Digital Object Identifier

Privacy-aware Secure Anonymous Communication Protocol in CPSS Cloud Computing

FENGYIN LI[1], CAN CUI[2], DONGFENG WANG[1], ZHONGXING LIU[1], NEBRASE ELMRABIT[3], YING WANG[1], HUIYU ZHOU[3].
[1]School of Information Science and Engineering, Qufu Normal University, Rizhao, Shandong, 276826, P.R. China.
[2]School of Cyber Science and Engineering, Shandong University of Political Science and Law, Jinan, Shandong, 250014, P.R. China.
[3] School of Informatics, University of Leicester, University Road, Leicester, LE1 7RH, UK.

Corresponding author, Huiyu Zhou, (e-mail: hz143@leicester.ac.uk)

ABSTRACT Cloud computing has emerged as a promising paradigm for the Internet of Things (IoT) and Cyber-Physical-Social Systems (CPSS). However, the problem of how to ensure the security of data transmission and data storage in CPSS is a key issue to address. We need to protect the confidentiality and privacy of users' data and users' identity during the transmission and storage process in CPSS. In order to avoid users' personal information leakage from IoT devices during the process of data processing and transmitting, we propose a certificateless encryption scheme, and conduct a security analysis under the assumption of Computational Diffie-Hellman(CDH) Problem. Furthermore, based on the proposed cryptography mechanism, we achieve a novel anonymous communication protocol to protect the identity privacy of communicating units in CPSS. In the new protocol, an anonymous communication link establishment method and an anonymous communication packet encapsulation format are proposed. The Diffie-Hellman key exchange algorithm is used to construct the anonymous keys distribution method in the new link establishment method. And in the new onion routing packet encapsulation format, the session data are firstly separated from the authentication data to decrease the number of cryptography operations. That is, by using the new onion routing packet we greatly reduces the encryption operations and promotes the forwarding efficiency of anonymous messages, implementing the privacy, security and efficiency in anonymous communication in cyber-physical-social systems.

INDEX TERMS Internet of Things, Cloud Computing, CPSS, Privacy Protection, Anonymous Communication

I. INTRODUCTION

CLOUD computing has emerged as a promising paradigm for the Internet of Things (IoT) and Cyber-Physical-Social Systems (CPSS) [1]. Cloud computing gives rise to research challenges spanning from design and implementation of mechanisms to trust and identity management in federated environments [2]. A central challenge is the effective security and privacy mechanism to assure the data confidentiality and identity privacy during the communication between highly distrusted mobile devices in CPSS.

IoT devices collect user data, transmit and store them on the cloud server, for later data analysis, PublicLeaks-based criminal smart contracts can be used to protect private data [3]. However, in most cases, user data collected by IoT devices are transmitted and presented in the form of plaintext in the network and the cloud servers, especially those sensitive data in enterprise information system [4]. This causes the collected user data to be maliciously modified or illegally exploited. The attacker may extract confidential information of the users from the exposed data.

How to protect the privacy of user data to assure the rapid and secure network communication is one of the key issues to be solved in public communication environments. Criminal smart contracts were used to prevent criminals' illegal behaviors [5]. Locality-Sensitive Hashing (LSH) were used to provide privacy-preserving and scalable mobile service recommendation [6]. In the next generation of Internet architectures, security has become part of the network architecture, rather than being achieved through additional networks layers on the existing Internet [7] [8]. In the next generation network architecture, anonymous communication privacy protection is implemented by encrypting the address data which representing the user's identity [9]. The entire data packet of the user is securely forwarded anonymously through the anonymous communication link, ensuring the identity privacy of the users [10] [11]. The user identity is not visible. However, because the network device needs to know some information about the user to route and forward message, users' identity cannot be completely hidden in many kinds of applications [12] [13] [14] [15]. Therefore, anonymous communication hides important information from untrusted parties and discloses necessary information to the licensor or trusted party. By using an encryption technology, anonymous communication hides the user's personal information in the communication process, such as user identity, network location, etc.

Currently, many protocols have tried to implement user identity anonymity. In 2008, Wang et al presented a communication protocol (SA-MAC) based on unconditional secure and efficient source anonymous message authentication code, which can hide the identity of both senders and receivers [16]. while the authentication overload of this protocol is relatively high. Furthermore, it only focuses on the authentication process of the message, and the link structure is mainly ring-shaped, and the protocol lacks flexibility. In 2012, Chen et al presented an effective anonymous communication protocol (EAC) for wireless networks, which guarantees the anonymity of the protocol from three angles, and the scheme only uses a hash function and symmetry encryption with less computational overhead [17]. In 2014, Ardagna et al presented a solution that allows smartphone users to implement an end-to-end anonymous communication protocol in a mobile cloud computing environment, which solves the mobile privacy problem generated in mobile cloud computing scenarios [18]. In 2015, Lo et al presented a new anonymous secure communication routing protocol (HASR. HASR can implement anonymous communication in Mobile Ad hoc network (MANET) environment [19]. Routing efficiency is improved by omitting a separate anonymous communication link establishment process, but messages security is difficult to guarantee.

Onion routing is also the most commonly used method in current anonymous communication networks [20]. In 2004, Dingledine et al introduced the second generation of onion router Tor, which added a directory server to achieve forward secrecy and protect the user's location privacy through relay node sets [21] [22]. In the protocol proposed by Camenisch in 2005, each node only knows its previous node and the posterior node on the anonymous communication path, but the anonymity of this protocol is not acquired [23]. In 2007, Aaron presented an IO-automata model of an onion routing protocol, describing the case of guaranteeing protocol anonymity and unlinkability [24]. In 2019, Piccialli et al proposed an anonymous network architecture for on-board and mobile devices in an onion network based on P2P (Peer-to-Peer) technology [25] [26]. However, the server in the solution becomes the system bottleneck and is more vulnerable to attacks. The certificateless onion routing protocol proposed in 2017 greatly improved the computing load of the relay router and obtained a higher data transmission rate [27] [28]. However, its packet structure cannot counter the data tampering attack of the relay routing nodes.

In the next-generation network architecture, not only the user's identity privacy and data security but also the communication efficiency of the system should be assured. This is a key issue in the anonymous communication network. The proposed protocol in this paper is implemented to address the issue.

The main contributions of this paper are as follows.

(1) To avoid users' personal information leakage from IoT devices, we propose a certificateless encryption scheme, and conduct a security analysis under the assumption of Computational Diffie-Hellman(CDH) Problem.

(2) To improve the messages forwarding efficiency, we propose a new anonymous communication packet encapsulation format, effectively decreasing the number of unnecessary cryptography operations.

(3) To achieve the identity privacy, an anonymous communication link establishment method is proposed.

(4) We complete an anonymous protocol, implementing the privacy, security and efficiency in anonymous communication in cyber-physical-social systems.

