J Med Syst (2018) 42: 135
https://doi.org/10.1007/s10916-018-0985-7

MOBILE & WIRELESS HEALTH

CDAKA: A Provably-Secure Heterogeneous Cross-Domain Authenticated Key Agreement Protocol with Symptoms-Matching in TMIS

Xiaoxue Liu[1] **· Wenping Ma**[1]

Received: 12 March 2018 / Accepted: 1 June 2018 / Published online: 18 June 2018

Abstract

Telecare Medical Information System (TMIS) provides the flexible and convenient e-health care. It helps the patients to gain health monitoring information and provides patients to share their experience wirelessly. Traditional authentication and key agreement (AKA) protocols in TMIS are mostly considered in same-domain environment. However, future generation network may integrate various of wireless mesh networks under various domain. What's more, patients heterogeneous cross-domain service has become an inevitable trend. However, there is still no heterogeneous cross-domain authenticated protocol between PKI-domain and IBC-domain in TMIS. In this paper, we propose a heterogeneous cross-domain AKA protocol with symptoms-matching in TMIS (short for CDAKA). It not only keeps good security features, but also truly provides patients' anonymity to protect sensitive information from illegal interception. It still provides patients in two different domains to share their experience, broaden their understanding of illness by using their mobile device freely. Besides, it can realize AKA with extremely low computing cost and communication cost. What's more, it is proved to be secure against known possible attacks under the Elliptic Curve Computable Diffie-Hellman problem (ECDHP) assumption in the random oracle model. Hence, these features make CDAKA protocol very suitable for mobile application scenarios, where resource is severely constrained and security is particularly concerned.

Keywords Anonymity · Heterogeneous cross-domain · Provably-secure · Authenticated key agreement · Symptoms-matching

Introduction

Aging is a universal phenomenon affecting all countries, although its dynamic can be different in each. According to the lasted census report, the population of the world is on the trend of aging rapidly, where more than 12.3% of the world's population are over 60 years old, partly due to a longer life span but declining birth rate. It is predicted that the population over 60 will exceed the ones under 15 by 2050. Further, the incidence of mortality rate among the elderly people is much higher than non-elderly ones [1]. Meanwhile, elderly people are suffering from different chronic conditions and disability. Hence, using health care services will be a necessity of life. For example, the US will need to hire 2.3 million new health care workers by 2025 in order to adequately take care of its aging population, a new report finds (http://money.cnn.com/2018/05/04/news/economy/health-care-workers-shortage/index.html). It is no doubt that the demand of medical service is increasing, not just for the US, but the world.

The rapid development of mobile Internet has greatly changed our daily life, especially in Telecare Medical Information System (TMIS). In TMIS, the patients can receive professionals symptom diagnosis from the health care providers to direct their treatment. On the other hand, these patients also have the intention to communicate with other patients who have the same symptom. Then, they want to build a symptom-matching based communication to

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

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

facilitate the illness-related information exchange, treatment experience sharing and specialist doctor recommendation. Besides, they may chat with each other to talk about their real-time illness conditions and encourage each other to overcome the disease, regardless of the patients' locations and conditions. Sometimes, self-confidence is more effective than drugs in patients' conditions.

Traditional authentication and key agreement (AKA) protocols in TMIS are mostly considered in same-domain environment. In same-domain environment (Public Key Infrastructure-PKI or Identity Based Cryptography-IBC), several session keys can be easily established by real world meeting. They actually have to stay in the same hospital for the treatment. However, no opportunity is offered for them to meet in the real life. Actually, the patients are always physically affiliated to different medical domains. Those with the same symptoms most likely come from various medical institutions in different cities or even different countries. What's more, the patients with some rare diseases could hardly find the fellow sufferers in the same area [2]. Let's consider a scenario, as shown in Fig. 1b. A patient PA_i in PKI-domain needs to share his/her treatment experience with the other patient PA_j in IBC-domain multiple times in a short time and requests a secure communication service. Additionally, PA_i needs to communicate with the trusted authority (IBC) in IBC-domain, which further communicates with the trusted authority PKI. As a consequence, trusted authorities PKI and IBC will easily become the bottleneck of the system. The involvement of the trusted authorities in both domains also increases the authentication delay. Hence, they do not reach the case. (The CDAKA protocol is even simpler for this case on each short time, as shown in Fig. 1c.) Therefore, it is very important and urgent to design heterogeneous cross-domain authenticated key agreement mechanism to provide the interactions with different domain patients.

On the other hand, most of medical datas are transmitted and exposed during the unsecured-public communication channel, the patient's privacy is susceptible to be divulged. Most patients in TMIS are connected with each other wirelessly. Then, the adversaries may eavesdrop, intercept, delete, and modify all messages in the common communication channel. Hence, it is easily overlooked when the origin of data is traced. Specifically, the patients locations, jobs, and home addresses can be acquired and the habits and tastes can be derived immediately. It also largely reduces the difficulty of guessing the patients' real identities. When the least expected thing happens, unauthorized adversaries may get access to the patients current health condition, medical history and other binding information like mobile phone number and credit card number. Undoubtedly, the patient will suffer much more than the illness itself. Considering the worst condition, if the adversary has an attempt at harming the patient, he may modify the patients vital health information. And when these modified messages are transmitted to their sick friends, wrong information can be made and the patients life may be threatened [3, 4]. Obviously, patient's privacy protection has not been adequately addressed and it is still an urgent demanding in medical environment. The protocols designed for TMIS should take patients' privacy-protection into account.

Our contributions

In the CDAKA protocol, the patient PA_i in PKI-domain can remotely communicate with the other patient PA_j who is in IBC-domain by themselves without the help of their registration centers. It perfectly eliminates the bottlenecks of systems. Hereafter, PA_i and PA_j not only realize mutual authentication but also establish a session key. Compared with [5, 6], the CDAKA protocol not only needs lower computational consumption, but also can provide the following security features.

- First, the CDAKA protocol not only can provide patient's anonymity to protect patient's privacy by randomized-dual pseudonym $PID_i(PID_j)$, but also can provide patient's traceability if necessary. When a patient $PA_i(PA_j)$ sends the false messages to deceive others, $PKI(IBC)$ extracts PA_i's(PA_j's) static anonymous identity $pid_i(pid_j)$ from randomized-dual pseudonym $PID_i(PID_j)$ and obtains PA_i's(PA_j's) real identity by decrypting pid_i using its private key. Besides, the register center PKI in PKI-domain and register center IBC in IBC-domain, no one can obtain the others' real identities. Hence, the CDAKA protocol is practical in the privacy-enhanced scenarios.
- Second, the CDAKA protocol can truly realize heterogeneous cross-domain authentication and obtain the session key among the mobile terminal patients in different remote medical domains. The entire process only costs two-round communications with low computation cost and communication cost. Hence, the CDAKA protocol is very simple, efficient and energy-saving and it is very suitable for computation-limited mobile devices.
- Third, the CDAKA protocol based on certificateless cryptography can overcome the key escrow problem of identity-based public key cryptography. The patients' full private keys consist of two parts: the secret information chosen by patients themselves and the partial private keys generated by registration centers. It properly resolves the complicated certificate management problems in traditional public key infrastructure system.
- Fourth, the CDAKA protocol is proved to be secure under the Elliptic Curve Computational Diffie-Hellman problem (ECDHP) assumption in the random oracle

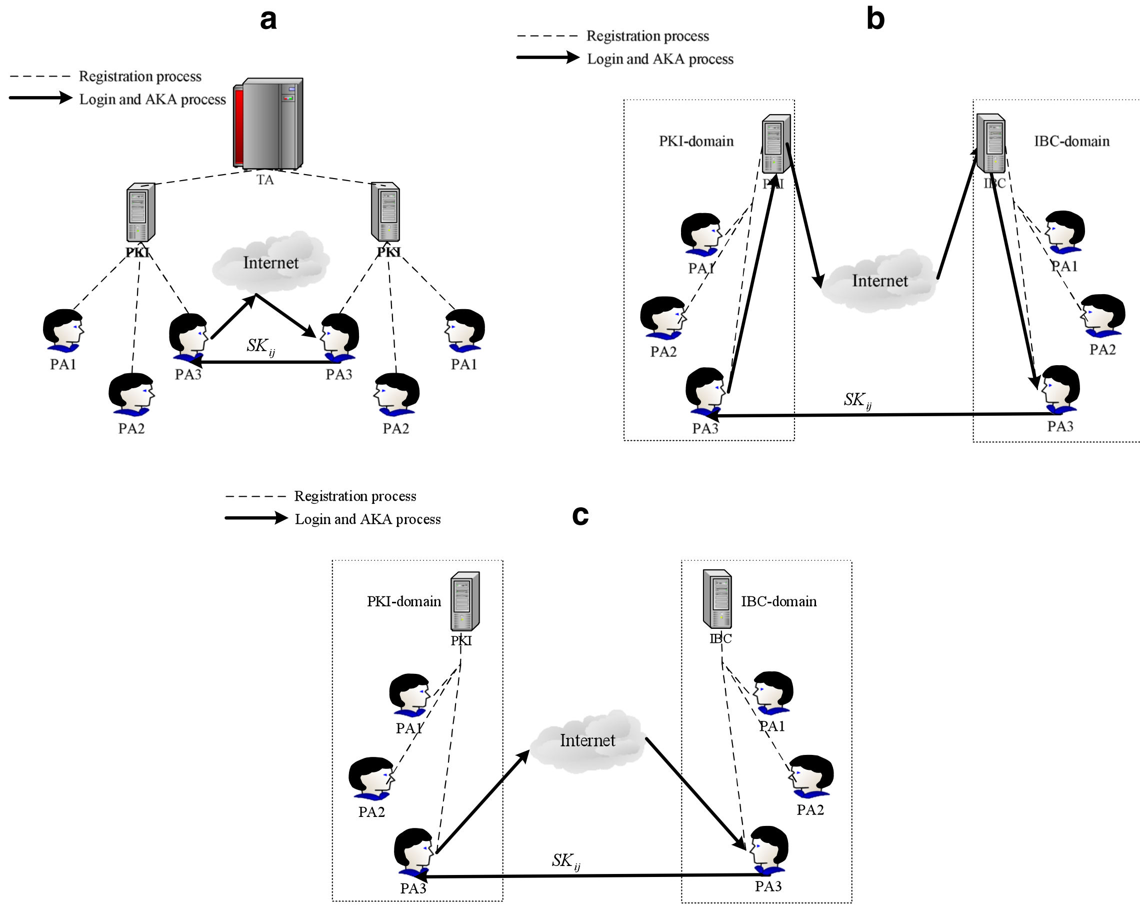


Fig. 1 Architecture for accessing cross-domain medical service in TMIS

model. The CDAKA protocol is proved secure against possible known attacks and satisfy the secure requirements of AKA protocols for heterogeneous cross-domain architecture. Hence, the CDAKA protocol is practical in complex network environment.

