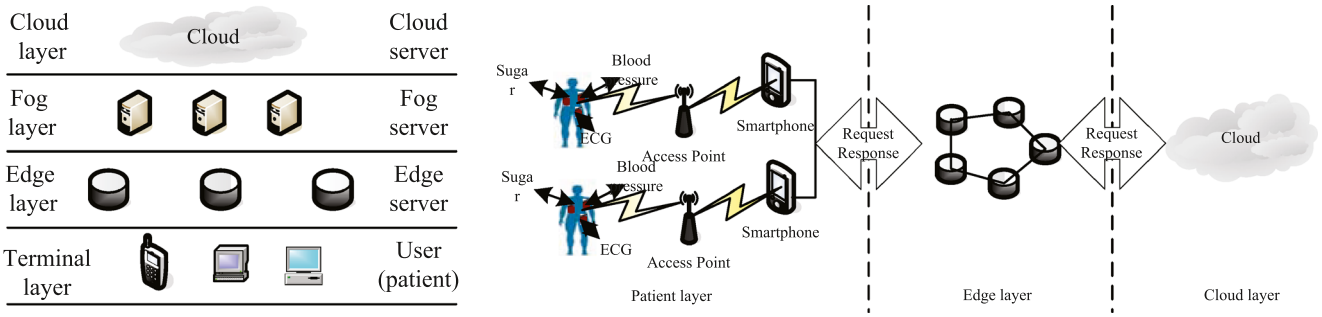

Fig. 1 Architecture for accessing medical edge-cloud service

collaborative relationship to better meet people's medical needs [3].

Due to the benefits provided by MEC, the remote patients can obtain good medical resources directly access their medical devices efficiently without any long delay. Patients are monitored by electronic medical devices or tiny sensors to collect their vital signs and other health data, which are sent to cloud via near edge server. Based on these signs and data, doctors give the corresponding diagnosises, directly and timely. It can fundamentally solve the difficult problem of medical treatment in remote areas.

Clearly, e-health care is completely dependent on these signs and data. However, they are transmitted and exposed during the unsecured public communication channel. Adversaries may eavesdrop, intercept, delete, and modify all the data in this channel, easily [4, 5]. When the least expected thing happens, one unauthorized adversary may obtain a patients current health condition, medical history and other binding fixed information like mobile phone number and credit card number. Considering the worst condition, if the adversary has an attempt at harming the patient, he may modify this patient's vital health data. Once, these modified data is transmitted to his doctors, the wrong diagnosis can be made and the patient's life may be threatened [6–8]. Undoubtedly, the patient will suffer much more than the illness itself. Not surprisingly, they are the often case in our daily routine. Obviously, the patients' privacy protection has not been adequately addressed and it is still an urgent and much stronger requirement in TMIS.

Although MEC service is at its inception, it still inherits those security-privacy issues, which in turn causes a serious security concern in MEC environment. Undoubtedly, security is the primary concern and patients's privacy is also one of the nontrivial concerns in MEC architecture [1–3]. In order to achieve efficiency, convenience, security and privacy of the said system, many efforts have been made.

Main contributions

In order to provide efficiency, convenience, security and privacy medical care, a novel privacy-preserving mutual authentication is constructed in this paper and the main contributions of NPMA scheme are summarized below:

- Security, privacy-preserving, accountability (patients/edge-servers), efficiency and dynamic removal (edge-servers) requirements for the authentication framework in MEC architecture. Meanwhile, NPMA scheme is the first study to construct an authentication scheme in this application scenario on the scope of our knowledge.
- For the multi-senders mode (edge-servers to cloud) in NPMA scheme, secret sharing is utilized to design a key establishment between edge-servers with cloud without key negotiation rounds, which undoubtedly accelerates the authenticating speed. In other words, only authorized edge-servers (i.e. registered at trust center TC) can get the secret shared value to transmit legal ciphertext to cloud. What's more, a large amount of ciphertexts may arrive over the same period, NPMA scheme also provides a batch verification algorithm to improve the efficiency. Hence, NPMA scheme is very suitable for computation-limited mobile devices.
- In order to guarantee the integrity, timeliness and confidentiality of the data stored in the cloud, we design a novel feedback (response) mechanism based on patients(edge-servers) temporary keys, which are only obtained by patients, edge-servers and cloud, correspondingly. When patients successfully store a message into the cloud, cloud and edge server feedback the result to patients in a timely manner.
- NPMA scheme based on certificateless cryptography can overcome the key escrow problem of identity-based public key cryptography. The full private keys of participants consist of two parts: the secret value chosen by participants themselves and the partial secret keys generated by TC. It properly resolves the complicated certificate management problems in traditional public key infrastructure system.
- NPMA scheme is proved to be secure under the Elliptic Curve Computational Diffie-Hellman problem (ECDHP) (For details, see Section "Cryptography materials") assumption in the random oracle model.

The NPMA scheme is proved secure against possible known attacks and satisfy the secure requirements of authentication scheme. Hence, the NPMA scheme is practical in complex network environment.

Related works

MEC is closely related to fog computing, which is introduced by Cisco Systems [9]. Based on its initial definition, it was deemed as an extension of cloud computing paradigm, which provides computing, storage, and networking services between end devices and cloud [10]. Stojmenovic et al. [11] discussed the advantages of fog computing and its applications, such as smart traffic lights in vehicular networks, smart grid and software defined networks. Subsequently, most authentication schemes have been proposed [2, 12–14]. Koo et al. [12] designed a privacy-preserving deduplication technique for efficient ownership management in fog computing. Wang et al. [13] also presented a technique for anonymous and secure aggregation scheme in fog-cloud computing in which a fog node aggregates the data from terminal nodes, and forwards the aggregated data to cloud server. Wazid et al. [2] proposed a secure key management and user authentication scheme for fog computing services, where smart devices and fog servers are able to establish pairwise secret keys for their secure communication. Different from above ones, Ma et al. [14] added the fog computing into vehicular Ad-Hoc networks. But fog computing has not fundamentally eliminated the weaknesses of cloud computing, these schemes are not applicable to remote area.

MEC was firstly used to describe the execution of services at the edge of the network in 2013, when IBM and Nokia Siemens Network introduced a platform that could run applications within a mobile base station [15]. Subsequently, some authentication schemes have been proposed [16–18]. Considering the time consumption of the computing task execution and the mobility of the vehicles, Zhang et al. [16] presented an efficient predictive combination-mode relegation scheme, where tasks are adaptively off-loaded to the MEC servers through direct uploading or predictive relay transmissions. Illustrative results indicate that the proposed scheme greatly reduced the cost of computation and improves task transmission efficiency. In [17], a provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. Li et al. [18] defined the privacy preserving data aggregation for MEC assisted IoT applications. In [19, 20], the authors presented their vision of exploiting MEC for e-health applications. They held that among the most promising approaches for enabling e-health are MEC capabilities and next-generation wireless networking technologies that can provide real-time and cost-effective patient remote monitoring.

Subsequently, most authentication authentications have been proposed [21–26]. However, these did not take into account MEC scenarios.

Organization

The rest of paper is organized as follows. Some cryptography materials about NPMA scheme are introduced in Section "Cryptography materials". Section "NPMA: A novel privacy-preserving mutual authentication for edge-cloud architecture" presents the NPMA scheme. Detailed security analysis and proof are given in Section "Security analysis and proof of NPMA scheme". The comparisons of the performance and security features between NPMA scheme with other related schemes are discussed in Section "Performance evaluation". Section "Conclusion and ongiong work" concludes this paper.

Cryptography materials

In this section, we review some cryptography materials used in NPMA scheme.

Table 1 Notations

Symbol	Description
PA_i, Es_j	the ith patient and jth edge-sever, respectively
C	the cloud-sever
TC	the trusted center
ID_{PA_i}, ID_{Es_j}	the identity of PA_i and Es_j
pid_{PA_i}, pid_{Es_j}	the static anonym of PA_i
$PID^1_{PA_i}, PID^2_{PA_i}$	the dynamic anonyms of PA_i
$PID^1_{Es_j}, PID^2_{Es_j}$	the dynamic anonyms of Es_j
(pk, sk)	the pair of public and private keys
λ	the security parameter
$\{s, c_1, c_2\}$	the master private key of TC
t	the number of edge-sever
θ	the secret shared value
$E_{key}(M)$, $D_{key}(M)$	encrypt/decrypt the message M by using key key
T^i	ith timestamp
ΔT	maximum transmission delay associated with a message
$h(\cdot)$	a cryptographically secure one way hash function
$\oplus$, $\|\|$	bitwise XOR operation and concatenation operation
$\Rightarrow$	a secured communication channel
$\rightarrow$	a public communication channel

Elliptic Curve discrete logarithm problem(**EDLP**): Given $(P, aP) \in G_p$ for any unknown $a \in Z_q^*$, the goal of the EDLP is to compute a. Define the advantage of any probabilistic polynomialtime algorithm $\mathcal{A}$ against EDLP in G_p. For every probabilistic $\mathcal{A}$, the advantage is negligible, which will be used in the security analysis of NPMA scheme.

Elliptic Curve Computable Diffie-Hellman problem (**ECDHP**): Choose G_p as elliptic curve group G_p with the generator of P, whose order is a prime q. Given $(P, aP, bP) \in G_p$ for any unknown $a, b \in Z_q^*$, the goal of the ECDHP is to compute abP. For every probabilistic $\mathcal{A}$ against ECDHP in G_p, the advantage is negligible.

NPMA: A novel privacy-preserving mutual authentication for edge-cloud architecture

The NPMA is composed of eight basic phases: **System setup phase**, **Patient registration phase**, **Edge-server registration phase**, **Cloud-server registration phase**, **Shared value setting phase**, **Request and Authentication phase**, **Response phase**, and **Tracing, Removing and Adding phase**. To simplify the subsequent description, some symbol notations are given in Table 1. Figure 1 simply depicts the MEC network model. At the beginning, trusted center TC sets up the systems:

System setup phase

Inputs a security parameter λ, trusted center TC is implemented to generate system parameters. Below are the detailed steps:

(1) TC chooses a q-order additive group G_p with a generator P over a non-singular elliptic curve $E(F_p)$, where p and q are large prime numbers;
(2) Selects seven cryptographically secure one-way hash functions $h_1(\cdot)$: $\{0,1\}^* \rightarrow Z_q^*$, $h_2(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^*$, $h_3(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^{\lambda}$, $h_4(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^{\lambda}$, $h_5(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^{\lambda}$, $h_6(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$, $h_7(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^*$;
(3) Chooses a cryptographic symmetric encryption/decryption pair $E_{key}(\cdot)/D_{key}(\cdot)$ with symmetric key key;
(4) TC selects $\tau \in_R G_p/1_{G_p}$ and $c_1, c_2 \in_R Z_q^*$, sets $u, v \in G_p$, where $c_1 u = c_2 v = \tau$;
(5) Chooses a selects $s \in_R Z_q^*$, and calculates $pk_{TC} = sP$;
(6) Publishes the system parameters $PP = \{p, P, u, v, \tau, E_{key}(\cdot)/D_{key}(\cdot), pk_{TC}, h_i(\cdot)(i = 1...7)\}$;
(7) Keeps $MSK = \{s, c_1, c_2\}$ as the master private key.