II. PRELIMINARY

Definition 1 Discrete Logarithm Problem (DL Problem)

Let G be a finite cyclic group with prime order q, and g is one of its generator. If given g and $g^a mod q$, computing $a \in Z_q$ is difficult.

Definition 2 Computational Diffie-Hellman Problem (CDH Problem)

Let G be a finite cyclic group with prime order q, and g is one of its generator. If given $g^u mod q$ and $g^v mod q$, computing $y = g^{uv} mod q$ is computationally difficult.

Definition 3 Security Model

Assume that A_I and A_{II} represent type I attacker and type II attacker, respectively. A_I can perform partial private key query, public key query, private key query, and public key replacement query operation, but does not know the system master key. A_{II} has the system master key, which can perform partial private key query, public key query, private key query as A_I but can not perform public key replacement query.

III. NEW CERTIFICATELESS ENCRYPTION SCHEME

This paper designs a new certificateless encryption scheme applicable to anonymous communication packet encapsulation format and anonymous communication link establishment process as follows.

A. CERTIFICATELESS ENCRYPTION SCHEME

The certificateless encryption scheme of this paper consists of the following five sub-algorithms.

Setup: KGC inputs the security parameter l to generate a cyclic multiplication group G with a prime order q (length is 2^l) and a generator g. KGC chooses three hash functions $H_1 : \{0,1\}^* \times G \rightarrow Z_q$, $H_2 : G \rightarrow \{0,1\}^*$. KGC randomly selects $sk_{sys} \xleftarrow{\$} Z_q$ as the system master key, obtains $pk_{sys} = g^{sk_{sys}} modq$ as the system public key. The system public parameter is $param = (q, g, G, H_1, H_2, pk_{sys})$. The message $m \in \{0,1\}^*$ is plaintext space, and the ciphertext $c \in C$ is ciphertext space.

Partial Private Key Extraction: KGC generates a partial private key for the user after verifying that the identity of the user ID is legal. KGC randomly selects $k_{ID} \xleftarrow{\$} Z_q$ and calculates $s_{D_{ID}} = g^{k_{ID}} modq$. Let $r_{D_{ID}} = k_{ID} + sk_{sys} \cdot H_1(ID, s_{D_{ID}}) modq$, gets a partial private $d_{ID} = (s_{D_{ID}}, r_{D_{ID}})$, KGC sends it to the user ID through the secure channel.

Set Public-Private Keys: After receiving the $d_{ID} = (s_{D_{ID}}, r_{D_{ID}})$ from KGC, the user runs this key generation algorithm to generate his own public-private key pair. He picks a random value $t_{ID} \xleftarrow{\$} Z_q$ and computes $u_{ID} = g^{t_{ID}} modq$. Then, the public key of the user is $pk_{ID} = (s_{D_{ID}}, u_{ID})$, and the private key is $sk_{ID} = (r_{D_{ID}}, t_{ID})$. The user transmits his public key to the KGC through a secure channel and KGC publishes it in the public information list. The user keeps the private key $sk_{ID} = (r_{D_{ID}}, t_{ID})$ secret.

Encryption: The user wishes to encrypt the information m. Firstly, he randomly chooses $\omega_{ID} \xleftarrow{\$} Z_q$, calculates $W = g^{\omega_{ID}} modq$. The user encrypts the information and calculates $c_1 = (s_{D_{ID}} \cdot pk_{sys}^{H_1(ID, s_{D_{ID}})})^{\omega_{ID}}$, $c_2 = u_{ID}^{\omega_{ID}} modq$, $c_3 = H_2(c_1 \cdot c_2) \oplus m$ mod q by using the public key of the recipient, and transmits the ciphertext $c = (W, c_3)$ to the recipient.

Decryption: The recipient decrypts the ciphertext $c = (W, c_3)$ with his private key, calculates $c_1 = W^{r_{D_{ID}}}$, $c_2 = W^{t_{ID}}$ and obtains the message plaintext $m = c_3 \oplus H_2(c_1 \cdot c_2)$.

B. PERFORMANCE ANALYSIS

Correctness

The correctness of the certificateless encryption scheme can be proved by the following equation. Using the sender's public key $pk_{ID} = (s_{D_{ID}}, u_{D_{ID}})$, the original message plaintext m can be obtained.

$$\begin{aligned}
& c_3 \oplus H_2(c_1 \cdot c_2) \\
& = c_3 \oplus H_2(W^{r_{D_{ID}}} \cdot W^{t_{ID}}) \\
& = c_3 \oplus H_2((g^{\omega_{ID}})^{k_{ID} + sk_{sys} H_1(ID, s_{D_{ID}})} \cdot (g^{\omega_{ID}})^{t_{ID}}) \quad (1) \\
& = c_3 \oplus H_2((s_{D_{ID}} \cdot pk_{sys}^{H_1(ID, s_{D_{ID}})}) \cdot u_{ID}^{\omega_{ID}}) \\
& = m
\end{aligned}$$

Security proof

It is assumed that Type I attacker A_I and Type II attacker A_{II} can break the encryption scheme of this section. The game plays between Challenger C and A_I or A_{II} until C derives the solution to the CDH problem based on the interaction results, breaking the security assumption of the CDH problem. According to the security proof requirements shown in references [13] [19], attackers A_I and A_{II} can adaptively perform various queries such as random oracle query, partial private key extraction, public key request, private key extraction, and decryption query. C will respond to these queries according to the distribution of responses in real attacks.

For the attacker A_I, C sets $g^u modq$ as part of the challenge ciphertext and $g^v modq$ as the system public key. For the attacker A_{II}, C sets $g^u modq$ as part of the challenge ciphertext and uses $g^v modq$ to generate the public key associated with the challenge identity ID.

The challenger C selects the system master key $sk_{sys} \in Z_q$, calculates the system public key $pk_{sys} = g^{sk_{sys}} modq$, and C sends the system key sk_{sys} to A_{II} .

Theorem 1 Under the security assumption of the CDH problem, if H_1 and H_2 are random oracles, the certificateless encryption scheme of this paper has type I security.

Proof Suppose there is a type I attacker A_I can break the encryption scheme of this paper with a non-negligible probability ε within Probabilistic Polynomial Time (PPT), we prove that using the ability of A_I, C can construct a method to solve the CDH problem. From the security assumptions of the CDH problem, it is impossible for A_I to break this scheme.

Assume that A_I raise queries to random oracles in finite polynomial time. In the following steps, we demonstrate that using A_I, Challenger C successfully constructs a solution to the CDH problem.

Setup

The challenger C sets $pk_{sys} = g^v modq$ and passes the system parameters $params = (q, g, G, H_1, H_2, pk_{sys})$ to A_I, where H_1, H_2 are random oracles controlled by C.