Related works

For better efficiency and accuracy, authentication has become an essential mechanism to assure the distributed systems' security and privacy from malicious adversaries. Due to the widespread applications of Internet and the great convenience of remote medical services, how to securely access the remote medical servers and get the corresponding service has received considerable attention. In recent years, various remote AKA protocols are successively proposed in TMIS [7–13].

Wu et al. [7] first proposed a novel authentication protocol for TMIS. However, it was vulnerable to insider attack and impersonation attack [8]. Later, Wei et al. [9] pointed out that the both protocols in [7] and [8] failed to meet multi-factor authentication and further proposed an improved protocol. Thereafter, Zhu et al. [10] described Wei et al.'s protocol [9] was vulnerable to off-line password guessing attack. Then, Lee-Liu [11] demonstrated that the new protocol in [10] could not withstand parallel session attack and presented an improved one. In 2013, Tan et al. [12] proposed an efficient biometrics-based authentication scheme for TMIS, which was claimed to resist many kinds of attacks. However, Yan et al. [13] declared that the protocol in [12] was vulnerable to **DoS** attack. In 2017, Zhang et al. [4] proposed a privacy protection dynamic authentication based on three-factor for TMIS. Later, Chaudhry et al. [14] proposed a lightweight authentication based on three-factor for TMIS. However, all schemes above are suitable for single-medical server in same-domain environment.

In 2015, Amin et al. [15] first proposed a novel AKA protocol for accessing remote multi-medical server in TMIS, which was claimed to resist many kinds of attacks. However, Amin et al.'s scheme [15] was vulnerable to internal attack, replay attack and the man-in-middle attack [16]. In 2017, Liu et al.'s [17] pointed out that the protocol in [16] still suffered from internal attack, impersonation attack and stolen smart card attack. Although, these protocols are suitable for multi-medical servers, they are still only for same-domain environment.

In previous years, researchers have presented several cross-domain authenticated key agreement schemes. In 2010, Sun et al. [18] proposed a scheme between PKI and IBC, but their scheme was vulnerable to insider attacks. Later, Huang et al. [19] proposed another scheme based on heterogeneous systems. However, their scheme could not deliver messages from PKI to IBC. In order to compensate for these loopholes, Li et al. [23] proposed a truly scheme between PKI and IBC, where the messages can be transmitted not only from PKI to IBC but also from IBC to PKI. Thereafter, several cross-domain protocols and models are proposed in [2, 5, 23–27]. However, they are managed by one trusted authority(TA) as shown in Fig. 1a. The TA needs to participate in each registration and authentication processes and is possible for the system bottleneck.

Moreover, in the PKI system, the certificate authority (CA) is responsible to distribute, storage, verify and revoke the certificate, which brings a high management cost. In IBC system, each user has an identity and the secret keys of all users are generated by a key generation center (KGC). The identity based cryptosystem will be broken easily if the storage server of KGC is hacked since all the users secret keys are escrowed to KGC. The certificateless cryptography authentication system does not require the certificate system and solves the key escrow problem since the KGC only knows part of the secret key of user. It admirably avoids the disadvantages of PKI and IBC. Some signcryption schemes from IBC to certificateless public key infrastructure (CLPKI) was proposed in [2, 5, 9, 20–23].

However, there is still no certificateless heterogeneous cross-domain authenticated key agreement protocols between PKI-domain and IBC-domain applied to TMIS. It becomes a big obstacle for the patients from PKI-domain and IBC-domain to connect with each other for some help. Although, Yuan et al. [6] proposed a heterogeneous cross-domain authenticated key agreement protocol, as shown in Fig. 1b, it needs heavy calculations because of the public encryption/signature algorithms or other time-consuming computation (such as bilinear pairing). What's more, trusted authorities need to take apart in registration phase and AKA phase. As a consequence, it will easily become the bottleneck of the system. Therefore, it is not suitable for the energy-limited mobile devices. Consider a huge number of mobile terminal patients have limit computation and energy (battery-powered), they frequently login through a remote terminal according to their needs. The low energy remote AKA protocols are urgently required. Therefore, it is unsurprising that constructing the efficient and energy-saving AKA protocols keep pace with the development of the mobile Internet. In this paper, a novel heterogeneous cross-domain authenticated key agreement protocol with symptoms-matching in TMIS is proposed.

Organization

The rest of paper is organized as follows. Some mathematical preliminaries about ECDHP is introduced in "Mathematical preliminaries". "Adversarial model" briefly reviews adversarial model and the CDAKA protocol is presented in "Network frame of CDAKA protocol". Detailed security analysis and proof are given in "Security analysis and proof of CDAKA protocol". The comparisons of the performance and security features between CDAKA protocol with other related schemes are discussed in "Performance evaluation". "Conclusion and ongoing work" concludes this paper.

Mathematical preliminaries

Let P be a large prime number. An elliptic curve $E(F_P)$ over the finite field F_P is defined by the equation: $y^2 = x^3 + \alpha \cdot x + \beta$ mod P, where $\alpha, \beta \in F_P$ and $\triangle = 4\alpha^3 + 27\beta^2 \neq 0$ mod P. All points on $E(F_P)$ are form an additive group G_1 [5, 23].

Elliptic Curve Computable Diffie-Hellman problem (**ECDHP**):

Choose G_1 as an additive cyclic group generated by P, whose order is a prime q. Given $(P, aP, bP) \in G_1$ for any unknown $a, b \in Z_q^*$, , the goal of the ECDHP is to compute abP. Define the advantage of any probabilistic polynomialłtime algorithm $\mathscr{A}$ against ECDHP in G_1. For every probabilistic $\mathscr{A}$, the advantage is negligible, which will be used in the security analysis of our proposed CDAKA protocol.

Adversarial model

There are two types adversaries who have different abilities considered in certificateless cryptography: Type-I $\mathscr{A}_I$ and Type-II $\mathscr{A}_{II}$ [2, 21, 28, 29].

Type-I $\mathscr{A}_I$:$\mathscr{A}_I$ dose not have access to the master-key. However, $\mathscr{A}_I$ may request public keys, replace public keys with values of its choice, extract partial private and private keys and make decryption queries, all for identities of its choice. Some natural restriction on $\mathscr{A}_I$ are as follows:

- $\mathscr{A}_I$ cannot extract the private key for challenge identity at any point.
- $\mathscr{A}_I$ cannot request the private key for any identity if the corresponding public key has already been replaced.

- $\mathscr{A}_I$ cannot both replace the public key for the challenge identity before the challenge phase and extract the partial private key for challenge identity in some phase.
- In Phase 2, $\mathscr{A}_I$ cannot make a decryption query on the challenge ciphertext for the combination of challenge identity and public key that was used to encrypt plaintext.

Type-II $\mathscr{A}_{II}$: The master-key is possessed by $\mathscr{A}_{II}$. But he has no ability to replace the public key of any user. Adversary $\mathscr{A}_{II}$ can compute partial private keys for itself, given master-key. It can also request public keys, make private key extraction queries and decryption queries, both for identities of its choice. The restrictions on this type of adversary are:

- $\mathscr{A}_{II}$ cannot replace public keys at any point.
- $\mathscr{A}_{II}$ cannot extract the private key for challenge identity at any point.
- In Phase 2, $\mathscr{A}_{II}$ cannot make a decryption query on the challenge ciphertext for the combination of challenge identity and public key that was used to encrypt plaintext.

Network frame of CDAKA protocol

The CDAKA protocol is composed of **Registration phase, Login phase** and **Authentication and Key agreement phase**. To simplify the subsequent description, some symbol notations are given in Table 1. Figure 1c simply depicts the heterogeneous cross-domain authentication model. At the beginning, each domain sets up their systems:

In the PKI-domain, PKI randomly selects its private key ω, where $\omega \in Z_q^*$ and computes the corresponding public key $Pub_i = \omega P$. PKI chooses three cryptographically secure one-way hash functions $H_i(\cdot)$: $\{0,1\}^* \rightarrow Z_q^*$ and $i = \{1,2\}$, $H_3(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^l$. PKI chooses a cryptographic symmetric encryption/decryption pair $E(\cdot)/D(\cdot)$ with symmetric key. Then, PKI publishes $\{q, P, E(\cdot)/D(\cdot), Pub_i, H_1, H_2, H_3\}$ and saves ω secretly.