Patient registration phase

If a patient PA_i wants to send his/her health data to cloud, he/she should register in trusted center TC firstly. The steps and the detailed interactive processes between between PA_i and TC are shown in Fig. 2.

(1) PA_i chooses his/her ID_{PA_i}, secret value $x_{PA_i} \in_R Z_q^*$ and computes $P_{PA_i} = x_{PA_i} P$. Then, PA_i sends registration request message $\{ID_{PA_i}, P_{PA_i}\}$ to TC through a secured channel;
(2) Upon receiving the registration message from PA_i, TC selects two random values $\{\xi_{PA_i}, q_{PA_i}\} \in_R Z_q^*$, computes $pid_{PA_i} = E_s(ID_{PA_i}, \xi_{PA_i})$, $Q_{PA_i} = q_{PA_i} P$, $\alpha_{PA_i} = h_1(P_{PA_i} || Q_{PA_i} || pk_{TC})$, and $y_{PA_i} = \alpha_{PA_i} q_{PA_i} + s$. Then, returns $\{pid_{PA_i}, (y_{PA_i}, Q_{PA_i})\}$ to PA_i also through a secured channel;
(3) After receiving the return message from TC, PA_i computes $\alpha_{PA_i} = h_1(P_{PA_i} || Q_{PA_i} || pk_{TC})$ and checks $y_{PA_i} P ? = \alpha_{PA_i} Q_{PA_i} + pk_{TC}$. If it does not hold, terminates it. Otherwise, PA_i sets $sk_{PA_i} = (x_{PA_i}, y_{PA_i})$ and $pk_{PA_i} = (P_{PA_i}, Q_{PA_i})$;
(4) PA_i stores$\{pid_{PA_i}, sk_{PA_i}\}$ securely and publishes pk_{PA_i}.

Edge-server registration phase

When a edge-server Es_j wishes to join the system, he/she should register in TC firstly. The operations illustrated in Fig. 3 are explained as follows:

(1) Es_j with the identity ID_{Es_j} chooses a secret value $x_{Es_j} \in_R Z_q^*$ and computes $P_{Es_j} = x_{Es_j} P$. Then, Es_j

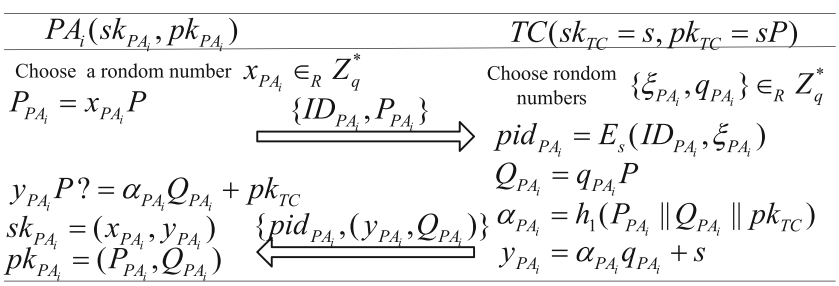

Fig. 2 Patient registration phase

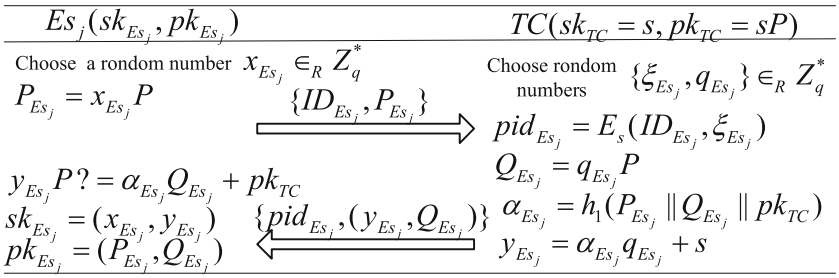

Fig. 3 Edge-server registration phase

sends $\{ID_{Es_j}, P_{Es_j}\}$ to TC with registration request through a secured channel;

(2) Upon receiving the registration message from Es_j, TC chooses two random values $\{\xi_{Es_j}, q_{Es_j}\} \in_R Z_q^*$, computes $pid_{Es_j} = E_s(ID_{Es_j}, \xi_{Es_j})$, $Q_{Es_j} = q_{Es_j} P$, $\alpha_{Es_j} = h_1(P_{Es_j} || Q_{Es_j} || pk_{TC})$ and $y_{Es_j} = \alpha_{Es_j} q_{Es_j} + s$. Then, returns $\{pid_{Es_j}, (y_{Es_j}, Q_{Es_j})\}$ to Es_j through a secured channel;

(3) After receiving the return message from TC, Es_j computes $\alpha_{Es_j} = h_1(P_{Es_j} || Q_{Es_j} || pk_{TC})$ and checks $y_{Es_j} P ? = \alpha_{Es_j} Q_{Es_j} + pk_{TC}$. If it does not match, terminates it. Otherwise, Es_j sets $sk_{Es_j} = (x_{Es_j}, y_{Es_j})$ and $pk_{Es_j} = (P_{Es_j}, Q_{Es_j})$.

(4) Es_j stores $\{pid_{Es_j}, sk_{Es_j}\}$ securely and publishes pk_{Es_j}.

Cloud-server registration phase

When the cloud-server C wants to be a legal medical service provider in this special system, similarly, he/she should register in TC to obtain his/her public/private key pairs. The details of this phase are shown in Fig. 4 and described below:

(1) C chooses a secret value $x_c \in_R Z_q^*$ and computes $P_c = x_c P$. Then, C sends P_c to TC with registration request through a secured channel;

(2) Upon receiving the registration message from C, TC chooses a random value $q_c \in_R Z_q^*$, computes $Q_c = q_c P$, $\alpha_c = h_1(P_c || Q_c || pk_{TC})$ and $y_c = \alpha_c q_c + s$. Then, returns $\{y_c, Q_c\}$ to C also through a secured channel;

(3) After receiving the return message from TC, C computes $\alpha_c = h_1(P_c || Q_c || pk_{TC})$ and checks $y_c P ? = \alpha_c Q_c + pk_{TC}$. If the value is valid, C sets $sk_c = (x_c, y_c)$ and $pk_c = (P_c, Q_c)$.

(4) C stores sk_c securely and publishes pk_c.

Shared value setting phase

Suppose $\{Es_1, Es_2, ..., Es_t\}$ is the validation edge-server set of MEC network, where t is the number of edge-sever. To guarantee the privacy preserving message delivery and minimize the number of the messages exchanged between the edge-servers and C, C generates a secret shared value θ, computes and releases pk_{Es} as an extemporaneous group shared public key. Each edge-server encrypts the authentication messages by using pk_{Es}, which can be a huge time savings for message authentication. It is required that all the authentication messages only can be authenticated by C with his/her secret key and the adversary cannot recover θ from pk_{Es}.

(1) C chooses a random value $r \in_R Z_q^*$, computes $R = rP$, $\alpha_{Es_j} = h_1(P_{Es_j} || Q_{Es_j} || pk_{TC})$, $S_{Es_j} = r(\alpha_{Es_j} Q_{Es_j} + pk_{TC} + P_{Es_j})$ and $\beta_{Es_j} = h_1(R, S_{Es_j})$, where $j = 1...t$;

(2) C chooses the other random value $\theta \in_R Z_q^*$ as the secret shared value, computes $pk_{Es} = \theta P$ and $f(x) = \prod(x - \beta_{Es_j}) + \theta \pmod q = x^t + a_{t-1}x^{t-1} + ... + a_1 x + a_0$. Here, $\{a_0, ..., a_{t-1}\} \in Z_q^*$;

(3) C stores θ securely and publishes $\{a_0, a_1, ..., a_{t-1}, R, pk_{Es}\}$

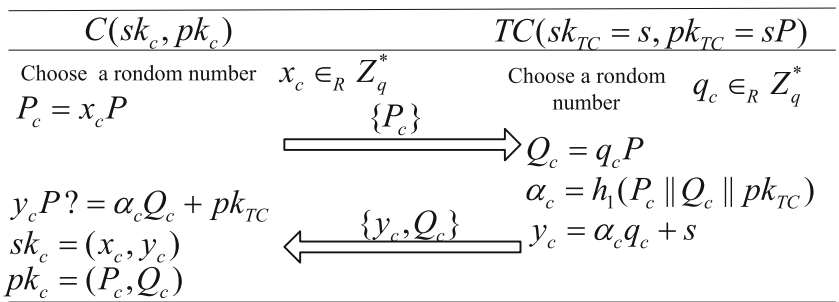

Fig. 4 Cloud-server registration phase

Request and authentication phase

PA_i obtains his/her health data $m_{PA_i} = (pid_{PA_i}, data_{PA_i}, T^1_{PA_i})$ via his/her mobile device (such as smartphone, iPad) from body sensor (in/on PA_i's body). Later, PA_i will store it in C for later access by his/her-self or other legally authorized ones ultimately. Here, PA_i sends the request to edge server Es_j nearby. The details of this phase as shown in Fig. 5 are described below:

(1) PA_i chooses a random value $a_{PA_i} \in_R Z^*_q$, computes $A_{PA_i} = a_{PA_i}P$, $\alpha_{Es_j} = h_1(P_{Es_j}||Q_{Es_j}||pk_{TC})$, $key_{P-E} = (x_{PA_i} + y_{PA_i} + a_{PA_i})(P_{Es_j} + \alpha_{Es_j}Q_{Es_j} + pk_{TC})$, $PID^1_{PA_i} = h_2(key_{P-E}, A_{PA_i}) \oplus pid_{PA_i}||m_{PA_i}$, $M_{i1} = h_3(pid_{PA_i}, PID^1_{PA_i}, key_{P-E}, A_{PA_i}, m_{PA_i}, T^1_{PA_i})$ and sets $Trans1 = \{PID^1_{PA_i}, A_{PA_i}, M_{i1}, T^1_{PA_i}\}$. Then, $PA_i \rightarrow Es_j$: $Trans1$;