Phase 1

A_I performs the following oracle queries, Challenger C responds to the queries from A_I.

H_1 Query: For a given (ID,$s_{D_{ID}}$), Challenger C prepares a hash list L_{H_1} to record all H_1 hash queries and responds as follows, where the hash list is empty at the beginning.

If $\langle (ID, s_{D_{ID}}), \alpha \rangle$ exists in L_{H_1}, C returns α as the answer.

Otherwise, randomly selects $\alpha \xleftarrow{\$} Z_q$, adds $\langle (ID, s_{D_{ID}}), \alpha \rangle$ to L_{H_1} and returns α as the answer.

H_2 Query: For a given $(c_1 \cdot c_2)$ query, Challenger C prepares a hash list L_{H_2} to record all H_2 hash queries and responds as follows, where the hash list is empty at the beginning.

If $\langle (c_1 \cdot c_2), k \rangle$ exists in L_{H_2}, C returns k as the answer.

Otherwise, randomly selects $k \xleftarrow{\$} \{0,1\}^l$, adds $\langle (c_1 \cdot c_2), k \rangle$ to L_{H_2} and returns k as the answer.

Partial Private Key Query: For a given ID, Challenger C responds to partial private key queries as follows.

If the $\langle ID, (s_{D_{ID}}, r_{D_{ID}}) \rangle$ exists in the list L_{par}, C returns $d_{ID} = (s_{D_{ID}}, r_{D_{ID}})$ as the answer.

Otherwise, C proceeds as follows.

By randomly selects $r_{D_{ID}} \xleftarrow{\$} Z_q$, C calculates $s_{D_{ID}} = g^{r_{D_{ID}}} g^{-\alpha} modq$. Adds $\langle (ID, s_{D_{ID}}), \alpha \rangle$ to L_{H_1} and add $\langle ID, (s_{D_{ID}}, r_{D_{ID}}) \rangle$ to L_{par}. Returns $d_{ID} = (s_{D_{ID}}, r_{D_{ID}})$ as the answer.

Public Key Request Query: For a given ID, Challenger C responds to the public key request query as follows.

If the $\langle ID, (s_{D_{ID}}, u_{D_{ID}}), coin \rangle$ exists in L_{pub}, C returns $pk_{ID} = (s_{D_{ID}}, u_{D_{ID}})$ as the answer.

Otherwise, C selects $coin \in \{0, 1\}$ such that $Pr[coin = 0] = \delta$.

$coin \in \{0, 1\}$ is a guess as to whether ID is the challenge identity chosen by the attacker during the challenge phase. $coin = 0$ means that ID is not a challenging identity. $coin = 1$ represents that ID is the challenging identity.

When $coin = 0$, C does as follows.

If the $\langle ID, (s_{D_{ID}}, r_{D_{ID}}) \rangle$ exists in the list L_{par}, C randomly selects $t_{ID} \xleftarrow{\$} Z_q$, calculates $u_{ID} = g^{t_{ID}} modq$, adds $\langle ID, (r_{D_{ID}}, t_{ID}) \rangle$ to the list L_{priv} , adds $\langle ID, (s_{D_{ID}}, u_{ID}), coin \rangle$ to the list L_{pub}, and returns $pk_{ID} = (s_{D_{ID}}, u_{ID})$ as the answer.

When $coin = 1$, C does as follows. randomly selects $k_{ID}, t_{ID} \xleftarrow{\$} Z_q$, calculates $s_{D_{ID}} = g^{k_{ID}} modq$, $u_{ID} = g^{t_{ID}} modq$, adds $\langle ID, (t_{ID^*}), k_{ID} \rangle$ to L_{pub}, adds $\langle ID, (t_{ID^*}), k_{ID} \rangle$ to L_{priv}, adds $\langle ID, (s_{D_{ID}}, k_{ID}) \rangle$ to L_{par}, and gives $pk_{ID} = (s_{D_{ID}}, u_{ID})$ as a response.

Private Key Extraction Query: For a given ID, challenger C responds to the private key extraction query as follows.

C Runs the above public key query algorithm with ID as input to get $\langle ID, (s_{D_{ID}}, u_{ID}), coin \rangle \in L_{pub}$.

When $coin = 0$, C finds $\langle ID, (r_{D_{ID}}, t_{ID}) \rangle$ in L_{priv} and returns $sk_{ID} = (s_{D_{ID}}, r_{D_{ID}})$ as the answer. Otherwise, C returns "abort" and the algorithm ends. In this case, C cannot calculate $r_{D_{ID}}$.

Public Key Replacement Query: A_I sets $pk_{ID} = ({s_{D_{ID}}}', {u_{ID}}')$. To respond a public key replacement, C finds the $\langle ID, (s_{D_{ID}}, u_{ID}), coin \rangle$ in L_{pub} , replaces $(s_{D_{ID}}, u_{ID})$ with $({s_{D_{ID}}}', {u_{ID}}')$, and marks the tuple as "updated".

Decryption Query: For a given ciphertext $c = (W, c_3)$, Challenger C responds to the decryption query in the following manner.

C finds $\langle ID, (s_{D_{ID}}, u_{ID}), coin \rangle$ in L_{pub} . If it does not exist, C runs the above public key query algorithm to generate a public key $pk_{ID} = (s_{D_{ID}}, u_{ID})$ for the user ID .

If it exists and $coin = 0$, it is obvious that C knows the private key $sk_{ID} = (s_{D_{ID}}, r_{D_{ID}})$ of the corresponding ciphertext and can calculate the corresponding plaintext m as follows. Calculates $c_1 = W^{r_{D_{ID}}} modq$, $c_2 = W^{t_{ID}} modq$, and decrypts the plaintext $m = c_3 \oplus H_2(c_1 \cdot c_2)$ as a response to the decryption query. Otherwise, if it exists and $coin = 1$ (C does not know $r_{D_{ID}}$), C can also calculate the plaintext as follows.

C first calculates $c_2 = W^{t_{ID}} modq$, runs the H_1 query to get the $\langle (ID, s_{D_{ID}}), \alpha \rangle$, and then calculates $u_{ID}^{\omega_{ID}} \cdot (s_{D_{ID}} \cdot pk_{sys}^{\alpha})^{\omega_{ID}} modq = c_1 \cdot c_2$.

If the challenger C gets $\langle (c_1 \cdot c_2, k) \rangle$ in the L_{H_2}, so that $c = k \oplus m$, $c_1 \cdot c_2 = u_{ID}^{\omega_{ID}} \cdot (s_{D_{ID}} \cdot pk_{sys}^{\alpha})^{\omega_{ID}} modq = c_1 \cdot c_2$, then C can calculate the laintext $m = c_3 \oplus k$ and returns it to the attacker. And then C calculates $\hat{c_1} = (c_1 / W^{k_{ID^*}})^{\alpha^{-1}} = g^{uv} modq$ as the solution to the CDH problem.