In the IBC-domain, IBC randomly selects its private key s, where $s \in Z_q^*$ and computes the corresponding public key $Pub_j = sP$. IBC chooses three cryptographically secure one-way hash functions $H_i(\cdot)$: $\{0,1\}^* \rightarrow Z_q^*$ and $i = \{1,2\}$, $H_3(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^l$. IBC chooses a cryptographic symmetric encryption/decryption pair $E(\cdot)/D(\cdot)$ with symmetric key. Then, IBC publishes $\{q, P, E(\cdot)/D(\cdot), Pub_j, H_1, H_2, H_3\}$ and saves s secretly.

Registration phase

Patient PA_i in PKI-domain registration phase

When a patient PA_i in PKI-domain wants to access medical services in the system, he/she should register in PKI firstly. The following steps run between PA_i and PKI as shown in Fig. 2.

Ri1 PA_i chooses his/her ID_i and a random number $x_i \in Z_q^*$ and computes $P_i = x_i P$. Then, $PA_i \Rightarrow PKI$: (ID_i, P_i, S_i);

Ri2 Upon receiving the registration message from PA_i, PKI chooses random value $\xi_i, r_i \in Z_q^*$, computes $pid_i = E_\omega(ID_i, \xi_i, S_i)$, $R_i = r_i P$, $\alpha_i = H_1(pid_i||P_i||R_i||S_i)$, $y_i = \alpha_i\omega + r_i$ and stores pid_i in its database. Then, $PKI \Rightarrow PA_i$: (pid_i, R_i, y_i);

Table 1 Symbol notations

Symbol	Description
$\mathscr{S} = \{S_1, S_2, S_3...\}$	A set of disease symptom
PKI	The registration center of PKI-domain
IBC	The registration center of IBC-domain
PA_i	ith patient(user) who can access medical services in PKI-domain
PA_j	jth patient(user) who can access medical services in IBC-domain
PID_i/PID_j	randomized-dual pseudonym of PA_i/PA_j
pid_i/pid_j	Static anonymous identity of PA_i/PA_j
$(\omega, Pub_i = \omega P)$	The pair of master secret key and public key hold by PKI
$(s, Pub_j = sP)$	The pair of master secret key and public key hold by IBC
$E(\cdot)/D(\cdot)$	Secure symmetric encryption/decryption pair
(sk_i, Pub_i)	The pair of master secret key and public key hold by PK_i
(sk_j, Pub_j)	The pair of master secret key and public key hold by PK_j
$H(\cdot)$	A cryptographically secure one way hash function
$\oplus, \|\|$	Bitwise XOR operation and concatenation operation
$\rightarrow$	A public communication channel
$\Rightarrow$	A secure communication channel

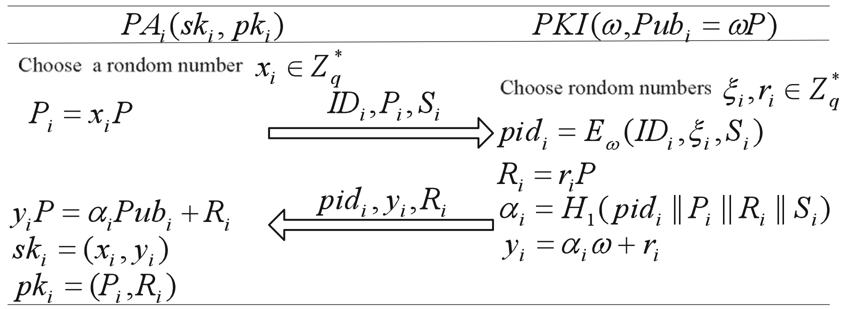


Fig. 2 Patient PA_i in PKI-Domain registration phase

Ri3 After receiving the message (pid_i, R_i, y_i) from PKI, PA_i checks $y_i P ?= \alpha_i Pub_i + R_i$. If the verification fails, the request is rejected. Otherwise, PA_i stores secret key $sk_i = (x_i, y_i)$ securely and airs public key $pk_i = (P_i, R_i)$.

Patient PA_j in IBC-domain registration phase

When a patient PA_j in IBC-domain wants to access medical services in the system, he/she should register in IBC firstly. The following steps run between PA_j and IBC as shown in Fig. 3.

Rj1 PA_j chooses his/her ID_j. Then, $PA_j \Rightarrow IBC$: (ID_j, S_j);

Rj2 Upon receiving the registration message from PA_j, IBC selects random value $\xi_j \in Z_q^*$, computes $pid_j = E_s(ID_j, \xi_j, S_j)$, $r_j = H_2(ID_j)$, $R_j = r_j P$, $\alpha_j = H_1(pid_j||R_j||S_j)$, $y_j = \alpha_j s + r_j$ and stores pid_j in its database. Then, $IBC \Rightarrow PA_j$: (pid_j, R_j, y_j);

Rj3 After receiving the message (pid_j, R_j, y_j) from IBC, PA_j checks $y_j P ?= \alpha_j Pub_j + R_j$. If the verification fails, the request is rejected. Otherwise, PA_j selects a random number $x_j \in Z_q^*$, computes $P_j = x_j P$, stores secret key $sk_j = (x_j, y_j)$ and airs public key $pk_j = (P_j, R_j)$.

Login phase

When PA_i and PA_j want to establish a session key to exchange status about their illness and share their positive experience of treatment, they will compute $\alpha_i = H_1(pid_i||P_i||R_i||S_i)$ and $\alpha_j = H_1(pid_j||R_j||S_j)$ respectively, exchange $\{\alpha_i, Pub_i, P_i, R_i, S_i\}$ and $\{\alpha_j, Pub_j, P_j, R_j, S_j\}$ preferentially to achieve mutual authentication. After receiving the interactive messages, they first check $S_i ?= S_j$. If it does not match, terminate the session. Otherwise, do the following steps as shown in Fig. 4:

L1 PA_i selects random value $a_i \in Z_q^*$, reads the current time T_i^1 and computes $M_{i1} = (\alpha_j Pub_j + P_j + R_j)(x_i + y_i)$, $M_{i2} = M_{i1} \oplus a_i P$, $PID_i = H_3(M_{i1}||M_{i2}) \oplus pid_i$, $M_{i3} = H_3(pid_i||Pub_i||P_i||R_i||Pub_j||P_j||R_j||T_i^1)$, $M_{i4} = H_3(M_{i1}||M_{i2}||M_{i3}||a_i P||pid_i)$. Then, $PA_i \rightarrow PA_j$: $\textbf{msg1} = \{PID_i, M_{i2}, M_{i4}, T_i^1\}$.

Authentication and key agreement phase

V1 Upon receiving **msg1**, PA_j reads the current time T_j^1, checks $|T_j^1 - T_i^1| ?\le \Delta T$ and the pair (PID_i, M_{i2}) according to PID_i. If that above verifications do not hold, the login request is rejected. Otherwise; PA_j

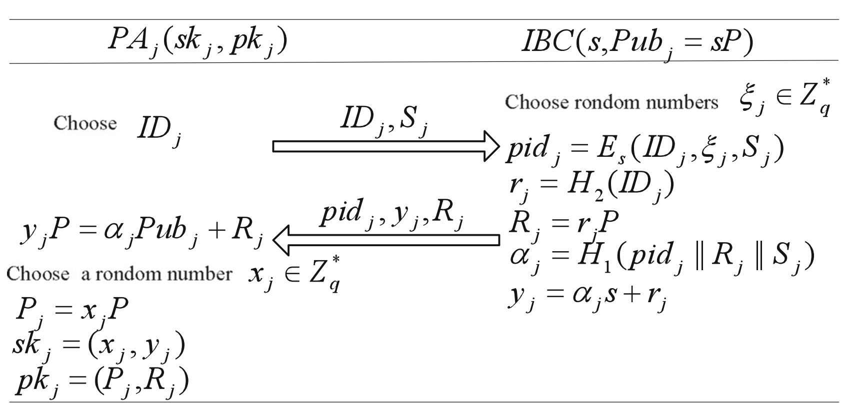
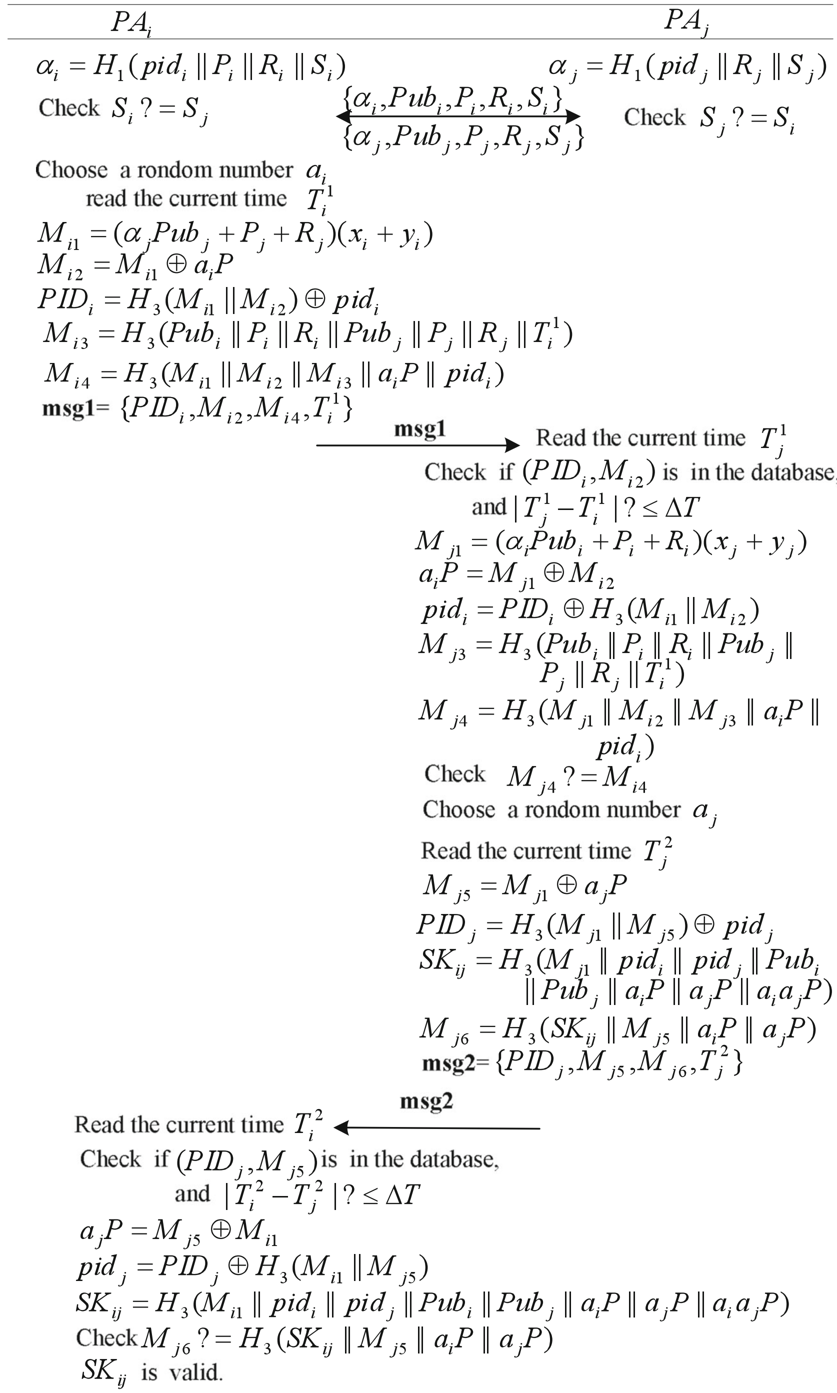