(2) Upon receiving the request message from PA_i, Es_j reads the current time $T^1_{Es_j}$ and checks $|T^1_{Es_j} - T^1_{PA_i}| \leq \Delta T$. If that above verification does not hold, the request is rejected. Otherwise, Es_j computes $\alpha_{PA_i} = h_1(P_{PA_i}||Q_{PA_i}||pk_{TC})$, $key_{E-P} = (P_{PA_i} + \alpha_{PA_i}Q_{PA_i} + pk_{TC} + A_{PA_i})(x_{Es_j} + y_{Es_j})$, $pid_{PA_i}||m_{PA_i} = PID^1_{PA_i} \oplus$

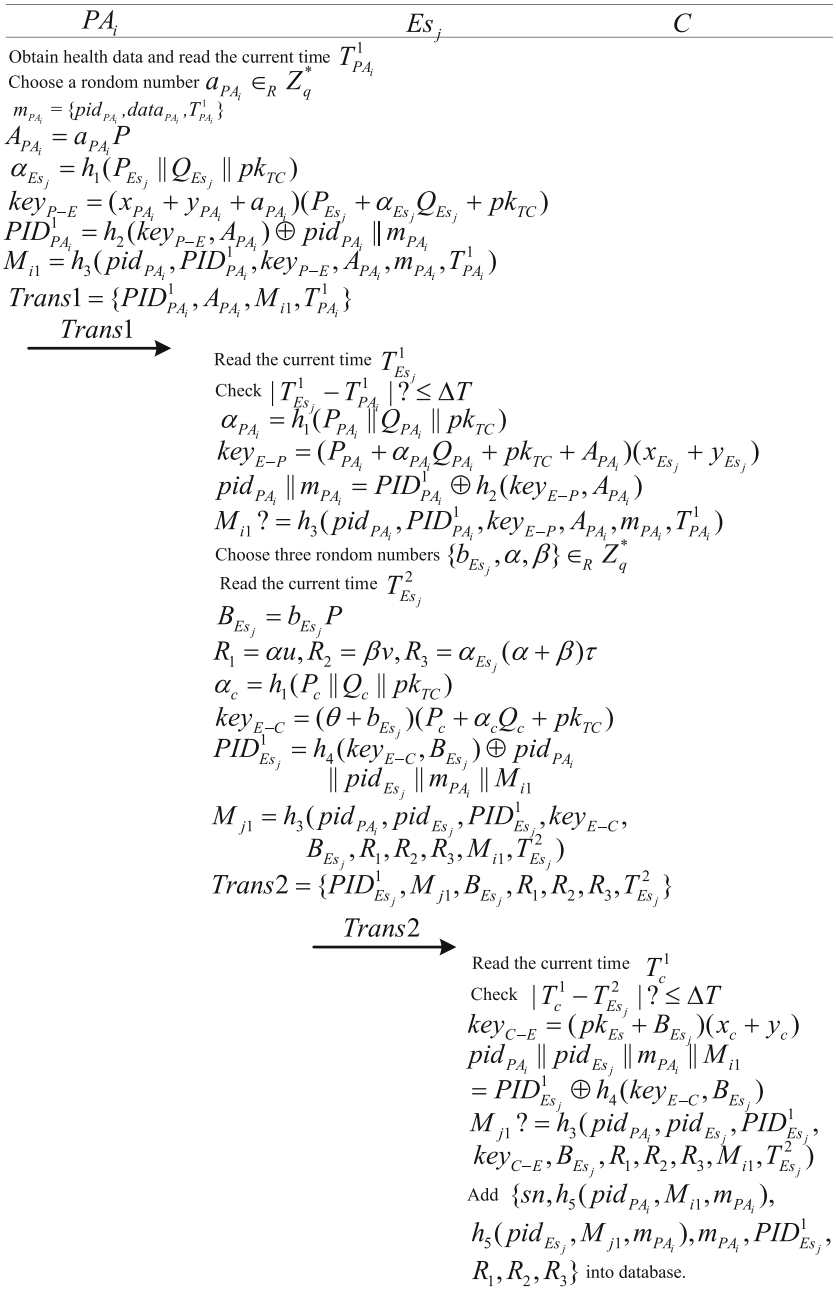

Fig. 5 Request and authentication phase

$h_2(key_{E-P}, A_{PA_i})$, and checks $M_{i1}?=h_3(pid_{PA_i}, PID^1_{PA_i}, key_{E-P}, A_{PA_i}, m_{PA_i}, T^1_{PA_i})$. If it does not hold, terminates it. Otherwise, Es_j chooses three random values $\{b_{Es_j}, \alpha, \beta\} \in_R Z^*_q$, reads the current time $T^2_{Es_j}$ and calculates $B_{Es_j} = b_{Es_j} P$, $R_1 = \alpha u$, $R_2 = \beta v$, $R_3 = \alpha_{Es_j}(\alpha+\beta)\tau, \alpha_c = h_1(P_c||Q_c||pk_{TC}), key_{E-C} = (\theta + b_{Es_j})(P_c + \alpha_c Q_c + pk_{TC})$, $PID^1_{Es_j} = h_4(key_{E-C}, B_{Es_j}) \oplus pid_{PA_i}||pid_{Es_j}||m_{PA_i}||M_{i1}$, $M_{j1} = h_3(pid_{PA_i}, pid_{Es_j}, PID^1_{Es_j}, key_{E-C}, B_{Es_j}, R_1, R_2, R_3, M_{i1}, T^2_{Es_j})$ and sets $Trans2 = \{PID^1_{Es_j}, M_{j1}, B_{Es_j}, R_1, R_2, R_3, T^2_{Es_j}\}$; Then, $Es_j \rightarrow C$: $Trans2$;

(3) On receiving the authentication message from Es_j, C reads the current time T^1_c and checks $|T^1_c - T^2_{Es_j}| \leq \Delta T$. If the value is valid, C computes $key_{C-E} = (pk_{Es} + B_{Es_j})(x_c + y_c)$, $pid_{PA_i}||pid_{Es_j}||m_{PA_i}||M_{i1} = PID^1_{Es_j} \oplus h_4(key_{E-C}, B_{Es_j})$ and checks $M_{j1}? = h_3(pid_{PA_i}||pid_{Es_j}, PID^1_{Es_j}, key_{C-E}, B_{Es_j}, R_1, R_2, R_3, M_{i1}, T^2_{Es_j})$. If it does not match, terminates it. Otherwise, C adds the pair $\{sn, h_5(pid_{PA_i}, M_{i1}, m_{PA_i}), h_5(pid_{Es_j}, M_{j1}, m_{PA_i}), m_{PA_i}, PID^1_{Es_j}, R_1, R_2, R_3\}$ into his/her database for later access by doctor or his/her-self ultimately, where the sequence number sn of this pair will pop up after m_{PA_i} has been saved. Further, based-on sn, the health historic data can be obtained from cloud data warehouse.

Response phase

After stored procedure executes successfully, C will give a response to Es_j. Upon receiving the response message from C, Es_j feedbacks the corresponding response to PA_i immediately. Thus, NPMA supports the need to manage the life-cycle of health data. Do the following steps as shown in Fig. 6:

(1) C reads the current time T^2_c and calculates $PID^2_{Es_j} = h_6(key_{C-E}, T^2_c) \oplus pid_{PA_i}||pid_{Es_j}||sn$, $M_{c1} = h_3(sn, pid_{PA_i}, pid_{Es_j}, PID^2_{Es_j}, key_{C-E}, M_{j1}, T^2_c)$ and sets $Trans3 = \{PID^2_{Es_j}, M_{c1}, T^2_c\}$; Then, $C \rightarrow Es_j$: $Trans$;

(2) Upon receiving the response message from C, Es_j reads the current time $T^3_{Es_j}$ and checks $|T^3_{Es_j} - T^2_c| \leq \Delta T$. If that above verification does not hold, the response is rejected. Otherwise, Es_j computes $pid_{PA_i}||pid_{Es_j}||sn = PID^2_{Es_j} \oplus h_6(key_{E-C}, T^2_c)$ and checks $M_{c1}?=h_3(sn, pid_{PA_i}, pid_{Es_j}, PID^2_{Es_j}, key_{C-E}, M_{j1}, T^2_c)$. If it does not hold, response is illegal. Otherwise, Es_j reads the current time $T^4_{Es_j}$ and calculates $PID^2_{PA_i} = h_7(key_{E-P}, T^4_{Es_j}) \oplus pid_{PA_i}||sn$, $M_{j2} = h_3(sn, pid_{PA_i}, PID^2_{PA_i}, key_{E-P}, M_{i1}, T^4_{Es_j})$ and sets $Trans4 = \{PID^2_{PA_i}, M_{j2}, T^4_{Es_j}\}$. Then, $Es_j \rightarrow PA_i$: $Trans4$;

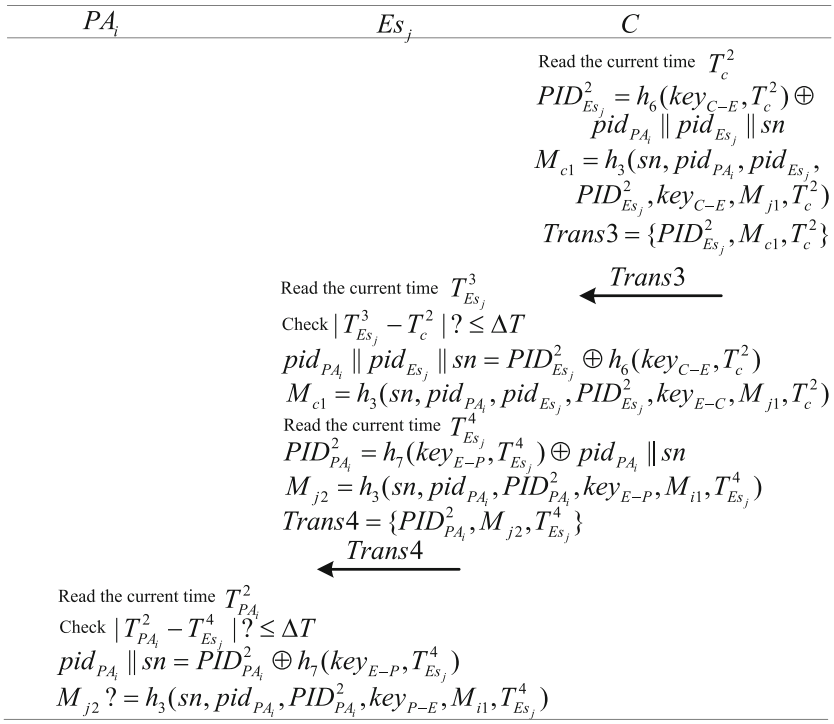

Fig. 6 Response phase

(3) On receiving the response message from Es_j, PA_i reads the current time $T^2_{PA_i}$ and checks $|T^2_{PA_i} - T^4_{Es_j}| \leq \Delta T$. If that above verification does not match, the response is rejected. Otherwise, PA_i computes $pid_{PA_i}||sn = PID^2_{PA_i} \oplus h_7(key_{P-E}, T^4_{Es_j}$ and checks $M_{j2}?=h_3(sn, pid_{PA_i}, PID^2_{PA_i}, key_{P-E}, M_{i1}, T^4_{Es_j})$. If it does not hold, response is illegal. Otherwise, PA_i believes that the stored procedure executes successfully.