Otherwise, C returns "reject" and the algorithm stops.

Challenge

Suppose that attacker A_I sends an identity ID^*, two plaintexts (m_0,m_1)to challenger C. Suppose that <ID^*,(ρ,u_{ID^*}),coin> is the public key of ID^* in L_{pub}, and the <ID,($s_{D_{ID^*}}$,$r_{D_{ID^*}}$> is his partial private key in L_{par}.

If $coin = 0$, the output is "abort" and the algorithm stops.

Otherwise, if $coin = 1$, since the public key may be replaced by an attacker, A_I sets the value $\rho = g^{k_{ID^*}} = g^{\lambda} modq$ as the public part of the private key of ID^*. C finds <ID,(t_{ID^*},$r_{D_{ID^*}}$> in L_{priv} and selects a random value $b \epsilon \{0, 1\}$. At this point, C calculates $\alpha^* = H_1(ID^*, s_{D_{ID^*}})$,$w^* = g^u modq$, $c_1 = W^{*r_{D_{ID^*}}} = W^{*(\lambda + \nu \cdot \alpha)} modq$, $c_2 = W^{*t_{ID^*}} modq$, and then generates the challenge ciphertext $(W^*, c_3^*)$and sends it to A_I. Where $H_2((s_{D_{ID^*}} g^{\nu \cdot \alpha})^u \cdot$ $\mathrm{u}_{ID^*}^u)$ and $r_{D_{ID^*}}$ are the private keys of challenging identity, but Challenger C does not know $r_{D_{ID^*}}$.

Phase 2

At this step, Challenger C can continue to respond various query operations according to the previous challenge results, but does not allow A_I to perform a private key query for the challenging identity ID^*.

Guess

Based on the previous challenges and query results,A_I outputs a guess for b. If an attacker A_I can decrypt challenge (W^*, c_3^*) to get the correct plaintext message m_0 or m_1, then it must have done H_2 query to get k. Therefore, C can find $(c_1 \cdot c_2)$ and the corresponding k in L_{H_2}, then A_I can output the guess b.

Analysis

If the attacker A_I makes H_2 query but does not perform the decryption query $(s_{D_{ID^*}} \cdot g^{b \cdot \alpha})^a \cdot {u_{ID^*}}^a$ and the private key query of the challenging identity ID^*, the challenger C does not abort the response during the game, and A_I can get the true plaintext m_0 or m_1 of the corresponding challenging ciphertext (W^*, c_3^*), which means that attackerA_I calculates the true value of $c_1 = g^{((\lambda + \nu \cdot \alpha) \cdot u)}$.

Suppose that the advantage of A_I breaks this encryption scheme type I security is ε. As long as C does not abort, Challenger C can use the ability of A_I to calculate the solution $\hat{c_1} = (c_1 / W^{k_{ID^*}})^{\alpha^{-1}} = g^{u\nu}$ of the CDH problem. Therefore, the probability that Challenger C does not abort is the probability of C to successfully break the security assumption of CDH.

Assume that the number of times that A_I does private key query is q_{priv}. The condition of that C does not stop during the challenging phase is $coin = 1$, the happening probability of which is $(1 - \delta)$. In addition, the condition of abort during the private key extraction query phase is $coin = 0$, the happening probability of which is δ. Because the number of private key queries is q_{priv}, the probability that the simulation algorithm does not stop during the entire private key extraction query phase is $\delta^{q_{priv}}$. Therefore, the probability that the whole simulation process will not stop is$(1-\delta)\delta^{q_{priv}}$. That is, if A_I can break the encryption scheme with a non-negligible probability ε within PPT, the probability that challenger C successfully breaks the CDH problem is $\varepsilon(1-\delta)\delta^{q_{priv}}$, which is non-negligible.

According to the security assumption of the CDH problem, the certificateless encryption scheme proposed in this paper has type I security.

Theorem 2 Under the security assumption of the CDH problem, if H_1 and H_2 are random oracles, then the certificateless encryption scheme of this paper has type II security.

Proof Suppose there is an attacker A_{II} who breaks the encryption mechanism of this paper with probability ε in the polynomial time. In the following steps, we prove that by using the ability of A_{II}, C can obtain the solution of the CDH problem with a non-negligible probability. That is, by simulating a type II security game, for a given input params=(q,G,g,g^u, g^{ν}), C can obtain $V = g^{uv}$ as a solution to the CDH problem.

Setup

Challenger C randomly selects $sk_{sys} \xleftarrow{\$} Z_q$ and calculates the public key $pk_{sys} = g^{sk_{sys}} modq$. C generates the public parameter params=(q,g,G,pk^{sys}, H_1, H_2) and sends it to A_{II}. C sends the master key sk^{sys} to A_{II} through a secure channel.

Phase 1

A_{II} performs the following random query, which is responded by the challenger C.

H_1 Query:For a given input(ID,$s_{D_{ID}}$), Challenger C responds to the H_1 query as follows.

If <(ID,$s_{D_{ID}}$),α> exists in L_{H_1}, C returns α as the answer.

Otherwise, C randomly selects $\alpha \xleftarrow{\$} Z_q$, and adds <(ID,$s_{D_{ID}}$),$\alpha$> to L_{H_1} and returns α as the answer.

H_2 Query: For a given input(c_1, c_2), Challenger C responds to the H_2 query as follows.

If<($c_1 \cdot c_2$),k> exists in L_{H_2}, C returns k as the answer.

Otherwise, C randomly selects $k \xleftarrow{\$} \{0,1\}^l$ and adds $\langle(c_1 \cdot c_2, k)\rangle$ to L_{H_2} and returns k as the answer.

Partial Private Key Query: For a given input ID, Challenger C responds to partial private key queries as follows.

If $\langle ID,(s_{D_{ID}}, r_{D_{ID}})\rangle$ exists in the list L_{par}, C returns $d_{ID} = (s_{D_{ID}}, r_{D_{ID}})$ as the answer. Otherwise, C proceeds as follows.

Using the system key sk_{sys}, C generates a partial private key for the user ID as follows .

C selects $k_{ID} \xleftarrow{\$} Z_q$ randomly and computes $s_{D_{ID}} = g^{k_{ID}} modq$. C looks up the list L_{H_1} to get the $\langle(ID, s_{D_{ID}}), \alpha\rangle$. Lets $r_{D_{ID}} = k_{ID} + sk_{sys} \cdot H_1(ID, s_{D_{ID}}) modq$. Adds $\langle ID,(s_{D_{ID}}, r_{D_{ID}})\rangle$ to L_{par}. At last, C returns $(s_{D_{ID}}, r_{D_{ID}})$ as the response.