Fig. 3 Patient PA_j in IBC-Domain registration phase

Fig. 4 Authentication and key agreement phase



computes, $M_{j1} = (\alpha_i Pub_i + P_i + R_i)(x_j + y_j)$, $a_i P = M_{j1} \oplus M_{i2}$, $pid_i = PID_i \oplus H_3(M_{j1}||M_{i2})$, $M_{j3} = H_3(pid_i||Pub_i||P_i||R_i||Pub_j||P_j||R_j||T_i^1)$, and checks $M_{j4} ? = H_3(M_{j1}||M_{i2}||M_{j3}||a_i P||pid_i)$. If the equality is not established, the login request is rejected. Otherwise;

V2 PA_j replaces the M_{i2}^{old} with M_{i2}, stores the pair (PID_i, M_{i2}) in database, generates a random value a_j, reads the current time T_j^2, and computes $M_{j5} = M_{j1} \oplus a_j P$, $PID_j = H_3(M_{j1}||M_{j5}) \oplus pid_j$, $SK_{ij} = H_3(M_{j1}||pid_i||pid_j||Pub_i||Pub_j||a_i P||a_j P||a_i a_j P)$, $M_{j6} = H_3(SK_{ij}||M_{j5}||a_i P||a_j P||T_j^2)$. Then, $PA_j \rightarrow PA_i$: **msg2** $= \{PID_j, M_{j5}, M_{j6}, T_j^2\}$;

V3 Upon receiving **msg2**, PA_i reads the current time T_i^2, checks $|T_i^2 - T_j^2| ? \le \Delta T$ and checks the pair (PID_j, M_{j5}) according to PID_j. If that above verifications do not hold, the authentication request is rejected. Otherwise; PA_i replaces the M_{j5}^{old} with M_{j5}, stores the pair (PID_j, M_{j5}) in database, computes $a_jP = M_{j5} \oplus M_{i1}$, $pid_j = PID_j \oplus H_3(M_{j1}||M_{j5})$, $SK_{ij} = H_3(M_{j1}||pid_i||pid_j||Pub_i||Pub_j||a_iP||a_jP||a_ia_jP)$, and checks $M_{j6}? = H_3(SK_{ij}||M_{j5}||a_iP||a_jP||T_j^2)$. If it is not equal, the session is terminated. Otherwise, PA_j is authenticated by PA_i. At last, PA_i and PA_j share the session key SK_{ij}.

Security analysis and proof of CDAKA protocol

In this section, we will analyze the security of the CDAKA protocol under the same adversary model mentioned in "Adversarial model".

Security analysis

Completeness and mutual authentication and key agreement

In the CDAKA protocol, all the authentication information $(M_{i2}, M_{i4}, M_{j5}, M_{j6})$ are based on secret value $M_{i1}(M_{j1})$,

$$\begin{aligned} M_{i1} &= (\alpha_j Pub_j + P_j + R_j)(x_i + y_i) = (y_jP + P_j)(x_i + y_i) \\ &= (y_jP + x_jP)(x_i + y_i) \\ M_{j1} &= (\alpha_i Pub_i + P_i + R_i)(x_j + y_j) = (y_iP + P_i)(x_j + y_j) \\ &= (y_iP + x_iP)(x_j + y_j) \end{aligned}$$

(Here, $y_i = \alpha_i\omega + r_i$, $y_iP = \alpha_i Pub_i + R_i$ and $y_j = \alpha_j s + r_j$, $y_jP = \alpha_j Pub_j + R_j$), which is only shared between PA_i and PA_j, which anyone cannot obtain it except PA_i and PA_j. In the whole protocol as shown in Fig. 4, PA_i authenticates PA_j, and PA_j authenticates PA_i. In the end, they share a session key SK_{ij}. Hence, the CDAKA achieves mutual authentication and key agreement.

Patient anonymity

The CDAKA protocol adopts the anonymous blind identities $PID_i = H_3(M_{i1}||M_{i2}) \oplus pid_i$ and $PID_j = H_3(M_{j1}||M_{j5}) \oplus pid_j$ instead of the static identity ID_i and ID_j in the public communication channel. Meanwhile, they are differen in each run. Here, $pid_i = E_\omega(ID_i, \xi_i, S_i)$ and $pid_j = E_s(ID_j, \xi_j, S_j)$. By using a secure cryptographic symmetric encryption, the malicious adversary $\mathscr{A}$ cannot extract the ID_i and ID_j without knowing ω or s required to successfully decrypt the ciphertext. Further, in the CDAKA protocol, the patients PA_i and PA_j cannot know the others' real identity either. In this way, the CDAKA protocol provides patient anonymity, which can prevent the privacy leakage of patient identity.

Patient traceability

If a patient PA_i sends same false messages to deceive others, PKI or IBC can extract real identity of PA_i or PA_j by decrypting pid_i or pid_j using their private key ω or s. Hence, the CDAKA achieves patient traceability to prevent malicious users from doing something to harm systems.

Cross-domain communication

According to the specification of CDAKA protocol, two patients PA_i in PKI-domain and PA_j in IBC-domain separately registered with PKI and IBC can authenticate each other and generate a session key for secure communication. Hence, the CDAKA protocol can provide heterogeneous cross-domain communication.

Perfect forward secrecy

In the CDAKA protocol, suppose $\mathscr{A}$ steals both private keys of two patients PA_i and PA_j. We also assume that $\mathscr{A}$ intercepts messages **msg1** $= \{PID_i, M_{i2}, M_{i4}, T_i^1\}$, **msg2** $= \{PID_j, M_{j5}, M_{j6}, T_j^2\}$ transmitted between PA_i and PA_j. Using their private keys, the adversary is able to compute $M_{i1} = (\alpha_j Pub_j + P_j + R_j)(x_i + y_i)$ and $M_{j1} = (\alpha_i Pub_i + P_i + R_i)(x_j + y_j)$ to obtain $a_iP = M_{i1} \oplus M_{i2}$ and $a_jP = M_{j1} \oplus M_{j5}$ further. To obtain the session key $SK_{ij} = H_3(M_{j1}||pid_i||pid_j||Pub_i||Pub_j||a_iP||a_jP||a_ia_jP)$, $\mathscr{A}$ has to compute a_ia_jP from a_iP and a_jP. In other words, $\mathscr{A}$ has to solve the ECDHP. Due to the hardness of the ECDHP, the CDAKA protocol provides perfect forward secrecy.

Impersonation attack

If $\mathscr{A}$ can obtain the information **msg1** $= \{PID_i, M_{i2}, M_{i4}, T_i^1\}$, **msg2** $= \{PID_j, M_{j5}, M_{j6}, T_j^2\}$ in public channel. $\mathscr{A}$ (other domain servers and malicious-legitimate patients) cannot get the secret information M_{i1} and M_{j1} only shared between PA_i and PA_j. So $\mathscr{A}$ can not figure out the valid authentication message $M_{j4} = H_3(M_{j1}||M_{i2}||M_{j3}||a_iP||pid_i)$ and $M_{j6} = H_3(SK_{ij}||M_{j5}||a_iP||a_jP)$ to pass the authentication. Hence, the CDAKA protocol can resist the impersonation attack.

Internal attacks

Assume that $\mathscr{A}$ is a malicious-legitimate patient, $\mathscr{A}$ uses his own information in public channel. He obtains nothing about other patients' secret information M_{i1} and M_{j1}. And he also cannot get the random values a_iP

or $a_j P$. So he cannot succeed in forging authentication information $M_{j4} = H_3(M_{j1}||M_{i2}||M_{j3}||a_i P||pid_i)$ and $M_{j6} = H_3(SK_{ij}||M_{j5}||a_i P||a_j P)$ to pass the authentication. Hence, the CDAKA protocol can resist the internal attacks.