Tracing, removing and adding phase

When PA_i complains to TC about that the health data m_{PA_i} has been tampered by a edge-server. TC dose following steps:

Tracing phase:

(1) TC obtains the stored pair $\{sn, h_5(pid_{PA_i}, M_{i1}, m_{PA_i}), h_5(pid_{Es_j}, M_{j1}, m_{PA_i}), m_{PA_i}, PID^1_{Es_j}, R_1, R_2, R_3\}$ according to m_{PA_i}. Based on this pair, TC considers the three elements $\{R_1, R_2, R_3\}$, computes the unique corresponding parameter of each edge-server $\alpha_{Es_j} = R_3/(c_1R_1 + c_2R_2)$ by using the managed private key $MSK = (c_1, c_2)$, since $R_3/(c_1R_1 + c_2R_2) = \alpha_{Es_j}(\alpha + \beta)\tau/(c_1\alpha u + c_2\beta v) = \alpha_{Es_j}$

(2) TC broadcasts α_{Es_j}, adds α_{Es_j} into blacklist and sends α_{Es_j} to C.

Removing phase:

Upon receiving the tracing message from TC, C searches the public key list to verify whose public key pair satisfies the equation $\alpha_{Es_j} = h_1(P_{Es_j}||Q_{Es_j}||pk_{TC})$. Then, C carries out the following to block all the following request from Es_j.

(1) C chooses a new random value $r^{new} \in_R Z^*_q$, computes $R^{new} = r^{new}P$, $\alpha_{Es_m} = h_1(P_{Es_m}||Q_{Es_m}||pk_{TC})$, $S_{Es_m} = r^{new}(\alpha_{Es_m}Q_{Es_j} + pk_{TC} + P_{Es_m})$ and $\beta_{Es_m} = h_1(R^{new}, S_{Es_m})$, where $m = 1...t$ and $m \neq j$;

(2) C chooses the other random value $\theta^{new} \in_R Z^*_q$ as the secret shared value, computes $pk^{new}_{Es} = \theta^{new}P$ and $f(x) = \prod(x - \beta_{Es_m}) + \theta^{new} \pmod q = x^{t-1} + a'_{t-2}x^{t-2} + ... + a'_1x + a'_0$. Here, $\{a'_0, ..., a'_{t-2}\} \in Z^*_q$;

(3) C stores θ^{new} securely and publishes $\{a'_0, a'_1, ..., a'_{t-2}, R^{new}, pk^{new}_{Es}\}$.

After that, the jth corrupted edge-server cannot obtain the new shared value θ^{new} to encrypt any other messages. It also cannot recover the encrypted critical messages submitted by C.

Adding phase:

Suppose that C adds one edge-server such as Es^{new} in the system, which wants to obtain the secret shared value. C performs the following operations:

(1) C computes $\alpha_{Es^{new}} = h_1(P_{Es^{new}}||Q_{Es^{new}}||pk_{TC})$, $S_{Es^{new}} = r(\alpha_{Es^{new}}Q_{Es^{new}} + pk_{TC} + P_{Es^{new}})$ and $\beta_{Es^{new}} = h_1(R, S_{Es^{new}})$;

(2) C computes $f(x)^{new} = [(f(x)^{new} - \theta)(x - \beta_{Es^{new}}) + \theta] = \prod(x - \beta_{Es_j}) + \theta \pmod q = x^{t+1} + a^{new}_t x^t + a^{new}_{t-1}x^{t-1} + ... + a^{new}_1 x + a^{new}_0$. Here, $a^{new}_0, ..., a^{new}_t \in Z^*_q$;

(3) C publishes $\{a^{new}_0, a^{new}_1, ..., a^{new}_t, R, pk_{Es}\}$.

After that, the new edge-server Es^{new} can obtain the shared value θ to encrypt some messages using θ. It also can recover the encrypted critical messages submitted by C.

Security analysis and proof of NPMA scheme

In this section, we first make a security analysis to prove that the NPMA scheme can satisfy the security requirements aforementioned in the [1, 3, 14]. Second, we further prove that the NPMA scheme is provably secure based on the security model.

Security analysis

Completeness and mutual authentication

In the NPMA scheme, all authentication information $\{M_{i1}, M_{j1}, M_{j2}, M_{c1}\}$ are based on secret values $\{key_{P-E}(= key_{E-P}), key_{E-C}(= key_{C-E})\}$, where

$key_{P-E} = (x_{PA_i} + y_{PA_i} + a_{PA_i})(P_{Es_j} + \alpha_{Es_j}Q_{Es_j} + pk_{TC}) = (x_{PA_i} + y_{PA_i} + a_{PA_i})(x_{Es_j} + \alpha_{Es_j}q_{Es_j} + s)P = (P_{PA_i} + \alpha_{PA_i}Q_{PA_i} + pk_{TC} + A_{PA_i})(x_{Es_j} + y_{Es_j}) = key_{E-P}$;

$key_{E-C} = (\theta + b_{Es_j})(P_c + \alpha_cQ_c + pk_{TC}) = (pk_{Es} + B_{Es_j})(x_c + \alpha_cq_c + s)(pk_{Es} + B_{Es_j})(x_c + t_c) = key_{C-E}$

(Here, $y_{PA_i} = \alpha_{PA_i}q_{PA_i} + s$, $y_{PA_i}P = \alpha_{PA_i}Q_{PA_i} + pk_{TC}$, $y_{Es_j} = \alpha_{Es_j}q_{Es_j} + s$, $y_{Es_j}P = \alpha_{Es_j}Q_{Es_j} + pk_{TC}$, $y_c = \alpha_cq_c + s$ and $y_cP = \alpha_cQ_c + pk_{TC}$)

key_{P-E} is only shared between PA_i and Es_j, which anyone cannot obtain except PA_i and Es_j. key_{E-C} is only shared between Es_j and C, which anyone cannot obtain except Es_j and C $(j = 1...t)$. In the whole scheme, Es_j can authenticate PA_i based on the validation of their published transactions. In addition, PA_i can identify a legitimate Es_j by recomputing key_{P-E}. Similarly, C can

authenticate each Es_j based on the validation of their published transactions. In addition, Es_j can identify a legitimate C by recomputing key_{C-E}. Any probabilistic polynomial time adversary cannot successfully forge a valid key_{P-E}/key_{E-C} for a target message because solving EDLP and ECDHP is computationally hard. Clearly, NPMA scheme achieves mutual authentication between PA_i-Es_j and Es_j-C.

Patient and edge-server anonymity

The NPMA scheme adopts the anonymous blind identities $PID^1_{PA_i} = h_2(key_{P-E}, A_{PA_i}) \oplus pid_{PA_i}||m_{PA_i}$, $PID^2_{PA_i} = h_7(key_{E-P}, T^4_{Es_j}) \oplus pid_{PA_i}||sn$, $PID^1_{Es_j} = h_4(key_{E-C}, B_{Es_j}) \oplus pid_{PA_i}||pid_{Es_j}||m_{PA_i}||M_{i1}$ and $PID^2_{Es_j} = h_6(key_{C-E}, T^2_c) \oplus pid_{PA_i}||pid_{Es_j}||sn$ instead of the static anonym pid_{PA_i} and pid_{Es_j} in the public communication channel. Meanwhile, they are different in each run. From pid_{PA_i}, Es_j cannot obtain anything about the PA_i's true identity ID_{PA_i}. Similarly, based on pid_{Es_j}, C cannot obtain anything about the Es_j's true identity ID_{Es_j}. Here, $pid_{PA_i} = E_s(ID_{PA_i}, \xi_{PA_i})$ and $pid_{Es_j} = E_s(ID_{Es_j}, \xi_{Es_j})$. By using a secure cryptographic symmetric encryption, the malicious adversary $\mathcal{A}$ cannot extract the ID_{PA_i} and ID_{Es_j} without knowing s required to successfully decrypt the ciphertext, further. In this way, the NPMA scheme provides patient and edge-server anonymity, which can prevent the privacy leakage of patient and edge-server.

Patient and edge-server traceability

If Es_j sends some false messages to deceive C, C can computes the unique corresponding parameter of Each edge-server $\alpha_{Es_j} = R_3/(c_1R_1 + c_2R_2)$ by using the managed private key $MSK = (c_1, c_2)$. In addition, if $PA_i(Es_j)$ sends some false messages to deceive $Es_j(C)$, TC also can extract real identity of $PA_i(Es_j)$ by decrypting $pid_{PA_i}(pid_{Es_j})$ using his/her private key s. Hence, the NPMA scheme achieves patient and edge-server traceability to prevent malicious patients and edge-servers from doing something to harm systems.

Perfect forward secrecy

In NPMA scheme, even after an attacker $\mathcal{A}$ has successfully compromised both public/private key pairs, he/she can only reveal the ciphertext belonging to the current shared secret. Since the ephemeral key is different in each shared secret, perfect forward secrecy is guaranteed.

No verifier table

As described above, mutual authentication only requires the patient, edge-server and cloud to generate public/private key pairs without the help of TC. Then, these keys are used to generate messages for authentication. Clearly, there is no verifier table maintained by TC.