Public Key Request Query: For a given input ID, Challenger C responds to the public key request query as follows:

If $\langle ID,(s_{D_{ID}}, u_{ID}), coin\rangle$ exists in L_{pub}, C returns $pk_{ID} = (s_{D_{ID}}, u_{ID})$ as the response.

Otherwise, C selects $coin \in \{0,1\}$ such that $Pr[coin = 0] = \delta$.

If $coin = 0$, C does the following steps.

If $\langle ID,(s_{D_{ID}}, r_{D_{ID}})\rangle$ exists in list L_{par} (otherwise, C needs to run the above partial private key query algorithm to get a partial private key $(s_{D_{ID}}, r_{D_{ID}})$.), C randomly selects $t_{ID} \xleftarrow{\$} Z_q$, and calculates $u_{ID} = g^{t_{ID}} modq$. Adds $\langle ID,(t_{D_{ID}}, r_{D_{ID}})\rangle$ to L_{priv} and adds $\langle ID,(s_{D_{ID}}, u_{ID}), coin\rangle$ to L_{pub}. At last, C returns $pk_{ID} = (s_{D_{ID}}, u_{ID})$ as the response.

If $coin = 0$, C randomly selects $t_{ID} \xleftarrow{\$} Z_q$, calculates $u_{ID} = (g^u)^{t_{ID}} modq$. C adds $\langle ID,(s_{D_{ID}}, u_{ID}), coin\rangle$ to L_{pub}, adds $\langle ID,(*, r_{D_{ID}}), t_{ID}\rangle$ to L_{priv}, and returns to $pk_{ID} = (s_{D_{ID}}, u_{ID})$ as the response.

Private Key Extraction Query: For a given input ID , Challenger C responds to the private key extraction query as follows.

First, C runs the public key query algorithm to get the public key $\langle ID,(s_{D_{ID}}, u_{ID}), coin\rangle \in L_{pub}$ of the identity ID .

If $coin = 0$, C looks up $\langle ID,(t_{ID}, r_{D_{ID}})\rangle$ in L_{priv} and returns $sk_{ID} = (t_{ID}, r_{D_{ID}})$ as the response. Otherwise, C returns "abort" and the algorithm stops.

Decryption Query: For a given ciphertext (W, c_3) , Challenger C responds the decryption query as follows.

C looks up $\langle ID,(s_{D_{ID}}, u_{ID}), coin\rangle$ in the L_{pub}. If it does not exist, C runs the above public key query algorithm to generate a public key for the user ID.

If $coin = 0$, C looks for $\langle ID,(t_{ID}, r_{D_{ID}})\rangle$ in L_{priv}, gets $sk_{ID} = (t_{ID}, r_{D_{ID}})$, and then runs the decryption query algorithm to get $c_1 = W^{r_{D_{ID}}}$, $c_2 = W^{t_{ID}}$, $m = c_3 \oplus H_3(c_1 \cdot c_2)$. At last, C returns plaintext m as a response.

If $coin = 1$, C calculates $c_1 = W^{r_{D_{ID}}}$, runs the H_1 query to get the $\langle(ID, s_{D_{ID}}), \alpha\rangle$.

If $\langle(c_1 \cdot c_2), k\rangle$ exists in L_{H_1}, C calculates $u_{ID}^{\omega_{ID}} = ((g^u)^{t_{ID}})^{\omega_{ID}} modq$, $W = g^{\omega_{ID}}$,$c_1 \cdot c_2 = u_{ID}^{\omega_{ID}} \cdot (s_{D_{ID}} \cdot pk_{sys}^{\alpha})^{\omega_{ID}}$, and then C performs an H_2 query. If Challenger C succeeds in doing a query in L_{H_2}, and gets $H_2(c_1 \cdot c_2)$, he returns plaintext $m = c_3 \oplus H_2(c_1 \cdot c_2)$ as the response.

Otherwise, C returns "reject" and the algorithm stops.

Challenge

The attacker A_{II} chooses a challenge identity ID^* and two plaintext messages (m_0, m_1) and sends them to Challenger C.

C performs a public key query in L_{pub} to get $\langle ID^*,(s_{D_{ID^*}}, u_{ID^*}), coin\rangle$, and if $coin = 1$, the algorithm continues. If $coin = 0$, the algorithm stops.

C looks up $\langle ID^*,(t_{ID^*}, r_{D_{ID^*}})\rangle$ in the L_{priv}. Selects random bit $b \in \{0,1\}$ and decides which plaintext to be encrypted. Sets $W^* = g^v$, $\alpha^* = H_2(ID^*, s_{D_{ID^*}})$, where $c_1 = W^{*r_{D_{ID^*}}}$, $c_2 = W^{*ut_{ID^*}} = (g^u)^{t_{ID^*}v}$,$c_3 = (W^{*r_{D_{ID^*}}} \cdot W^{*ut_{ID^*}}) \oplus m_b$, and sends (W^*, c_3^*) as the target ciphertext to the attacker A_{II}.

Phase 2

At this step, Challenger C continues to respond the various query operations of Phase 1 based on the results of the previous query, but does not allow A_{II} to make a public key replacement.

Guess

In the end, A_{II} outputs a guess bit b based on the previous query and challenge results. If attacker A_{II} can decrypt the ciphertext to get the corresponding plaintext, then he must have performed H_2 query.

Analysis

Suppose attacker A_{II} made an H_1 query on the $(ID^*, s_{D_{ID^*}})$ but did not do decryption query to $(s_{D_{ID}} \cdot pk_{sys}^{\alpha})^b \cdot {u_{ID^*}}^b$, and there is no private key query and Challenger C did not stop the algorithm. Suppose attacker A_{II} can break the encryption scheme of this paper and obtain the plaintext of the corresponding challenge ciphertext. At this time, Challenger C can use the ability of the attacker A_{II} to get $c_2^{1/t_{ID^*}} = g^{uv}$ as a solution to the CDH problem.

In order to analyse the probability that Challenger C successfully solves the CDH problem, suppose that the number of private key queries performed by A_{II} is q_{priv}, and the parameters given by type I security proof are used. The probability that C will not stop ($coin = 1$) is $(1-\delta)$ during the challenge phase. The probability that a single private key extraction query will not stop is δ, and the probability that the entire private key query phase will not abort is $\delta^{q_{priv}}$, so that the probability that the whole simulation process will not stop is $(1-\delta)\delta^{q_{priv}}$. That is, if A_{II} can break the encryption scheme with a non-negligible probability ε within PPT, the probability that challenger C successfully solves the CDH problem is $\varepsilon(1-\delta)\delta^{q_{priv}}$, which is non-negligible.

According to the security assumption of the CDH problem, the encryption scheme of this paper has Type II security.