Replay attack

Suppose $\mathscr{A}$ intercepts the massage **msg1**, where $M_{i2} = M_{i1} \oplus a_i P$, $M_{i4} = H_3(M_{i1}||M_{i2}||M_{i3}||a_i P||pid_i)$, and replies this message to PA_j. However, PA_j stores the pair (PID_i, M_{i2}) in its database. Later, when PA_j receives the next login request message **msg1**, PA_j compares M_{i2} corresponding to PID_i. If it matches, PA_j ensures that this request message is a replay message and rejects this request. Or else, PA_j replaces M_{i2} with M_{i2}^{new}. So does the PA_i. Hence, the CDAKA protocol can resist the replay attack.

Man-in-the-middle attack

In this attack, $\mathscr{A}$ may try to impersonate a valid patient PA_i, or his partner PA_j by intercepting the message. However, in the CDAKA protocol the secret values M_{i1} and M_{j1} are only shared between PA_i and PA_j, they will never be discovered by anybody else except PA_i and PA_j. Hence, the CDAKA protocol is secure against man-in-the-middle attack.

Security proof

Assuming that the ECDHP is hard, the security of the CDAKA protocol is demonstrated blow.

Theorem 1 *In the random oracle, if there exists a type-I adversary $\mathscr{A}_I$, who is able to forge a legitimate login message or its partner's respond message with a non-negligible probability ε in time t. We show that there is a challenger $\mathscr{C}$ who can solve the ECDHP with a non-negligible probability ε', where*

$$\varepsilon' \geq \left(1 - \frac{2}{q_{ep}+1}\right)^{q_{ep}} \left(1 - \frac{2}{q_{sq}+1}\right)^{q_{sq}} \frac{1}{nm} \frac{2}{q_{H_3}} \varepsilon,$$

in time

$$t' \leq t + 2q_{se}t_{se} + 2(q_{es} + q_{ep} + 2q_{sq})t_{sm}.$$

Here, $q_{se}, q_{H_i}, q_{es}, q_{ep}, q_{sq}$ denote the times of symmetric-encryption queries, hash-query,extract-secret-value queries, extract-partial-secret-value queries and send queries. n and m denote the number of patients in PKI-domain and IBC-domain separately. t_{se} and t_{sm} denote the time of symmetric-encryption and scalar multiplications separately.

Proof Let $\mathscr{C}$ be a ECDHP challenger who receives a random instance $(P, Q_1 = aP, Q_2 = bP)$ of ECDHP in G_1. A type-I adversary $\mathscr{A}_I$ interacts with $\mathscr{C}$ as follows. We show how $\mathscr{C}$ may use $\mathscr{A}_I$ to solve the ECDHP, that is to compute abP. □

Setup: $\mathscr{C}$ randomly selects the initiator patient PA_I in PKI-domain and the responder patient PA_J in IBC-domain as the challenge patients. Then, $\mathscr{C}$ generates six numbers $\alpha_I, r_I, x_I\alpha_J, r_J, x_J \in Z_q^*$ randomly, computes $Pub_i = \alpha_I^{-1}(Q_1 - r_I P - x_I P)$, $Pub_j = \alpha_J^{-1}(Q_2 - r_J P - x_J P)$ and gives $\{q, P, G_1, Pub_i, Pub_j, H_1, H_2, H_3\}$ to $\mathscr{A}_I$ as public parameters. $\mathscr{C}$ maintains the following lists to avoid inconsistency and for quick response to the adversary $\mathscr{A}_I$:

Symmetric encryption query: A list L_{se} is utilized to store the query result. Obtaining a symmetric encryption query on m_k and key k_k. $\mathscr{C}$ checks whether a tuple (m_k, k_k, c_k) exists in L_{se}. If it exists, c_k is returned. Otherwise, $\mathscr{C}$ selects a randomized string $c_k \in \{0, 1\}^*$, stores in L_{se} and sends c_k to $\mathscr{A}_I$.

Hash query: $\mathscr{C}$ maintains several initialized-empty lists L_{H_k}. Upon receiving the Hash query with m_k. $\mathscr{C}$ checks whether a tuple (m_k, n_k) exists in L_{H_k}. If it exists, n_k is returned. Otherwise, $\mathscr{C}$ selects a randomized value n_k, stores in L_{H_k} and sends n_k to $\mathscr{A}_I$, where $k = 1, 2, 3$.

Extract secret value of (PA_k): A initialized-empty list L_{PA}^1 is utilized to store the query result. Obtaining a secret value extraction on patient PA_k with identity ID_k. $\mathscr{C}$ checks whether a tuple (PA_k, ID_k, x_k, P_k) exists in L_{PA}^1. If it exists, x_k is returned. Otherwise, $\mathscr{C}$ selects a random number $x_k \in Z_q^*$, computes $P_k = x_k P$, stores the new tuple in L_{PA}^1 and sends x_k to $\mathscr{A}_I$.

Extract partial secret value query(PA_k): $\mathscr{C}$ maintains several initialized-empty lists L_{PA}^2. Upon receiving the partial secret value query on the patient PA_k, $\mathscr{C}$ checks whether a tuple (PA_k, pid_k, R_k, y_k) exists in L_{PA}^2. If it exists, y_k is returned. Otherwise, $\mathscr{C}$ calculates as following:

- If $PA_k = PA_I$, $\mathscr{C}$ selects random number $\xi_I \in Z_q^*$, random string $pid_I \in \{0, 1\}^*$ and inserts the tuple $((ID_I \oplus \xi_I), \perp, pid_I)$ into list L_{se}. $\mathscr{C}$ computes $R_I = r_I P$, sets $y_I = \perp$ and reads P_I from the list L_{PA}^1 according to PA_I. At last, $\mathscr{C}$ stores $(pid_I, P_I, R_I, S_I, \alpha_I)$ and $(ID_I, pid_I, r_I, R_I, \perp)$ into L_{H_1} and L_{PA}^2 separately.
- If $PA_k = PA_J$, $\mathscr{C}$ selects random number $\xi_J \in Z_q^*$, random string $pid_J \in \{0, 1\}^*$ and inserts the tuple $((ID_J \oplus \xi_J), \perp, pid_J)$ into list L_{se}. $\mathscr{C}$ reads $H_2(ID_J)$ from the list L_{PA}^1 according to ID_J, computes $R_J = r_J H_2(ID_J)$, sets $y_J = \perp$. At last, $\mathscr{C}$ stores $(pid_J, R_J, T_J, \alpha_J)$

and $(ID_J, pid_J, r_J, R_J, \perp)$ into L_{H_1} and L^2_{PA} separately.

– Otherwise, $\mathscr{C}$ selects random value $\xi_k, \alpha_k \in Z_q^*$, $pid_k \in \{0,1\}^*$, inserts $((ID_k \oplus \xi_k), \perp, pid_k)$ into list L_{se}, computes $R_k = \alpha_k^{-1} r_k P + Pub_k$, and sets $y_k = \alpha_k r_k$ (Here, if PA_k is in PKI-domain, $Pud_k = Pub_i$ and r_k is random number chosen by $\mathscr{C}$. Otherwise, $Pud_k = Pub_j$ and $r_k = H_2(ID_k)$). At last, $\mathscr{C}$ stores $(pid_k, P_k, R_k, S_k, \alpha_k)$ and (ID_k, pid_k, R_k, y_k) into L_{H_1} and L^2_{PA} separately.

Request public key of (PA_k): A initialized-empty list L^3_{PA} is utilized to store the query result. Obtaining a request public key on patient PA_k. $\mathscr{C}$ checks whether a tuple $(PA_k, x_k, P_k, r_k, R_k)$ exists in L^3_{PA}. If it exists, (P_k, R_k) is returned. Otherwise, $\mathscr{C}$ responds (P_k, R_k) by accessing to list L^1_{PA} and list L^2_{PA} and set d_k:=0 (d_k denotes the time of public key replacement). At last, the tuple $(PA_k, x_k, P_k, r_k, R_k, d_k)$ is inserted to L^3_{PA}.

Replace public key of (PA_k): Upon receiving the replace public key query on the patient PA_k, $\mathscr{C}$ first makes a request public key on (PA_k) and finds the tuple $(PA_k, x_k, P_k, r_k, R_k, d_k)$ on L^3_{PA}. Then, $\mathscr{C}$ replaces $pk_k = (P_k, R_k)$ with $pk'_k = (P'_k, R'_k)$ which is chosen by $\mathscr{A}_I$ and puts $d_k := d_k + 1$. At last, the tuple $(PA_k, x'_k, P'_k, r'_k, R'_k, d_k)$ is inserted to L^4_{PA}.

Send query of (PK_k, M): Obtaining the send query with mesage M, $\mathscr{C}$ responds the query as follows:

– $M = (M_{i2}, M_{i4})$: The query is message M from PA_i to PA_j.
  - If $PA_i = PA_I$, $\mathscr{C}$ aborts the session.
  - If $PA_i \neq PA_I$, $PA_j = PA_J$, $\mathscr{C}$ aborts the session.
  - If $PA_i \neq PA_I$, $PA_j \neq PA_j$, $\mathscr{C}$ runs according to the specification of the protocol, where $\mathscr{C}$ knows the private key of PK_i.