Resilience to impersonation attack

If $\mathcal{A}$ can obtain the information $Trans1 = \{PID^1_{PA_i}, A_{PA_i}, M_{i1}, T^1_{PA_i}\}$, $Trans2 = \{PID^1_{Es_j}, M_{j1}, B_{Es_j}, R_1, R_2, R_3, T^2_{Es_j}\}$, $Trans3 = \{PID^2_{Es_j}, M_{c1}, T^2_c\}$ and $Trans4 = \{PID^2_{PA_i}, M_{j2}, T^4_{Es_j}\}$ in public channel. Here, M_{i1} and M_{j2} are only shared between PA_i and Es_j; M_{j1} and M_{c1} are only shared between Es_j and C. Hence, $\mathcal{A}$ can not figure out the valid authentication messages $\{M_{i1}, M_{j1}, M_{c1}, M_{j2}\}$ to pass the authentication. Hence, the NPMA scheme can resist the impersonation attack.

Resilience to internal attacks

Assume that $\mathcal{A}$ is a malicious-legitimate patient (edge-server), $\mathcal{A}$ uses his/her own information in public channel. He/She obtains nothing about other patients' (edge-server') secret information key_{P-E} and key_{E-C}. And he/she also cannot get the random values a_{PA_i} or b_{Es_j}. Hence, he/she cannot succeed in forging authentication information $\{M_{i1}, M_{j1}, M_{c1}, M_{j2}\}$ to pass the authentication. Hence, the NPMA scheme can resist the internal attacks.

Resilience to replay attack

Suppose $\mathcal{A}$ intercepts the $Trans1$, $Trans2$, $Trans3$ and $Trans4$ and replies these messages to corresponding partners. The communicating parties (i.e. PA_i, Es_j) generates new random numbers (a_{PA_i}, b_{Es_j}) in each session, which are involved in request and authentication phase. So that the edge-server or cloud-server can check the freshness of those random numbers. What's more, the freshness of these messages can be determined by the time stamps. Therefore, the NPMA scheme resists replay attack.

Resilience to man-in-the-middle attack

In this attack, $\mathcal{A}$ may try to impersonate a valid patient PA_i, edge-server Es_j or cloud-server C by intercepting the message. However, in the NPMA scheme the secret values key_{P-E} and key_{E-C} are only shared between PA_i and Es_j

and Es_j and C, they will never be discovered by anybody else except these participants. Hence, the NPMA scheme is secure against man-in-the-middle attack.

Security proof

Security model and notations

Formal security model for authentication schemes is mainly developed from Bellare et al. [17]. On the basis of [27, 28], a security proof of NPMA is given under the hardness assumption of **EDLP** and **ECDHP**.

(1) *Participants and initialization*
In NPMA scheme, participants are PA_i, Es_j and C (Here, PA_i-Es_j and Es_j-C share the same authentication method, so we treat PA_i-Es_j as an example. PA_i and Es_j are modeled as a set of random oracles, which not only can be independent, but can be executed concurrently. Each participants has a public/private key pair (sk_{PA_i}, pk_{PA_i}) or (sk_{Es_j}, pk_{Es_j}), where $sk_{PA_i} = (x_{PA_i}, y_{PA_i})$, $pk_{PA_i} = (P_{PA_i}, Q_{PA_i})$, $sk_{Es_j} = (x_{Es_j}, y_{Es_j})$, $pk_{Es_j} = (P_{Es_j}, Q_{Es_j})$.

(2) *Execution of the NPMA scheme*

$\mathcal{S}$ simulates the whole scheme for $\mathcal{A}$, and they interacts each other via the oracle queries. These oracle queries simulate the $\mathcal{A}$'s capabilities in a real attack. The following lists all possible oracle queries:

Execute(PA_i, Es_j): Execute oracle query simulates $\mathcal{A}$'s passive eavesdropping attack. The outputs is the messages that were exchanged between PA_i and Es_j during the real execution in NPMA scheme.

Send(PA_i, Es_j, m): Send oracle simulates $\mathcal{A}$'s active attack. $\mathcal{A}$ sends a message m to PA_i/Es_j. PA_i/Es_j provides the corresponding outputs of m according to the NPMA scheme.

Reveal(PA_i, Es_j): Reveal oracle query can simulate the known key attack. After this query, $\mathcal{A}$ can obtain the negotiated shared secret key between PA_i and Es_j, which helps $\mathcal{A}$ to judge whether two shared secret keys are independent.

Corrupt(PA_i, Es_j, a): Corrupt oracle query simulates $\mathcal{A}$'s corruption capability. By using this query, we can simulate the key compromise impersonation attack.

- ***Corrupt***(PA_i, a): If $a = 1$, it outputs the PA_i's secret value key; If $a = 2$, it outputs PA_i's extemporaneous private key and partial private key.
- ***Corrupt***(Es_j, a): If $a = 1$, it outputs the Es_j's secret value key; If $a = 2$, it outputs Es_j's extemporaneous private key and partial private key.

Ephemeral key reveal(PA_i, Es_j): Ephemeral key reveal oracle simulates the key leak attack. By using this query, $\mathcal{A}$ can obtain some temporary secret information of PA_i/Es_j. However, it can be asked only once.

Test(PA_i, Es_j): Test oracle query can not be used to simulate $\mathcal{A}$'s attack, but it is used to define the semantic security of the shared secret key. Upon querying it, a value will be returned according to a predefined random bit b. If $b = 1$, $\mathcal{A}$ would obtain the shared secret key between PA_i and Es_j. Otherwise, $\mathcal{A}$ only obtains a random same length value. It also can be asked only once.

(3) *Security goals*
sid is the message sets sent and received between PA_i and Es_j in NPMA scheme. PA_i/Es_j has been accepted if the shared secret key has been negotiated successfully .

Partner (Par): Let be $PA_i \in Clients$ and $Es_j \in Server$. The instances PA_i and Es_j are partnered if they meet the following conditions: (1) Both PA_i and Es_j are accepted; (2) Both PA_i and Es_j share the same sid; (3)$Par_{PA_i} = Es_j$ and $Par_{Es_j} = PA_i$.

Freshness: PA_i and Es_j are fresh if they meet the following conditions: (1) PA_i/Es_j has accepted and has shared the same secret key key_{P-E}; (2) Each of them has not been made $Revealqueries$. (3) PA_i/Es_j is asked $Corruptqueries$ at most only once.

Semantic security: The significant goal of authentication schemes is semantic security. During one session of the NPMA scheme, $\mathcal{A}$ can do polynomial times with $Execute, Send, Reveal, Corruptqueries$, a single $Testquery$ for some fresh instances which have been completed. The output of $Testquery$ is a bit b'. Then, $\mathcal{S}$ compares b' with b, which was selected in $Testquery$. If $b' = b$, $\mathcal{A}$ wins the game and $Succ$ stands for this case. Accordingly, $\mathcal{A}$'s advantages to destroy the semantic security is:

$$Adv_{NPMA}^{AP}(\mathcal{A}) \underset{=}{\text{def}} 2Pr[Succ(\mathcal{A})] - 1 = 2Pr[b' = b] - 1$$

Security proof

Before starting to prove, we recall **ECDHP** on which the security proof relies.

ECDHP Assumption: Let G_p be a q-order additive group with a generator P, $a, b \in Z_q^*$ and $aP, bP \in G_p$. Let $\mathcal{A}$ be a **ECDHP**-adversary with running time at most T. $Adv_{G_p}^{ECDHP}(\mathcal{A})$ denotes the probability that $\mathcal{A}$ succeeds in computing abP from aP and bP by $Adv_{G_p}^{ECDHP}(T) = \max_{\mathcal{A}}\{Adv_{G_p}^{ECDHP}(\mathcal{A})\}$, where the maximum is taken over all the adversaries with the running time at most T.

Theorem *Let G_p be a q-order additive group and let $\mathcal{A}$ be an adversary against the semantic security in time*

bound T by less than q_s sessions, q_d Sendqueries, q_e Executequeries, and q_h Hashqueries. Then we have:

$$\begin{aligned} Adv_{NPMA}(\mathcal{A}) \leq\ & \frac{14(q_h)^2}{2^{\lambda+1}} + \frac{4(q_d+q_e)^2}{2(q-1)} + \frac{2q_d}{2^{\lambda}} + \frac{2q_d}{2^{2\lambda}} \\ & + \frac{2q_d}{(q-1)^2} + \frac{4q_d}{(q-1)^2} Adv_{G_p}^{ECDHP}(T) \\ & + \frac{12q_d}{q-1} Adv_{G_p}^{ECDHP}(T) \\ & + 5q_s q_h Adv_{G_p}^{EDLP}(T + 2(q_d+q_e)t_m). \end{aligned}$$

where t_m demotes the modular multiplication time in G_p.

Proof The idea of subject proof is that if $\mathcal{A}$ breaks the semantic security of NPMA scheme successfully, $\mathcal{S}$ can solve **ECDHP** from $\mathcal{A}$'s answers. $\mathcal{S}$ simulates the whole NPMA scheme for $\mathcal{A}$. Our proof defines a sequence of hybrid games, starting with the real attack and ending with a game where $\mathcal{A}$ has no advantage. For each $Game_n$, we define events $Succ_n$ as corresponding case where $\mathcal{A}$ guesses the bit b involved in the $Testquery$, correctly. $AskH_n$ denotes that $\mathcal{A}$ successfully calculates the secret value $key_{P-E}(= key_{E-P})$ by querying h_2 with $\{(key_{P-E}(= key_{E-P}), A_{PA_i})\}$, querying h_3 with $\{pid_{PA_i}, PID^1_{PA_i}, key_{P-E}(= key_{E-P}), A_{PA_i}, m_{PA_i}, T^1_{PA_i}\}$ and $\{sn, pid_{PA_i}, PID^2_{PA_i}, key_{P-E}(= key_{E-P}), M_{i1}, T^4_{Es_j}\}$, querying h_7 with $\{key_{E-P}(= key_{E-P}), T^4_{Es_j}\}$, correspondingly. Here, $key_{P-E} = (x_{PA_i} + y_{PA_i} + a_{PA_i})(P_{Es_j} + \alpha_{Es_j} Q_{Es_j} + pk_{TC}) = (P_{PA_i} + \alpha_{PA_i} Q_{PA_i} + pk_{TC} + A_{PA_i})(x_{Es_j} + y_{Es_j}) = key_{E-P}$; □

$Game_0$: $Game_0$ is the real attack to NPMA scheme without any limits, where $Succ_0 = \{b' = b\}$. According to the definition, we obtain:

$$Adv_{NPMA}(\mathcal{A}) = 2Pr[Succ_0] - 1 \tag{1}$$

$Game_1$: In $Game_1$, $\mathcal{S}$ simulates the random oracles h_2, h_3 and h_7 keeps three hash lists L_h^2, L_h^3 and L_h^7. h_1 only be used to calculate parameter.