IV. ANONYMOUS COMMUNICATION PROTOCOL BASED ON CERTIFICATELESS ENCRYPTION

Onion routing technology plays a vital role in anonymous communication networks, not only hiding the user's network location, but also resisting traffic analysis attacks [29] [30]. Based on the above encryption scheme, this paper proposes a new packet encapsulation format using the framework of onion routing protocol, and implements a new anonymous communication protocol, which can realize fast and secure anonymous message forwarding.

The anonymous communication protocol is divided into two parts: anonymous communication link establishment and anonymous message forwarding. The establishment of anonymous communication links is mainly to implement the distribution of shared keys [31]. The shared keys are divided into two categories, one is for encapsulation of onions to encrypt and decrypt operations during the establishment of anonymous communication links, and the other is for the encryption and decryption operations of anonymous data packets during message forwarding [32]. Both shared keys are generated based on the certificateless cryptosystem using the DH key exchange.

A. ANONYMOUS COMMUNICATION LINK ESTABLISHMENT

When a user ID_A wants to establish an anonymous communication link with a destination D, KGC firstly selects L (this protocol takes $L = 3$ as an example) onion routing nodes in the public list, and obtains the long-term public key $pk_i(i = 1, 2, 3)$ for the relay nodes and long-term public key pk_D for the destination node D.

A user ID_A uses his own private key to generate temporary public-private key pairs (X_D, Y_D) and $(X_i, Y_i)(i = 1, 2, 3)$, respectively with the destination node D of the link and the three routing nodes $R_i(i = 1, 2, 3)$ when establishing the link. Based on these temporary public-private key pairs, the user ID_A uses the DH key exchange algorithm to generate shared session keys K_D and $K_i(i = 1, 2, 3)$ with destination node D and three routing nodes $R_i(i = 1, 2, 3)$, respectively.

User ID_A firstly generates a link establishment request REQ, encrypts it using the shared key K_D. And then ID_A uses the shared key $K_i(i = 1, 2, 3)$ encrypt the anonymous link setup request layer by layer to obtain a three-layer onion link setup packet.

The encrypted content includes next-hop router, session deadline, and onion encrypted packets that needs to be forwarded to the next-hop router.

Moreover, ID_A distributes the temporary key Y_D and $Y_i(i = 1, 2, 3)$ selected by itself to the destination node D and each onion routing node $R_i(i = 1, 2, 3)$, using the long-term public key pk_i of the node R_i.

The three-layer onion packet structure is shown as follows.

$$\begin{aligned}&\{\{\{\{(REQ, -, EXP, -)_{K_D}, (Y_D)_{pk_D}, EXP, R_D\}_{K_3},\\&(Y_3)_{pk_3}, EXP, R_3\}_{K_2}, (Y_2)_{pk_2}, EXP, R_2\}_{K_1}, (Y_1)_{pk_1}\}\end{aligned} \tag{2}$$

A user ID_A sends the three-layer onion to the first routing node R_1 of the anonymous link. R_1 decrypts the public key of the onion packet by using its long-term private key sk_i and gets the temporary public key Y_1. The decryption operation of the onion packet is performed in two steps:

(1) Using the DH algorithm, calculate the shared session key K_1 from the sender ID_A based on the temporary public key $Y_1 : K_1 = (Y_1)^{sk_1}$.

(2) Decrypt the first layer of onion using the shared session key K_1, get the next-hop routing node R_2, the session expiration time EXP, and the two-layer onion packet that needs to be forwarded to R_2.

$$\begin{aligned}&\{\{\{(REQ, -, EXP, -)_{K_D}, (Y_D)_{pk_D}, EXP, R_D\}_{K_3},\\&(Y_3)_{pk_3}, EXP, R_3\}_{K_2}, (Y_2)_{pk_2}\}\end{aligned} \tag{3}$$

R_1 first checks the validity of the session time. If the deadline is not exceeded, R_1 forwards the decrypted two-layer onion encrypted data packet to routing node R_2.

After having received the two-layer encrypted onion packet, the routing node R_2 performs the same operation as the routing node R_1. He decrypts the temporary public key Y_2

with its private key, and calculates the shared session key K_2. R_2 uses K_2 to decrypt the second layer of the onion to collet the next hop routing node R_3, the session expiration time EXP, and one-layer of onion encrypted packet that needs to be forwarded to R_3 as follows.

$$\{\{(REQ,-,EXP,-)_{K_D},(Y_D)_{pk_D},EXP,R_D\}_{K_3},\\(Y_3)_{pk_3}\} \tag{4}$$

R_2 forwards the decrypted one-layer onion encrypted data packet to routing node R_3.

Routing node R_3 performs the same operation as routing node R_2, obtains the inner onion encrypted packet as follows. And R_3 forwards it to the destination R_D of the anonymous link.

$$(REQ,-,EXP,-)_{K_D},(Y_D)_{pk_D} \tag{5}$$

The destination D receives the inner onion and decrypts $(Y_D)_{pk_D}$ with its private key sk_D to obtain the temporary public key Y_D. The temporary public key Y_D and its own long-term private key sk_D are used to generate the shared key K_D and D decrypts $(REQ,-,EXP,-)_{K_D}$ to get an anonymous link establishment request REQ and session expiration time. After decryption, it is found that the packet has no next-hop router, and it confirms D that it is the ultimate receiver of the packet, and it can start analysing the link establishment request REQ .

If D agrees to the link establishment request, by interacting with KGC, based on the certificateless cryptosystem, it generates a pair of session temporary public-private key pair $(X_D^{'},Y_D^{'})$. The subsequent session applies the DH key exchange algorithm to generating a session key $K_D^{'}$, where $K_D^{'}=(Y_D)^{X_D^{'}}$ for the subsequent data transmission phase between the source ID_A and the destination D. Then D generates the acknowledgment information ACK of the anonymous link establishment. D uses the K_D to encrypt its own session temporary public key $Y_D^{'}$ and acknowledgement information, and returns it to its previous node R_3 according to the previously reserved routing information.

When the onion routing node $R_i(i=1,2,3)$ receives the confirmation message, it generates a pair of session temporary public-private key pair $(X_i^{'},Y_i^{'})(i=1,2,3)$ in the same way.

The onion routing node $R_i(i=1,2,3)$ can generate the shared session key $K_i^{'}$, where $K_i^{'}=(Y_i)^{X_i^{'}}(i=1,2,3)$ with the temporary public key Y_i of ID_A. In the later session phase, the onion routing node $R_i(i=1,2,3)$ can use this session key $K_i^{'}(i=1,2,3)$ to perform encryption and decryption operations on the onion packet.