– $M = (M_{j5}, M_{j6})$: The query is message M from PA_j to PA_i.
  - If $PA_j = PA_J$, $\mathscr{C}$ aborts the session.
  - If $PA_j \neq PA_J$, $PA_i = PA_I$, $\mathscr{C}$ aborts the session.
  - If $PA_j \neq PA_j$, $PA_i \neq PA_I$, $\mathscr{C}$ runs according to the specification of the protocol, where $\mathscr{C}$ knows the private key of PK_j.

Reveal query of (PK_k): Upon receiving the query, $\mathscr{C}$ checks if $PA_k = PA_I$ or $PA_k = PA_J$. If yes, $\mathscr{C}$ aborts the session. Otherwise, $\mathscr{C}$ returns the session key between PA_k and its partner to $\mathscr{A}_I$.

Corrupt query of (PK_k): Obtaining the corrupt query, $\mathscr{C}$ looks up the list L^1_{PA} and the list L^2_{PA} for the tuples (PA_k, ID_k, x_k, P_k) and (PA_k, pid_k, R_k, y_k). Then, $\mathscr{C}$ returns (x_k, P_k, R_k, y_k) to $\mathscr{A}_I$.

Finally, $\mathscr{A}_I$ outputs a legitimate login message (M_{i2}, M_{i4}) or its partner's respond message (M_{j5}, M_{j6}). If $(PK_i, PK_j) \neq (PK_I, PK_J)$, $\mathscr{C}$ aborts the game. Otherwise, $\mathscr{C}$ randomly chooses a tuple $(*, M_{i1}, *)$ or $(*, M_{j1}, *)$ from the list L_{H_3} and outputs M_{i1} or M_{j1} as the solution of ECDHP.

To complete the the proof, we shall show that $\mathscr{C}$ solves the given instances of ECDHP with probability ε'. First, we analyze several events for $\mathscr{C}$ to succeed:

– $E1$: $\mathscr{C}$ does not abort any $\mathscr{A}_I$'s "Extract partial secret value queries".
– $E2$: $\mathscr{C}$ does not abort any $\mathscr{A}_I$'s "Send queries".
– $E3$: $\mathscr{C}$ obtains a legitimate login message or its partner's respond message.
– $E4$: $(PK_i, PK_i) = (PK_I, PK_J)$.
– $E5$: $\mathscr{C}$ chooses a correct tuple from the list L_{H_3}.

Then, we have:

$$\Pr[E1] \geq \left(1 - \frac{2}{q_{ep}+1}\right)^{q_{ep}}$$
$$\Pr[E2|E1] \geq \left(1 - \frac{2}{q_{sq}+1}\right)^{q_{sq}}$$
$$\Pr[E3|E1 \wedge E2] \geq \varepsilon$$
$$\Pr[E4|E1 \wedge E2 \wedge E3] \geq \frac{1}{nm}$$
$$\Pr[E5|E1 \wedge E2 \wedge E3 \wedge E3] \geq \frac{2}{q_{H_3}}$$

Hence, we have:

$$\begin{aligned}\varepsilon' &= \Pr[E1 \wedge E2 \wedge E3 \wedge E4 \wedge E5] = \Pr[E1]\Pr[E2|E1]\Pr[E3|E1 \\ &\quad \wedge E2]\Pr[E4|E1 \wedge E2 \wedge E3]\Pr[E5|E1 \wedge E2 \wedge E3 \wedge E3] \\ &\geq \left(1 - \frac{2}{q_{ep}+1}\right)^{q_{ep}} \left(1 - \frac{2}{q_{sq}+1}\right)^{q_{sq}} \frac{1}{nm}\frac{2}{q_{H_3}}\varepsilon.\end{aligned}$$

The running time t for $\mathscr{C}$ is the sum of $\mathscr{A}_I$'s running time, the time that $\mathscr{C}$ responds queries and the time that $\mathscr{C}$ computes the ECDHP. Hence,

$$t' \leq t + 2q_{se}t_{se} + 2(q_{es} + q_{ep} + 2q_{sq})t_{sm}.$$

Theorem 2 *In the random oracle, if there exists a type-II adversary $\mathscr{A}_{II}$, who is able to forge a legitimate login message or its partner's respond message with a non-negligible probability ε in time t. We show that there is* *a challenger $\mathscr{C}$ who can solve the ECDHP with a non-negligible probability*

$$\varepsilon' \geq \left(1 - \frac{2}{q_{es}+1}\right)^{q_{es}} \left(1 - \frac{2}{q_{sq}+1}\right)^{q_{sq}} \frac{1}{nm}\frac{2}{q_{H_3}}\varepsilon.$$

in time

$$t' \leq t + 2q_{se}t_{se} + 2(q_{es} + 2q_{sq})t_{sm}.$$

Proof Let $\mathscr{C}$ be a ECDHP challenger who receives a random instance $(P, Q_1 = aP, Q_2 = bP)$ of ECDHP in G_1. A type-II adversary $\mathscr{A}_{II}$ interacts with $\mathscr{C}$ as follows. We show how $\mathscr{C}$ may use $\mathscr{A}_{II}$ to solve the ECDHP, that is to compute abP. □

Setup: $\mathscr{C}$ randomly selects the initiator patient PA_I in PKI-domain and the responder patient PA_J in IBC-domain as the challenge patients. Then, $\mathscr{C}$ generates two numbers $\omega, s \in Z_q^*$ randomly, computes $Pub_i = \omega P$, $Pub_j = sP$ and gives $\{q, P, G_1, Pub_i, Pub_j, H_1, H_2, H_3\}$ to $\mathscr{A}_{II}$ as public parameters. $\mathscr{C}$ maintains the following lists to avoid inconsistency and for quick response to the adversary $\mathscr{A}_{II}$:

Due to the initiate-respond process of "**Symmetric encryption query**", "**Hash query**" and "**Extract secret value query**" are same as **Theorem 1.**. We will not repeat them here. For more details, please refer to Theorem 1..

Request public key of (PA_k): A initialized-empty list L^3_{PA} is utilized to store the query result. Obtaining a request public key on patient PA_k. $\mathscr{C}$ checks whether a tuple $(PA_k, x_k, P_k, r_k, R_k)$ exists in L^3_{PA}. If it exists, (P_k, R_k) is returned. Otherwise, $\mathscr{C}$ calculates as following:

- If $PA_k = PA_I$, $\mathscr{C}$ obtains α_I, P_I by accessing to L_{H_1} and L^1_{PA} and computes $R_I = Q_1 - \alpha_I Pub_i - P_I$. At last, the tuple (PA_I, P_I, R_I) is inserted to L^3_{PA}.
- If $PA_k = PA_J$, $\mathscr{C}$ obtains α_J, P_J by accessing to L_{H_1} and L^1_{PA} an computes $R_J = Q_2 - \alpha_J Pub_j - P_J$. At last, the tuple (PA_J, P_J, R_J) is inserted to L^3_{PA}.
- If $PA_k \neq PA_I$, $PA_k \in PKI-domain$, $\mathscr{C}$ selects random number $r_k \in Z_q^*$, and computes $R_k = r_k P$. At last, the tuple (PA_k, P_k, R_k) is inserted to L^3_{PA}.
- If $PA_k \neq PA_J$, $PA_k \in IBC-domain$, $\mathscr{C}$ obtains $H_2(ID_k)$ by accessing to L_{H_2} and computes $R_k = H_2(ID_k)P$. At last, the tuple (PA_k, P_k, R_k) is inserted to L^3_{PA}.

Send query of (PK_k, M): Obtaining the send query with mesage M, $\mathscr{C}$ responds the query as follows:

- $M = (M_{i2}, M_{i4})$: The query is message M from PA_i to PA_j.
  - If $PA_i = PA_I$, $\mathscr{C}$ aborts the session.
  - If $PA_i \neq PA_I$, $PA_j = PA_J$, $\mathscr{C}$ aborts the session.
  - If $PA_i \neq PA_I$, $PA_j \neq PA_j$, $\mathscr{C}$ runs according to the specification of the protocol, where $\mathscr{C}$ knows the private key of PK_i.
- $M = (M_{j5}, M_{j6})$: The query is message M from PA_j to PA_i.
  - If $PA_j = PA_J$, $\mathscr{C}$ aborts the session.
  - If $PA_j \neq PA_J$, $PA_i = PA_I$, $\mathscr{C}$ aborts the session.
  - If $PA_j \neq PA_j$, $PA_i \neq PA_I$, $\mathscr{C}$ runs according to the specification of the protocol, where $\mathscr{C}$ knows the private key of PK_j.

Reveal query of (PK_k): Upon receiving the query, $\mathscr{C}$ checks if $PA_k = PA_I$ or $PA_k = PA_J$. If yes, $\mathscr{C}$ aborts the session. Otherwise, $\mathscr{C}$ returns the session key between PA_k and its partner to $\mathscr{A}_{II}$.

To complete the the proof, we shall show that $\mathscr{C}$ solves the given instances of ECDHP with probability ε'. First, we analyze several events for $\mathscr{C}$ to succeed:

- $E1$: $\mathscr{C}$ does not abort any $\mathscr{A}_{II}$'s "Extract secret value queries".
- $E2$: $\mathscr{C}$ does not abort any $\mathscr{A}_{II}$'s "Send queries".
- $E3$: $\mathscr{C}$ obtains a legitimate login message or its partner's respond message.
- $E4$: $(PK_i, PK_i) = (PK_I, PK_J)$.
- $E5$: $\mathscr{C}$ chooses a correct tuple from the list L_{H_3}.