- Initially, the list of $L_h^{\varpi} = (\varpi, i, x, y)$ is empty, where $\varpi = (2, 3, 7)$.
- Once, $\mathcal{A}$ issues a query with x, the same answer y in L_h^{ϖ} will be given if the request has been asked before. Otherwise, $\mathcal{S}$ chooses $y \in \{0,1\}^{\lambda}$ as an answer to $\mathcal{A}$, adds this new record (ϖ, i, x, y) to L_h^{ϖ}, where i is the query times, x is the request content set, y is the corresponding answer set. The $Execute$, $Reveal$, $Send$, $Corrupt$, $Test$ oracles are also simulated as real attack. Here, since $\mathcal{S}$ only dose the relevant records in $Game_1$. Compared with $Game_0$, it can easily see that $Game_1$ is perfectly indistinguishable from $Game_0$. Hence,

$$Pr[Succ_1] - Pr[Succ_0] = 0 \tag{2}$$

$Game_2$: In $Game_2$, $\mathcal{S}$ simulates all oracles as in $Game_1$. In order to easily analyze the following games, the possible collisions are considered, firstly. All executions will be terminated if a collision occurs in the massage from $Trans1$ and $Trans4$. Based on the birthday paradox, the collision probability in the output of hash oracle is at most $\frac{14(q_h)^2}{2^{\lambda+1}}$. Similarly, the collision probability of the messages from $Trans1$ and $Trans4$ is at most $\frac{4(q_d+q_e)^2}{2(q-1)}$. Hence,

$$Pr[Succ_2] - Pr[Succ_1] \leq \frac{14(q_h)^2}{2^{\lambda+1}} + \frac{4(q_d+q_e)^2}{2(q-1)} \tag{3}$$

$Game_3$: In $Game_3$, the executions are finished if $\mathcal{A}$ luckily guesses the authentication values M_{i1} or M_{j2}, with no hash queries. $Game_2$ has removed the collision possibility, and the guessed value of $\mathcal{A}$ is exactly the original value with the probability $\frac{2q_d}{2^{\lambda}} + \frac{2q_d}{2^{2\lambda}}$. Hence, $Game_3$ and $Game_2$ are indistinguishable, hence,

$$Pr[Succ_3] - Pr[Succ_2] \leq \frac{2q_d}{2^{\lambda}} + \frac{2q_d}{2^{2\lambda}} \tag{4}$$

$Game_4$: In $Game_4$, the executions are halted if $\mathcal{A}$ has luckily guessed the pairs (sk_{PA_i}, a_{PA_i}) or (sk_{Es_j}, b_{Es_j}) with no hash queries, and spoofed the *Clients* and *Server* successfully. Hence, $Game_4$ and $Game_3$ are indistinguishable, therefore,

$$Pr[Succ_4] - Pr[Succ_3] \leq \frac{2q_d}{(q-1)^2} \tag{5}$$

$Game_5$: In $Game_5$, the executions are terminated if $\mathcal{A}$ computes the secret parameters $(x_{PA_i} + y_{PA_i} + a_{PA_i})(P_{Es_j} + \alpha_{Es_j} Q_{Es_j} + pk_{TC})$ or $(P_{PA_i} + \alpha_{PA_i} Q_{PA_i} + pk_{TC} + A_{PA_i})(x_{Es_j} + y_{Es_j})$ via querying hash oracle, which is the unique method to compute $key_{P-E}(= key_{E-P})$. If the event $AskH_5$ does not happen, $Game_5$ and $Game_4$ are indistinguishable. Hence,

$$Pr[Succ_5] - Pr[Succ_4] \leq Pr[AskH_5] \tag{6}$$

Based on security model, $\mathcal{A}$ not only can query $Corrupt(PA_i, a)$, but also can query $Corrupt(Es_j, a)$. Hence:

$$\begin{aligned} & \Pr[AskH_5 WithCorrupt(PA_i, 1)] \\ =\ & \Pr[(x_{PA_i} + * + *)(P_{Es_j} + \alpha_{Es_j} Q_{Es_j} + pk_{TC})] \\ \leq\ & \frac{4q_d}{(q-1)^2} Adv_{G_p}^{ECDHP}(T) + 2q_s q_h Adv_{G_p}^{EDLP} \\ & \times (T + 2(q_d+q_e)t_m) \\ & \Pr[AskH_5 WithCorrupt(PA_i, 2)] \\ =\ & \Pr[(* + y_{PA_i} + a_{PA_i})(P_{Es_j} + \alpha_{Es_j} Q_{Es_j} + pk_{TC})] \\ \leq\ & \frac{4q_d}{q-1} Adv_{G_p}^{ECDHP}(T) + q_s q_h Adv_{G_p}^{EDLP} \\ & \times (T + 2(q_d+q_e)t_m) \end{aligned}$$

$$
\begin{aligned}
&\Pr[AskH_5WithCorrupt(Es_j,1)]\\
&=\Pr[(P_{PA_i}+\alpha_{PA_i}Q_{PA_i}+pk_{TC}+A_{PA_i})(x_{Es_j}+*)]\\
&\le \frac{4q_d}{q-1}Adv^{ECDHP}_{G_p}(T)+q_sq_hAdv^{EDLP}_{G_p}\\
&\quad\times(T+2(q_d+q_e)t_m)\\
&\Pr[AskH_5WithCorrupt(Es_j,2)]\\
&=\Pr[(P_{PA_i}+\alpha_{PA_i}Q_{PA_i}+pk_{TC}+A_{PA_i})(*+y_{Es_j})]\\
&\le \frac{4q_d}{q-1}Adv^{ECDHP}_{G_p}(T)+q_sq_hAdv^{EDLP}_{G_p}\\
&\quad\times(T+2(q_d+q_e)t_m)
\end{aligned}
$$

Note: Here, we prove $\Pr[AskH_5]$ is restrained as follows:

$$
\begin{aligned}
\Pr[AskH_5] =\ &\Pr[AskH_5WithCorrupt(PA_i,1)]\\
&+\Pr[AskH_5WithCorrupt(PA_i,2)]\\
&+\Pr[AskH_5WithCorrupt(Es_j,1)]\\
&+\Pr[AskH_5WithCorrupt(Es_j,2)]\\
\le\ &\frac{4q_d}{(q-1)^2}Adv^{ECDHP}_{G_p}(T)\\
&+\frac{12q_d}{q-1}Adv^{ECDHP}_{G_p}(T)\\
&+5q_sq_hAdv^{ECDHP}_{G_p}(T+2(q_d+q_e)t_m)
\end{aligned}
$$

Clearly, they are based on the **ECDHP**. In order to simplify the proof, we set $\Pr[AskH_5WithCorrupt(PA_i,1)]$ as an example of them:

$\mathcal{A}$ gets the account $(x_{PA_i}+*+*)(P_{Es_j}+\alpha_{Es_j}Q_{Es_j}+pk_{TC})$ by $Corrupt(PA_i,1)$, but $\mathcal{A}$ does not know $(y_{PA_i}+a_{PA_i})$. $\mathcal{S}$ sets one session of $[1,2,\cdots,q_s]$ as the test session. If the event $AskH_5$ happens, $\mathcal{S}$ can use $\mathcal{A}$ to solve **ECDHP**.

To non-test sessions, $\mathcal{S}$ responses $\mathcal{A}$ as $Game_4$, when $\mathcal{A}$ queries h_2, h_3 or h_7 with $(x_{PA_i}+y^*_{PA_i}+a^*_{PA_i})(P_{Es_j}+\alpha_{Es_j}Q_{Es_j}+pk_{TC})$ $(\{y^*_{PA_i},a^*_{PA_i}\}\in Z^*_p$ are chosen by $\mathcal{A})$. If Each lists L^{ϖ}_h has the record which contains $(x_{PA_i}+y^*_{PA_i}+a^*_{PA_i})(P_{Es_j}+\alpha_{Es_j}Q_{Es_j}+pk_{TC}),y)$, return y. Or else, $\mathcal{S}$ queries **EDLP** oracle to verify $PID^1_{PA_i}\oplus pid_{PA_i}||m_{PA_i}?=h_2(key_{P-E},A_{PA_i})$ or $PID^2_{PA_i}\oplus pid_{PA_i}||sn?=h_7(key_{E-P},T^4_{Es_j})$. If not equal, return $\perp$; or else, $\mathcal{S}$ chooses two random values $\{y^*_{PA_i},a^*_{PA_i}\}\in Z^*_p$ to $\mathcal{A}$, because $\mathcal{S}$ does not know $\{y_{PA_i},a_{PA_i}\}$, either.

To the test session, $\mathcal{A}$ uses $Ephemeral\ key\ reveal(PA_i,Es_j)$ to obtain one key_{P-E}, but $\mathcal{A}$ should compute other key_{P-E}. $\mathcal{S}$ chooses $aP,bP\in G_p$ randomly, let $Q_{PA_I}=\alpha^{-1}_{PA_I}(aP-Pub_{TA}-P_{PA_I}-A_{PA_I})$ and $Q_{Es_J}=\alpha^{-1}_{Es_J}(bP-Pub_{TA}-P_{Es_J})$. Recall $AskH_5$, when $PA_i=PA_I$ and $Par_{PA_i}=Es_J$, $\mathcal{A}$ queries $\{(x_{PA_I}+y^*_{PA_I}+a^*_{PA_I})(P_{Es_J}+\alpha_{Es_J}Q_{Es_J}+pk_{TC}),A_{PA_I}\}$. Then, $\mathcal{S}$ can get the **ECDHP** solution abP directly. If the session chosen by $\mathcal{A}$ is the same as that $\mathcal{S}$ pre-set in advance, the probability is $\frac{4q_d}{(q-1)^2}$. If the event $AskH_5$ does not happen, $Game_5$ and $Game_4$ are indistinguishable. Hence,

$$
\begin{aligned}
&\Pr[AskH_5WithCorrupt(PA_i,1)]\\
&=\Pr[(x_{PA_i}+*+*)(P_{Es_j}+\alpha_{Es_j}Q_{Es_j}+pk_{TC})]\\
&\le \frac{4q_d}{(q-1)^2}Adv^{ECDHP}_{G_p}(T)+2q_sq_hAdv^{EDLP}_{G_p}\\
&\quad\times(T+2(q_d+q_e)t_m)
\end{aligned}
$$

and in indeed $\Pr[Succ_5]=\frac{1}{2}$.