D and onion router $R_i(i=1,2,3)$ encrypt acknowledgement message ACK layer-by-layer using K_D, $K_i(i=1,2,3)$ and return the onion packet to the previous-hop node in turn as follows:

$$\{\{\{(ACK,Y_D^{'})_{K_D},Y_3^{'}\}_{K_3},Y_2^{'}\}_{K_2},Y_1^{'}\}_{K_1} \tag{6}$$

After having received the confirmation onion packet, the user ID_A decrypts the onion packet layer by layer using the shared key $K_i(i=1,2,3)$ and K_D to obtain the confirmation message of the destination D and the temporary public key $Y_i^{'}(i=1,2,3)$ and $Y_D^{'}$. ID_A can calculate the session keys $K_i^{'}(i=1,2,3)$ and $K_D^{'}$ of the routing nodes $R_i(i=1,2,3)$ and D in the anonymous data transmission phase based on the DH key exchange algorithm.

$$K_D^{'}=(Y_D^{'})^{X_D} \tag{7}$$

$$K_i^{'}=(Y_i^{'})^{X_i}(i=1,2,3) \tag{8}$$

At this point, the anonymous link establishment process ends.

B. ANONYMOUS MESSAGE FORWARDING

If the link is successfully established, anonymous data transmission can be performed between users. The user ID_A selects the anonymous message m to be sent, and uses the session key $K_i^{'}(i=1,2,3)$ generated during the link establishment phase to sequentially encrypt and generate onion routing packets, and forward messages along with the anonymous communication link.

In order to improve the transmission efficiency, this paper strips the anonymous message from the onion packet and only encrypts the message m with the session key $K_D^{'}$, and designs a new packet format (as shown in TABLE 1). It greatly reduces the number of encryption and decryption operations of anonymous messages, improving the efficiency of anonymous message forwarding.

TABLE 1: Anonymous message packet format

Head of the data packet with onion encryption $\{\{\{(-,EXP,-)_{K_D^{'}},EXP,R_D\}_{K_3^{'}},EXP,R_3\}_{K_2^{'}},EXP,R_2\}_{K_1^{'}}$
Valid data ciphertext $\{m\}_{K_D^{'}}$
Cryptographic signature σ

Therefore, in order to prevent the data packet from being illegally falsified during the message transmission process, this paper uses the special signature method to ensure the integrity of the anonymous message. If the message is changed, the signature verification fails and the message is discarded.

The user ID_A and the three onion routers sequentially encrypt the next hop routing information R and the session expiration time EXP on the anonymous path using the session keys $K_D^{'}$ and $K_i^{'}(i=1,2,3)$, respectively, to form an onion routing packet structure for anonymous message forwarding. The valid anonymous message m uses $K_D^{'}$ to encrypt just one layer, and then performs subsequent signature operation, which is appended to the onion routing packet.

After the user ID_A has obtained the signature key sk, he uses it to sign the packet and send it to the first node of the link.

After having received the data packet, the onion routing node $R_i(i = 1, 2, 3)$ uses the shared session key $K_i^{'}(i = 1, 2, 3)$ to decrypt the corresponding onion layer, obtains the next hop routing information, and forwards the inner layer onion to the next hop. All onion routing nodes undertake the same procedure until the packet reaches destination D. D firstly verify the validity of the signature. If the signature is invalid, the anonymous message is discarded. Otherwise, the anonymous communication message m of the sender is obtained by decrypting the innermost onion packet using the session key $K_D^{'}$.

The packet structure in the anonymous message forwarding process is as follows:

$$\begin{aligned}&\{\{\{\{(-, EXP, -)_{K_D'}, EXP, R_D\}_{K_3'}, EXP, R_3\}_{K_2'},\\&EXP, R_2\}_{K_1'} || \{m\}_{K_D'} || \{hash(m)\}_{Sign}\}\end{aligned} \tag{9}$$

Conversely, if the destination D wants to communicate anonymously with the user ID_A, suppose the anonymous message is $m^{'}$, the onion packet is forwarded in the reverse direction of the anonymous communication path. Destination D first encrypts the anonymous message and the expiration time EXP with $K_D^{'}$, and feeds back to the previous-hop router on the anonymous path. The previous-hop router continues to encrypt the packet with the session key along the anonymous path. Until the anonymous message is fed back to the source user ID_A, ID_A uses the session keys $K_i^{'}(i = 1, 2, 3)$ and $K_D^{'}$ to decrypt the onion in turn to get the anonymous message $m^{'}$ sent by D. The corresponding packet format is as follows.

$$\begin{aligned}&\{\{\{(m^{'}, \{hash(m^{'})\}_{Sign}, EXP)_{K_D'}, EXP\}_{K_3'}, EXP\}_{K_2'}\\&, EXP\}_{K_1'}\end{aligned} \tag{10}$$

The anonymous message forwarding process between User ID_A and Destination D completes.

C. PERFORMANCE ANALYSIS

1) SECURITY ANALYSIS

Theorem 3 If the encryption algorithm is provably secure, the new anonymous communication protocol is secure.

Proof If attacker A can break the security of the protocol in this paper, then we can build an algorithm, in which Challenger C can use $A's$ ability to solve the CDH problem.

Suppose that C is the challenger of certificateless encryption. C runs $Setup(1^l) \to (params, sk_{sys})$ and calculates (pk_G, sk_G), sends $params$ and (pk_G, sk_G) to A. If A issues a key extraction query on the ID^*, then C computes D_{ID^*} and returns D_{ID^*} to A . A sends ID^* , (m_0, m_1) and signing key sk^* to C. C calculates ciphertext c^* on m_0 or m_1, uses sk^* to calculate the corresponding signature σ, and sends (ID^*, σ, c^*) to A. If A issues a key extraction query on $ID = ID^*$, C returns D_{ID} to A. Finally, A outputs $b^{'} \in (0, 1)$, which can break the security of the certificateless encryption protocol. This contradicts to the security of the proved certificateless encryption protocol.

2) EFFICIENCY ANALYSIS

In order to analyze the anonymous communication protocol in this paper, the efficiency of the protocol is discussed in this section. The efficiency achieved by the specific parameters is compared with other current protocols. We analyze the cost of establishing a link of length n from the perspective of the source user and the onion routing nodes.

A comparison of the protocol in this paper with other protocols [27], [33] is shown in the following TABLE 2.

TABLE 2: Efficiency Analysis

index	IBE		CL-OR		This protocol	
	user	routing node	user	routing node	user	routing node
Encryption	1	0	1	0	1	0
Decrypt	0	1	0	1	0	1
Modular exponentiation	1	0	3	2	3	3
Algorithms in Groups	3	1	0	0	0	0
Pair	2	0	0	0	0	0

As can be seen in TABLE 2, the protocol of this paper is basically the same as the other two protocols except for the modulus exponentiation. But in our protocol, the value $s_{ID} \cdot pk_{sys}^{H_2(ID, s_{ID})}$ of each onion routing node can be pre-calculated to spare the computing overhead. In addition, since the key of KGC does not change, the pre-calculated value does not need to be changed repeatedly. The user has to calculate three exponentiations for each onion routing node. Since u_{ID} will frequently change in value $u_{ID}^{\omega_{ID}}$, it cannot be pre-calculated. On the other hand, each onion routing node performs three exponentiations to obtain the session key.