Then, we have:

$$\Pr[E1] \geq \left(1 - \frac{2}{q_{es}+1}\right)^{q_{es}}$$

$$\Pr[E2|E1] \geq \left(1 - \frac{2}{q_{sq}+1}\right)^{q_{sq}}$$

$$\Pr[E3|E1 \wedge E2] \geq \varepsilon$$

$$\Pr[E4|E1 \wedge E2 \wedge E3] \geq \frac{1}{nm}$$

$$\Pr[E5|E1 \wedge E2 \wedge E3 \wedge E3] \geq \frac{2}{q_{H_3}}$$

Hence, we have:

$$\varepsilon' = \Pr[E1 \wedge E2 \wedge E3 \wedge E4 \wedge E5] = \Pr[E1]\Pr[E2|E1] \Pr[E3|E1 \wedge E2]\Pr[E4|E1 \wedge E2 \wedge E3]\Pr[E5|E1 \wedge E2 \wedge E3 \wedge E3] \geq \left(1 - \frac{2}{q_{es}+1}\right)^{q_{es}} \left(1 - \frac{2}{q_{sq}+1}\right)^{q_{sq}} \frac{1}{nm}\frac{2}{q_{H_3}}\varepsilon.$$

Table 2 Computational notations

Operation	Times(ms)	Description
t_b	7.3	The time complexity for scarlar bilinear paring operation
t_m	8.5	The time complexity for multiplication operation
t_{sg}	28.1	The time complexity for signature generation operation
t_{ed}	3.85	The time complexity for encryption/decryption operation
t_{hp}	4.406	The time complexity for hash-to-point operation

The running time t for $\mathscr{C}$ is the sum of $\mathscr{A}_{II}$'s running time, the time that $\mathscr{C}$ responds queries and the time that $\mathscr{C}$ computes the ECDHP. Hence,

$$t' \leq t + 2q_{se}t_{se} + 2(q_{es} + 2q_{sq})t_{sm}.$$

Performance evaluation

In this paper, the communication cost is reduced by removing the unnecessary information transmitted, while remaining high security. The computation cost is mainly discussed in the following. We compare CDAKA protocol to the [5] and [6] protocols, both of which provide cross-domain authenticated key agreement. For convenience, we define some notations about the running time and energy cost in Tables 2 and 3 [4, 30–32, 34], respectively. In addition, we also discuss how our protocol is efficient than others from its implementation point of view later in this section as roughly shown in Fig. 5.

NOTE: We mainly focus on the efficiency of login and authentication phases, since these two phases are the main body of an authentication scheme and are executed much more frequently than the other phases.

Computation cost

We analyze and compare the computation cost of CDAKA protocol and related AKA protocols. Let t_h, t_c, t_x, t_b, t_m, t_{sg}, t_{ed} and t_{hp} denote hash function, concatenation operation, XOR operation, the time complexity for scarlar bilinear paring operation, multiplication operation, signature generation operation, encryption/decryption operation and hash-to-point operation. Since the time of hash function, concatenation operation and XOR operation are negligible as compared to the other five operations, we do not take t_h, t_c and t_x into account.

Based on the implementation results in [33], we analyze and compare the computation cost of related AHA protocols, as shown in Fig. 5a. The comparisons among related protocols are listed in Table 4.

In session initiator's side, He et al.'s [5] protocol has to carry out six multiplication operations and two hash-to-point operations. Therefore, the running time of patients is $6t_m + 2t_{hp} \approx 59.812$ ms. In session response's side, it cost six multiplication operations and two hash-to-point operations, too. Hence, the running time is 59.812 ms. In the trusted authenticated (TA) side, the TAs not participate in the these processes. Hence, the running time of them is 0 ms. The total time is $59.812 + 59.812 = 119.624$ ms.

In session initiator's side, Yuan et al.'s [6] protocol has to carry out one scalar bilinear paring operation, four multiplication operations, one signature generation operation, thirteen encryption/decryption operations and one hash-to-point operation. Therefore, the running time is $1t_b + 4t_m + 1t_{sg} + 13t_{ed} + 1t_{hp} \approx 123.856$ ms. In session response's side, it costs thirteen encryption/decryption operations. Hence, the running time is $13t_{ed} \approx 50.05$ ms. In the trusted authenticated (TA) side, it has to carry out two scalar bilinear paring operations, two multiplication operations, one signature generation operation and ten encryption/decryption operations. Hence, the running time is $2t_b + 2t_m + 1t_{sg} + 10t_{ed} \approx 98.2$ ms. In certificate authority (CA) side, it has to carry out three signature generation operations, seven encryption/decryption operations and one hash-to-point operation. Therefore, the running time is $3t_{sg} + 7t_{ed} + 1t_{hp} \approx 115.656$ ms. The total time is $123.856 + 50.05 + 98.2 + 115.656 = 387.762$ ms.

In session initiator's side, the CDAKA protocol has to carry out four multiplication operations and one hash-to-point operation. Therefore, the running time is $4t_m + 1t_{hp} \approx 38.406$ ms. In session response's side, it has to carry out the same operations. Hence, the running time is $4t_m + 1t_{hp} \approx 38.406$ ms. The users can remotely communicate with the other ones by themselves without the help of their

Table 3 Energy notations

Operation	Energy cost
$multiplicationoperation$	55 mJ/160 bits
$Hash - to - pointoperation$	28.5 mJ/160 bits
$signatureoperation$	52 mJ/160 bits
$encryption/decryptionoperation$	38 μJ/128 bits
$hashoperationof SHA - 1$	5.9 μJ/byte
$transmit$	59.2 μJ/byte
$receive$	26.9 μJ/byte

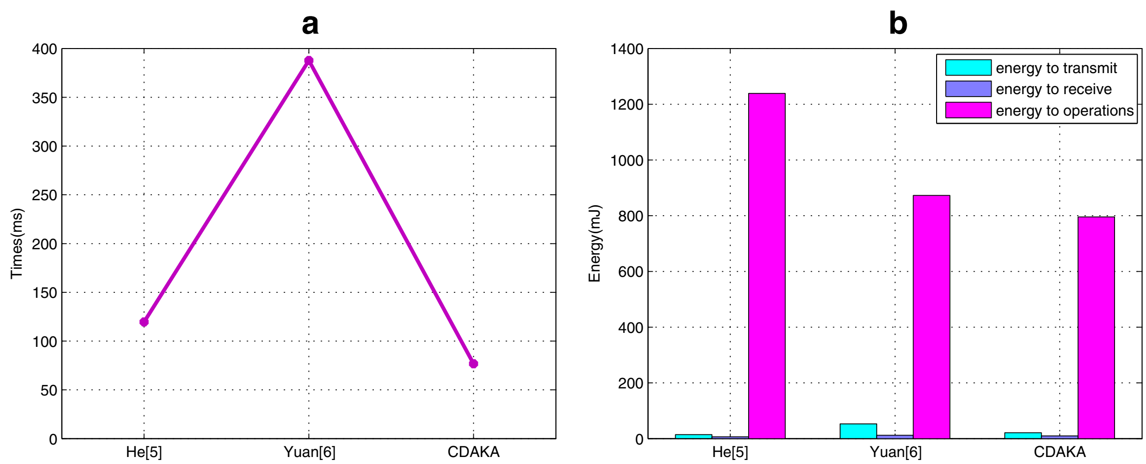


Fig. 5 Performance comparisons of related lightweight AKA protocol

registration centers. Hence, the running time is 0 ms. The total time is 38.406 + 38.406 = 76.812 ms

According to the above comparisons of computation cost, we know that the CDAKA protocol has much less running time than other two related AKA protocols [5, 6] in both sides of session initiator and session response.

Communication cost

In this subsection, we analyze and compare the communication costs of the CDAKA protocol and other two related AKA protocols [5, 6]. Because the size of P is 512 bits, then the size of an element in G_1.

Without loss of generality, let the sizes of an element in G_1, bilinear paring's value, signature value, encryption/decryption value is 512 bits. The size of the length of the pseudo identity is 128 bits. The size of the general hash functions output is 160 bits. The size of current timestamp is 32 bits.

In He et al.'s [5] protocol, among the interactive messages, there are six elements in G_1, two outputs of the general hash function and four pseudo identities. Therefore, the communication cost of He et al.'s [5] protocol is 6 * 512 + 2 * 160 + 4 * 128 = 3904 bits.

In Yuan et al.'s [6] protocol, among the interactive messages, there are two elements in G_1 and twenty-two encryption/decryption values. Therefore, the communication cost of Yuan et al.'s [6] protocol is 1 * 512 + 22 * 512 = 11776 bits.

In CDAKA protocol, among the interactive messages, there are ten elements in G_1, two outputs of the general hash function, two pseudo identities and two timestamp. Therefore, the communication cost of 10 * 512 + 2 * 160 + 2 * 128 + 2 * 32 = 5760 bits.

According to the above comparisons, we know that the CDAKA protocol increases the communication cost compared with He et al.'s [5] protocol. The reason for the increases is that CDAKA really implement authentication for multi-domain as Yuan et al.'s [6] protocol. It is worthy to achieve cross-domain authentication at the cost of increasing computation cost only. However, compared with Yuan et al.'s [6] protocol, the communication cost is greatly reduced.

Energy cost

In mobile devices, energy-saving is an important indicator. Here, we only discuss the client side or session initiator side from three part: energy to transmit, energy to receive and energy to operations, as shown in Fig. 5b.