Hence, it is easy to conclude with:

$$
\begin{aligned}
&Adv_{NPMA}(\mathcal{A})=2\Pr[Succ_0]-1\\
&=2\Pr[Succ_5]-1+2(\Pr[Succ_0]-\Pr[Succ_5])\\
&=2\{(\Pr[Succ_0]-\Pr[Succ_1])+(\Pr[Succ_1]-\Pr[Succ_2])\\
&\quad+(\Pr[Succ_2]-\Pr[Succ_3])+(\Pr[Succ_3]-\Pr[Succ_4])\\
&\quad+(\Pr[Succ_4]-\Pr[Succ_5])\}\\
&\le 2\{|\Pr[Succ_0]-\Pr[Succ_1]|+|\Pr[Succ_1]-\Pr[Succ_2]|\\
&\quad+|\Pr[Succ_2]-\Pr[Succ_3]|+|\Pr[Succ_3]-\Pr[Succ_4]|\\
&\quad+|\Pr[Succ_4]-\Pr[Succ_5]|\}\\
&\le \frac{14(q_h)^2}{2^{\lambda+1}}+\frac{4(q_d+q_e)^2}{2(q-1)}+\frac{2q_d}{2^{\lambda}}+\frac{2q_d}{2^{2\lambda}}\\
&\quad+\frac{2q_d}{(q-1)^2}+\frac{4q_d}{(q-1)^2}\\
&\quad Adv^{ECDHP}_{G_p}(T)+\frac{12q_d}{q-1}Adv^{ECDHP}_{G_p}(T)\\
&\quad+5q_sq_hAdv^{EDLP}_{G_p}(T+2(q_d+q_e)t_m).
\end{aligned}
$$

Thus,we get:

$$
\begin{aligned}
Adv_{NPMA}(\mathcal{A}) \le\ &\frac{14(q_h)^2}{2^{\lambda+1}}+\frac{4(q_d+q_e)^2}{2(q-1)}+\frac{2q_d}{2^{\lambda}}+\frac{2q_d}{2^{2\lambda}}\\
&+\frac{2q_d}{(q-1)^2}+\frac{4q_d}{(q-1)^2}\,Adv^{ECDHP}_{G_p}(T)\\
&+\frac{12q_d}{q-1}Adv^{ECDHP}_{G_p}(T)\\
&+5q_sq_hAdv^{EDLP}_{G_p}(T+2(q_d+q_e)t_m).
\end{aligned}
$$

Table 2 Computational notations

Operation	Times(ms)	Description
t_{bp}	5.427	Runtime of bilinear paring operation in G_p
t_{sm}	2.165	Runtime of scalar multiplication operation in G_p
t_{fe}	2.345	Runtime of a fuzzy extraction operation
t_{ed}	3.85	Runtime of for encryption/decryption
t_{hp}	5.493	Runtime of hash-to-point operation

Table 3 Energy notations

Operation	Energy cost (MICAz sensor)
$bilinearparingoperation$	129mJ/160 bits
$scalarmultiplicationoperation$	55mJ/160 bits
$fuzzyextractionoperation$	64mJ/160 bits
$hashoperationofSHA-1$	5.9μJ/byte
$transmit$	59.2μJ/byte
$receive$	26.9μJ/byte

Performance evaluation

This section, we present the performance analysis of NPMA scheme from three aspects: computation cost, communication cost and energy cost. Moreover, the efficiency comparison between NPMA scheme and the related schemes [2, 14, 17] will be shown in this section.

In our experiments, we choose a q-order group G_p with the generator P, and P is a point selected from non-singular elliptic curve $E(F_p)$: $y^2 = x^3 + 1$. For convenience, we define some notations about the running time and energy cost in Tables 2 and 3 [2, 27–32], respectively. In addition, we also discuss how NPMA scheme is efficient than other related schemes from its implementation point of view later as roughly shown in Fig. 7.

Computation cost

Let t_h, t_c, t_x, t_{bp}, t_{sm}, t_{ed} and t_{hp} denote hash function, concatenation operation, XOR operation, the time complexity for scarlar bilinear paring operation, multiplication operation and hash-to-point operation. Since the time of hash function, concatenation operation and XOR operation are negligible compared to the other four operations, we ignore t_h, t_c and t_x.

Based on the implementation results in [27, 30–32], we analyze and compare the computation cost of related scheme, as shown in Fig. 7a.

Wazid et al. [2] scheme costs two multiplication operations, eighteen hash-to-point operations and one fuzzy

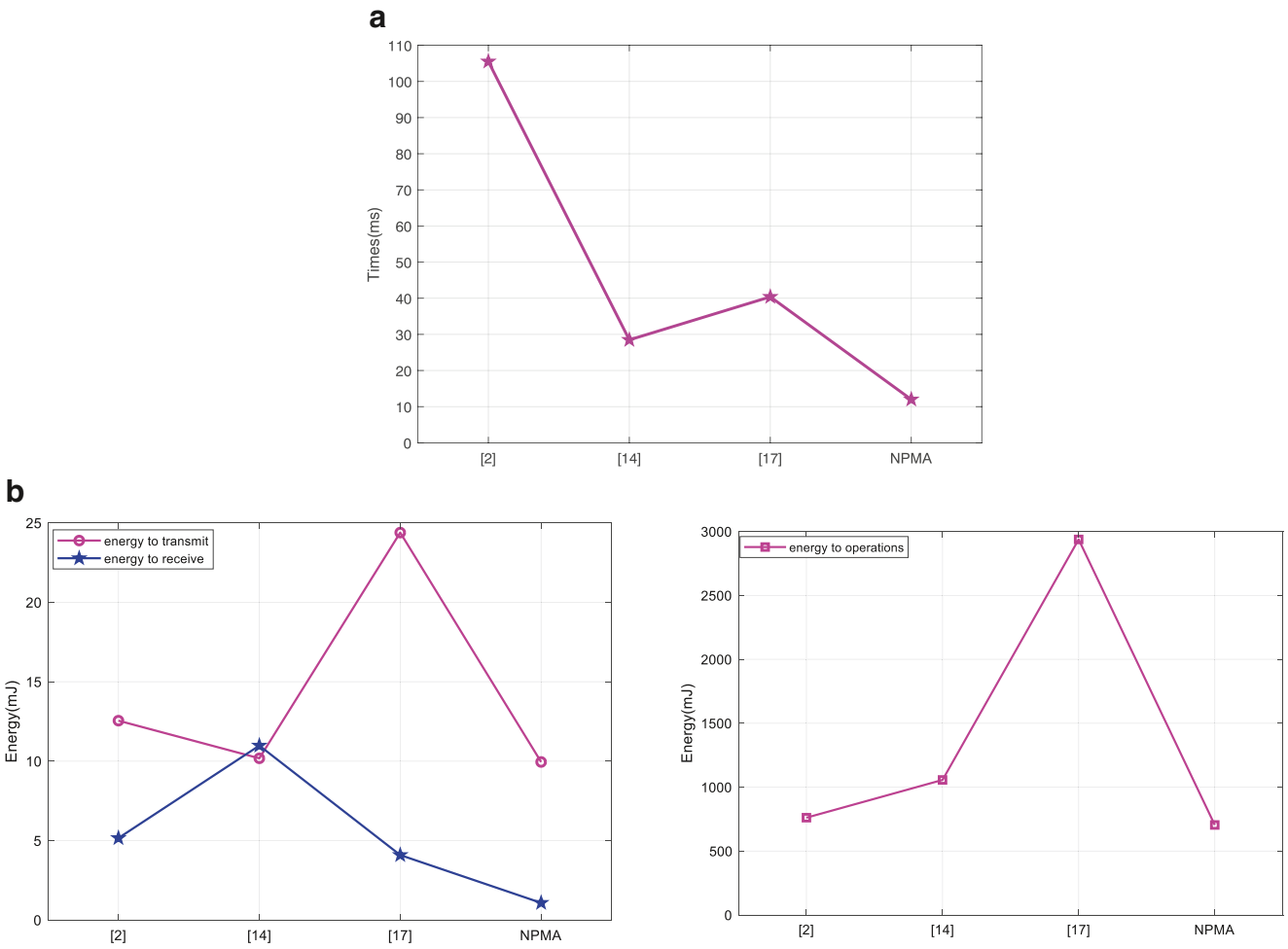

Fig. 7 Performance comparison of related schemes

extraction operation. Therefore, the running time is $2t_{sm} + 18t_{hp} + t_{fe} \approx 105.52ms$.

Ma et al. [14] scheme costs three multiplication operations and four hash-to-point operations. Therefore, the running time is $3t_{sm} + 4t_{hp} \approx 28.48ms$.

Jia et al. [17] scheme costs six multiplication operations, four hash-to-point operations and one scarlar bilinear paring operation. Therefore, the running time is $6t_{sm} + 4t_{hp} + t_{bp} \approx 40.39ms$.

NPMA scheme costs three multiplication operations, one hash-to-point operations. Therefore, the running time is $3t_{sm} + t_{hp} \approx 11.99ms$.

According to the above comparison of computation cost, NPMA scheme has much less running time than other three related schemes [2, 14, 17] in patients' side.

Communication cost

In this subsection, we analyze and compare the communication costs of NPMA scheme and other three related scheme [2, 14, 17]. Without loss of generality, the size of the element in G_p and Z_q^* is 1024 bits and 160 bits, respectively. The length of a patients identity is 32 bits. The length of the output of a hash function is dependent on its domain size. For a regular hash, we assume it to be 256 bits [27, 30–32]. The length of the timestamp is set as 32 bits. The comparisons among related schemes are listed in Table 4.

In Wazid et al. [2] scheme, among the interactive messages, there are three elements in G_p, eleven elements in Z_q^*, and three timestamps. Therefore, the communication cost is 3*1024+11*160+3*32=4928 bits

In Ma et al. [14] scheme, among the interactive messages, there are ten elements in G_p, nine elements in Z_q^*, and five timestamps. Therefore, the communication cost is 10*1024+9*160+5*32=11840 bits.

In Jia et al. [17] scheme, among the interactive messages, there are four elements in G_p, two elements in Z_q^*, one identity and two timestamps. Therefore, the communication cost is 4*1024+2*160+32+2*32=4512 bits.