From the above, in client side of He et al.'s [5] protocol, it needs to transmit two pseudo identities, three elements in G_1 and one output of the general hash function, total 1952 bits. According to [31, 34], it costs 14.44 mJ. It receives two

Table 4 Performance comparison among relevant authentication protocols

	He [5]	Yuan [6]	CDAKA
Computation cost of session initiator	59.812 ms	123.856 ms	38.406 ms
Computation cost of session response	59.812 ms	50.05 ms	38.406 ms
Computation cost of TA	0 ms	98.2 ms	0 ms
Computation cost of CA	0 ms	115.656 ms	0 ms
Communication cost/bit	3904	11776	5760

pseudo identities, three elements in G_1 and one output of the general hash function, total 1952 bits, which costs 6.56 mJ. The operations are six multiplication operations, two hash-to-point operations and three general hash functions. The energy is 1238.75 mJ.

In client side of Yuan et al.'s [6] protocol, it needs to transmit two elements in G_1 and twelve encryption/decryption values, total 7168 bits, which costs 53.04 mJ. It receives seven encryption/decryption values, total 3584 bits, which costs 12.05 mJ. The operations are one scalar bilinear paring operation, four multiplication operations, one signature generation operation, thirteen encryption/decryption operations and one hash-to-point operation. The energy is 872.75 mJ.

In client side of CDAKA protocol, it needs to transmit five elements in G_1, one outputs of the general hash function, one pseudo identities and one timestamp, total 2880 bits, which costs 21.13 mJ. It receives five elements in G_1, one outputs of the general hash function, one pseudo identities and one timestamp, total 2880 bits, which costs 9.684 mJ. The operations are four multiplication operations, one hash-to-point operation and foue general hash functions. The energy is 795.67 mJ.

According to the above comparisons, we know that the CDAKA protocol is energy-saving, which is very suitable for mobile application scenarios, where resource is severely constrained.

Security comparisons

To show the security advantages of CDAKA protocol, we present security comparisons between CDAKA protocol and other two related AKA protocols [5, 6]. The security comparisons are listed in Table 5. From Table 5, we can get that the protocol in [5] cannot provide cross-domain authentication and the the protocol in [6] cannot provide traceability. The CDAKA protocol can satisfy all ten security and function requirements. Therefore, the CDAKA protocol is more secure than other two related AKA protocols.

Table 5 Security features comparison among related authentication protocols

	He [5]	Yuan [6]	CDAKA
Mutual authentication and key agreement	Yes	Yes	Yes
Patient anonymity	Yes	Yes	Yes
Patient traceability	Yes	No	Yes
Cross-domain communication	No	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes
Resistance to impersonation attack	Yes	Yes	Yes
Resistance to internal attacks	Yes	Yes	Yes
Resistance to replay attack	No	Yes	Yes
Resistance to man-in-the-middle attack	Yes	Yes	Yes
Provable security	Yes	No	Yes

Conclusion and ongoing work

System security and patients privacy-preserved are a challenging issue in distributed medical heterogeneous cross-domain authentication systems. A provably-secure heterogeneous cross-domain authenticated key agreement protocol with symptoms-matching in TMIS presented in this paper is trying to find a balance between the system security and patients privacy-preserved. The CDAKA protocol investigates a systematic approach of heterogeneous cross-domain authentication from PKI-domain to IBC-domain or from IBC-domain to PKI-domain. Only the register centers PKI and IBC know patients' identities, it not only realizes anonymity to protect patient's privacy, but also addresses other prominent issues (e.g. patient traceability). Meanwhile the CDAKA protocol is proven to be secure under the Elliptic Curve Computable Diffie-Hellman problem (ECDHP) assumption in the random oracle model.. Compared with the recently relevant schemes, the CDAKA protocol has better performance (such as energy-saving) and better security features. Thus, CDAKA protocol is more secure and efficient for computation-limited mobile device. The future work is to fully identify the practical threats on heterogeneous cross-domain authentication protocols. Based on artificial intelligence, develop concrete heterogeneous cross-domain authentication with better performance.

Funding Information This study was funded by National Science Foundation of China under grant (61373171), The 111 Project under grant(B08038), National Key R&D Program of China(2017YFB0802400).

Compliance with Ethical Standards

Conflict of interest Author Xiaoxue liu declares that she has no conflict of interest. Author Wenping Ma declares that he 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. Tonetti, M., Jepsen, S., Jin, L., et al., Impact of the global burden of periodontal diseases on health, nutrition and wellbeing of mankind: A call for global action. *J. Clin. Periodontol.* 44(5):456–462, 2017.

2. Yang, Y., Zheng, X., Liu, X., et al., Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system. *Futur. Gener. Comput. Syst.* 84:160–176, 2017.
3. He, D., and Zeadally, S., Authentication protocol for an ambient assisted living system. *IEEE Commun. Mag.* 53:71–77, 2015.
4. Zhang, L., Zhang, Y., Tang, S., et al., Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement. *IEEE Trans. Ind. Electron.* 65(3):2795–2805, 2018.
5. He, D., Kumar, N., Wang, H., et al., A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network. *IEEE Trans. Dependable Secure Comput.* 13(9):1–13, 2016.
6. Yuan, C., Zhang, W., and Wang, X., EIMAKP: Heterogeneous cross-domain authenticated key agreement protocols in the EIM system. *Arab. J. Sci. Eng.* 42:3275–3287, 2017.
7. Wu, Z., Lee, Y., Lai, F., et al., A secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(3):1529–1535, 2012.
8. He, D., Cao, J., and Zhang, R., A more secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(3):1989–1995, 2012.
9. Wei, J., Hu, X., and Liu, W., An improved authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3597–3604, 2012.
10. Zhu, Z., An efficient authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3833–3838, 2012.
11. Lee, T., Chang, I., Lin, T., et al., A secure and efficient password-based user authentication scheme using smart cards for the integrated epr information system. *J. Med. Syst.* 37(3):3833–3838, 2013.
12. Tan, Z., An efficient biometrics-based authentication scheme for telecare medicine information systems. *Netw.* 2(3):200–204, 2013.
13. Yan, X., Li, W., Li, P., et al., A secure biometrics-based authentication scheme for telecare medicine information systems. *J. Med. Syst.* 37(5):1–6, 2013.
14. Chaudhry, S., Naqvi, H., and Khan, M., An enhanced lightweight anonymous biometric based authentication scheme for TMIS. *Multimed. Tools Appl.* 77(5):5503–5524, 2018.
15. Amin, R., and Biswas, G., A novel user authentication and key agreement protocol for accessing multi-medical server usable in TMIS. *J. Med. Syst.* 39(3):1–17, 2013.
16. Das, A., Odelu, V., and Goswami, A., A Secure and robust user authenticated key agreement scheme for hierarchical multi-medical server environment in TMIS. *J. Med. Syst.* 39(9):1–24, 2015.
17. Liu, X., Li, Y., et al., PAKA: A lightweight pseudonym authentication and key agreement protocol for multi-medical server architecture in TMIS. *KSII Trans. Internet Inf. Syst.* 11(2):924–944, 2017.
18. Sun, Y., and Li, H., Efficient signcryption between TPKC and IDPKC and its multi-receiver construction. *Sci. China Inf.* 53(3):557–566, 2010.
19. Huang, Q., and Wong, D., Heterogeneous signcryption with key privacy. *Comput. J.* 54(4):525–536, 2011.
20. Karati, A., Islam, S., and Karuppiah, M., Provably secure and lightweight certificateless signature scheme for IIoT environments. *IEEE Trans. Ind. Inf.* 1–11, 2018.
21. Ma, M., He, D., Kumar, N., et al., Certificateless searchable public key encryption scheme for industrial internet of things. *IEEE Trans. Ind. Inf.* 14(2):759–767, 2018.
22. Li, Y., and Wang, C., Privacy-preserving multi-receiver signcryption scheme for heterogeneous systems. *Secur. Commun. Netw.* 9(17):4574–4584, 2016.
23. Li, Y., Chen, W., Cai, Z., et al., CAKA: A novel certificateless-based cross-domain authenticated key agreement protocol for wireless mesh networks. *Wirel. Netw.* 22(8):2523–2535, 2016.
24. Wang, C., Liu, C., Niu, S., et al., An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme. In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 723–728: IEEE, 2017.
25. Zhang, Q., Gan, Y., Zhang, Q., et al., A dynamic and cross-domain authentication asymmetric group key agreement in telemedicine application. *IEEE Access* 6:24064–24074, 2018.
26. Chen, Q., Shi, S., Li, X., et al., SDN-based privacy preserving cross domain routing. IEEE Trans. *Dependable Secure Comput.* 1–13, 2018.
27. Luo, M., Luo, Y., Wan, Y., et al., Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT. *Secur. Commun. Netw.* 2018:1–10, 2018.
28. Al-Riyami, S., and Paterson, K., Certificateless public key cryptography. In: *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 452–473. Berlin: Springer, 2003.
29. Ma, M., He, D., Khan, M., et al., Certificateless searchable public key encryption scheme for mobile healthcare system. *Comput. Electr. Eng.* 65:413–424, 2017.
30. Kilinc, H., and Yanik, T., A survey of SIP authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* 16(2):1005–1023, 2014.
31. Wander, A., Gura, N., Eberle, H., et al., Energy analysis of public-key cryptography for wireless sensor networks. In: *Third IEEE International Conference on Pervasive Computing and Communications, PerCom*, 2005.
32. Huang, D., Misra, S., Verma, M., et al., PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs. *IEEE Trans. Intell. Transp. Syst.* 12(3):736–746, 2011.
33. He, D., Zeadally, S., Kumar, N., et al., Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Trans. Inf. Forensics Secur.* 11(9):2052–2064, 2016.
34. De Meulenaer, G., Gosset, F., Standaert, F. X., et al., On the energy cost of communication and cryptography in wireless sensor networks, pp. 580–585, 2008.