In NPMA scheme, among the interactive messages, there are five elements in G_p, four identities, four hash values and four timestamps. Therefore, the communication cost is 5*1024+4*32+4*256+4*32=6400 bits.

Table 4 Performance comparison among relevant authentication schemes

	Wazid et al. [2]	Ma et al. [14]	Jia et al. [17]	NPMA
communication cost/bit	4928	11840	4512	6400

Energy cost

In mobile devices, energy-saving is an important indicator. Here, we only discuss the patients' side from three part: energy to transmit, energy to receive and energy to operations, as shown in Fig. 7b.

Wazid et al. [2] scheme needs to transmit one element in G_p, four elements in Z_q^*, and one timestamp, total 1696 bits. According to [28–32], it costs 12.55 mJ. It receives one element in G_p, three elements in Z_q^*, and one timestamp, total 1536 bits, which costs 5.16 mJ. The operations are two multiplication operations, eighteen hash-to-point operations and one fuzzy extraction operation, which cost 761.12 mJ.

Ma et al. [14] scheme needs to transmit one element in G_p, two elements in Z_q^*, and one timestamp, total 1376 bits. which costs 10.18 mJ. It receives three elements in G_p, one element in Z_q^*, and one timestamp, total 3264 bits, which costs 10.98 mJ. The operations are three multiplication operations and four hash-to-point operations, which cost 1056.47 mJ.

Jia et al. [17] scheme needs to transmit three elements in G_p, one element in Z_q^*, one identity and one timestamp, total 3296 bits. which costs 24.39 mJ. It receives one element in G_p, one element in Z_q^*, and one timestamp, total 1216 bits, which costs 4.09 mJ. The operations are six multiplication operations, four hash-to-point operations and one scalar bilinear paring operation, which cost 2938.07 mJ.

NPMA scheme needs to transmit one element in G_p, one identity, one hash value and one timestamp, total 1344 bits. which costs 9.95 mJ. It receives one identity, one hash

Table 5 Security features comparison among relevant authentication schemes

	Wazid et al. [2]	Ma et al. [14]	Jia et al. [17]	NPMA
Mutual authentication	Yes	Yes	Yes	Yes
Patient anonymity	Yes	No	No	Yes
Patient traceability	No	No	No	Yes
Server traceability	No	No	No	Yes
Message confidentiality	Yes	Yes	Yes	Yes
Resistance to impersonation attack	Yes	Yes	Yes	Yes
Resistance to internal attacks	Yes	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes
Resistance to man-in-the-middle attack	Yes	Yes	Yes	Yes
Provable security	Yes	Yes	Yes	Yes
Response method	No	No	No	Yes

value and one timestamp, total 320 bits, which costs 1.08 mJ. The operations are three multiplication operations, one hash-to-point operations, which cost 704.12 mJ.

According to the above comparisons of energy cost, we know that the NPMA scheme also is energy-saving compared with the other three related schemes [2, 14, 17].

Security comparisons

To show the security advantages of NPMA scheme, we present security comparisons between NPMA and other three related ones [2, 14, 17]. From Table 5, we obtain that NPMA scheme can satisfy above security and function requirements. Therefore, NPMA scheme is more secure than other three related ones.

Conclusion and ongiong work

TMIS is not a far fetched concept, and the complexity of future TMIS will require a more robust security solution. In order to establish a secure remote user authentication for mobile Edge-Cloud architecture, we proposed a novel privacy-preserving mutual authentication in this paper (NPMA). The NPMA scheme leverages the underpinning characteristics of edge-cloud computing to realize a decentralized and privacy-preserving solution in TMIS. Specifically, we utilized double anonyms (one is used to anonymously authenticate patients (or edge servers), the other is used to trace malicious patients (or edge servers) under necessary), shared value (to efficiently authenticate gateways in few encryption times), and certificateless cryptography (to provide confidentiality of the requested messages). We then demonstrated the security of NPMA and evaluated the performance of the prototype, theoretically. The results show that NPMA scheme is very suitable for computation-limited mobile devices compared with other related existing schemes. The future research is to fully identify the practical threats on MEC-based authentication schemes with better performance and evaluate the their performance using software and hardware.

Funding This work is partially supported by National Key R&D Program of China No. 2017YFB0802400, the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University No.5001-20109195456, National Science Foundation of China under grant No. 61373171, The 111 Project under grant No. B08038, the Program for Excellent Young Talents in University of Anhui Province under Grant No. gxyqZD2019060.

Compliance with Ethical Standards

Conflict of interests Author Xiaoxue Liu declares that she has no conflict of interest. Author Wenping Ma declares that he has no conflict of interest. Author Hao Cao declares that she has no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Yang, Y., Zheng, X., and Tang, C., Lightweight distributed secure data management system for health internet of things[J]. *J. Netw. Comput. Appl.* 89:26–37, 2017.
2. Wazid, M., Das, A., Kumar, N. et al., Design of secure key management and user authentication scheme for fog computing services[J]. *Futur. Gener. Comput. Syst.* 91:475–492, 2019.
3. Tran, T., Hajisami, A., Pandey, P. et al., Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges[J]. *IEEE Commun. Mag.* 91(4):54C61, 2017.
4. Jiang, Q., Qian, Y., Ma, J. et al., User centric three-factor authentication protocol for cloud-assisted wearable devices[J]. *Int. J. Commun. Syst.* 32(6):1–20, 2019.
5. Jiang, Q., Ma, J., Yang, C. et al., Efficient end-to-end authentication protocol for wearable health monitoring systems[J]. *Comput. Electr. Eng.* 63:182–195, 2017.
6. Li, X., Peng, J., Niu, J. et al., A robust and energy efficient authentication protocol for industrial internet of things[J]. *IEEE Internet Things J.* 5(3):1606–1615, 2017.
7. Liu, X., and Ma, W., ETAP: energy-efficient and traceable authentication protocol in mobile medical cloud architecture[J]. *IEEE Access* 6:33513–33528, 2018.
8. Prasser, F., Kohlmayer, F., Spengler, H. et al., A scalable and pragmatic method for the safe sharing of high-quality health data[J]. *IEEE J. Biomed. Health Inform.* 22(2):611–622, 2017.
9. Bonomi, F., Milito, R., Zhu, J. et al., Fog computing and its role in the Internet of things. In: *Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13–16, 2012.
10. Hu, P., Dhelim, S., Ning, H. et al., Survey on fog computing: architecture, key technologies, applications and open issues. *J. Netw. Comput. Appl.* 98:27–42, 2017.
11. Stojmenovic, I., and Wen, S., The fog computing paradigm: scenarios and security issues[C]. In: *Federated Conference on Computer Science and Information Systems, Prague, Czech Republic*, pp. 1–8, 2014.
12. Koo, D., and Hur, J., Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing[J]. *Futur. Gener. Comput. Syst.* 78:739–752, 2018.
13. Wang, H., Wang, Z., and Domingo-Ferrer, J., Anonymous and secure aggregation scheme in fog-based public cloud computing[J]. *Futur. Gener. Comput. Syst.* 78:712–719, 2018.
14. Ma, M., He, D., Wang, H. et al., An efficient and provably-secure authenticated key agreement protocol for fog-based vehicular Ad-Hoc networks[J]. IEEE Int. Things Journal (2019 Early Access).
15. IBM News Releases, IBM and Nokia Siemens Networks announce world first mobile edge computing platform, 2013.
16. Zhang, Y., Lopez, J., and Wang, Z., Mobile edge computing for vehicular networks[J]. *IEEE Veh. Technol. Mag.* 14(1):27–108, 2019.
17. Jia, X., He, D., Kumar, N. et al., A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing[J]. IEEE Syst. J. (2019 Early Access).
18. Li, X., Liu, S., Wu, F. et al., Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications[J]. *IEEE Internet Things J.* 6(3):4755–4763, 2019.
19. Sodhro, A., Luo, Z., Sangaiah, A. et al., Mobile edge computing based QoS optimization in medical healthcare applications[J]. *Int. J. Inf. Manag.* 45(1):308–318, 2019.

20. Abdellatif, A., Mohamed, A., Chiasserini, C. et al., Edge computing for smart health: context-aware approaches, opportunities, and challenges[J]. IEEE Netw., 2019.
21. Aghili, S., Mala, H., Shojafar, M. et al., LACO: lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT[J]. *Futur. Gener. Comput. Syst.* 96:410–424, 2019.
22. Renuka, K., Kumari, S., and Li, X., Design of a secure three-factor authentication scheme for smart healthcare[J]. *J. Med. Syst.* 43(5):133–143, 2019.
23. Tang, W., Zhang, K., Ren, J. et al., Flexible and efficient authenticated key agreement scheme for bans based on physiological features[J]. *IEEE Trans. Mobile Comput.* 18(4):845–856, 2019.
24. Nguyen, D., Pathirana, P., Ding, M. et al., Blockchain for secure EHRs sharing of mobile cloud based e-health systems[J]. IEEE Access (2019 Early Access).
25. Li, X., Peng, J., Obaidat, M. et al., A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems[J]. IEEE Systems J. (2019 Early Access).
26. Wazid, M., Das, A., Kumar, N. et al., A novel authentication and key agreement scheme for implantable medical devices deployment[J]. *IEEE J. Biomed. Health Inform.* 22(4):1299–1309, 2017.
27. He, D., Ma, M., Zeadally, S. et al., Certificateless public key authenticated encryption with keyword search for industrial internet of things[J]. *IEEE Trans. Ind. Inf.* 14(8):3618–3627, 2017.
28. Wander, A., Gura, N., Eberle, H. et al., Energy analysis of public-key cryptography for wireless sensor networks[C]. In: *Third IEEE International Conference on Pervasive Computing and Communications, PerCom*, 2005.
29. Meulenaer, D., Gosset, F., Standaert, F. et al., On the energy cost of communication and cryptography in wireless sensor networks[C]. Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing IEEE, pp. 580–585, 2008.
30. Liu, X., and Ma, W., CDAKA: a provably-secure heterogeneous cross-domain authenticated key agreement protocol with symptoms-matching in TMIS[J]. *J. Medical Syst.* 42(8):135–147, 2018.
31. Odelu, V., Saha, S., Prasath, R. et al., Efficient privacy preserving device authentication in WBANs for industrial e-health applications[J]. *Comput. Secur.* 83:300–312, 2019.
32. Wang, D., and Wang, P., Two birds with one stone: two-factor authentication with security beyond conventional bound[J]. *IEEE Trans. Dependable Secure Comput.* 15(4):708–722, 2016.