The key point is that the proposed anonymous communication protocol does not require KGC to make frequent changes to the system key. The user only needs to obtain the key of KGC once, which also holds when the onion routing node requests its private key. For KGC, it does not require repeatedly running the key generation phase and has a lower computational load. At the same time, the use of a certificateless mechanism can avoid unnecessary scalability and complex key escrow issues, as well as management verification of user certificates. Compared with other protocols, it not only implements security in the encryption process, but also detects whether the data has been modified. The protocol can still achieve better efficiency.

V. CONCLUSION

To ensure the end-to-end user data confidentiality, integrity and identity privacy in CPSS [34], we respectively propose a certificateless encryption scheme, an anonymous communication packet encapsulation format, and an anonymous communication link establishment method. And at last we complete an anonymous protocol in CPSS. Performance analysis shows that the anonymous communication protocol of this paper has the identity anonymity, the messages confidentiality and messages integrity features. However, the proposed protocol in this paper just uses three relay nodes in the anonymous communication link. How to increase the number of relay nodes in anonymous communication link to strengthen the security of anonymous communication in CPSS is our key points in the future work.

VI. ACKNOWLEDGEMENT

This work was partly funded by EU Horizon 2020 DOMINOES Project (Grant Number: 771066).

REFERENCES

[1] X. Fei, N. Shah, N. Verba, K.-M. Chao, V. Sanchez-Anguix, J. Lewandowski, A. James, and Z. Usman, "Cps data streams analytics based on machine learning for cloud and fog computing: A survey," Future Generation Computer Systems, vol. 90, pp. 435–450, 2019.

[2] F. Longo, A. Puliafito, and O. Rana, "Guest editors' introduction to the special issue on fog, edge, and cloud integration for smart environments," 2019.

[3] Y. Wang, A. Bracciali, T. Li, F. Li, X. Cui, and M. Zhao, "Randomness invalidates criminal smart contracts," Information Sciences, vol. 477, pp. 291–301, 2019.

[4] Y. Wang, M. Zhao, Y. Hu, Y. Gao, and X. Cui, "Secure computation protocols under asymmetric scenarios in enterprise information system," Enterprise Information Systems, pp. 1–21, 2019.

[5] L. Zhang, Y. Wang, F. Li, Y. Hu, and M. H. Au, "A game-theoretic method based on q-learning to invalidate criminal smart contracts," Information Sciences, vol. 498, pp. 144–153, 2019.

[6] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," Future Generation Computer Systems, vol. 88, pp. 636–643, 2018.

[7] L. Qi, Q. He, F. Chen, W. Dou, S. Wan, X. Zhang, and X. Xu, "Finding all you need: web apis recommendation in web of things through keywords search," IEEE Transactions on Computational Social Systems, vol. 6, no. 5, pp. 1063–1072, 2019.

[8] J. Zhou, J. Sun, X. Zhou, T. Wei, M. Chen, S. Hu, and X. S. Hu, "Resource management for improving soft-error and lifetime reliability of real-time mpsocs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 12, pp. 2215–2228, 2018.

[9] C. Cui, F. Li, T. Li, J. Yu, R. Ge, and H. Liu, "Research on direct anonymous attestation mechanism in enterprise information management," Enterprise Information Systems, pp. 1–17, 2019.

[10] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," Wireless Communications and Mobile Computing, vol. 2018, 2018.

[11] Y. Xu, L. Qi, W. Dou, and J. Yu, "Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment," Complexity, 2017.

[12] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data," IEEE Journal on Selected Areas in Communications, vol. 11, p. 35, 2017.

[13] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," IEEE Communications Magazine, pp. 80–85, 2017.

[14] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based big-data-driven routing recommendation approach for heterogeneous networks," IEEE Network, vol. 33, no. 1, pp. 64–69, 2019.

[15] J. Zhou, J. Sun, P. Cong, Z. Liu, X. Zhou, T. Wei, and S. Hu, "Security-critical energy-aware task scheduling for heterogeneous real-time mpsocs in iot," IEEE Transactions on Services Computing, 2019.

[16] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "Anonymous communication protocol in overlay networks," IEEE International Conference on Communications, pp. 1648–1652, 2008.

[17] J. Chen, X. Du, and B. Fang, "An efficient anonymous communication protocol for wireless sensor networks," Wireless Communications Mobile Computing, vol. 14, pp. 1302–1312, 2012.

[18] C. A. Ardagna, M. Conti, M. Leone, and J. Stefa, "An anonymous end-to-end communication protocol for mobile cloud environments," IEEE Transactions on Services Computing, vol. 3, pp. 373–386, 2014.

[19] N. W. Lo, M. C. Chiang, and Y. H. Chao, "Hash-based anonymous secure routing protocol in mobile ad hoc networks," Asia Joint Conference on Information Security, pp. 55–62, 2015.

[20] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, "Hornet:high-speed onion routing at the network layer," ACM Sigsac Conference on Computer and Communications Security, pp. 1441–1454, 2015.

[21] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Journal of the Franklin Institute, vol. 2, pp. 135–139, 2004.

[22] J. Zhou, X. S. Hu, Y. Ma, J. Sun, T. Wei, and S. Hu, "Improving availability of multicore real-time systems suffering both permanent and transient faults," IEEE Transactions on Computers, vol. 68, no. 12, pp. 1785–1801, 2019.

[23] J. Camenisch and A. Lysyanskaya, "A formal treatment of onion routing," International Conference on Advances in Cryptology, pp. 129–187, 2005.

[24] J. Aaron, "A model of onion routing with provable anonymity," International Conference on Financial Cryptography International Conference on Usable Security, pp. 125–130, 2007.

[25] F. Picciallì, G. Casolla, S. Cuomo, F. Giampaolo, and V. S. di Cola, "Decision making in iot environment through unsupervised learning," IEEE Intelligent Systems, 2019.

[26] F. Burgstaller, A. Derler, S. Kern, G. Schanner, and A. Reiter, "Anonymous communication in the browser via onion-routing."

[27] D. Catalano, D. Fiore, and R. Gennaro, "A certificateless approach to onion routing," International Journal of Information Security, vol. 3, pp. 456–489, 2017.

[28] F. Piccialli, S. Cuomo, V. S. di Cola, and G. Casolla, "A machine learning approach for iot cultural data," Journal of Ambient Intelligence and Humanized Computing, pp. 1–12, 2019.

[29] R. A. Haraty and B. Zantout, "The tor data communication system," Journal of Communications Networks, vol. 4, pp. 415–420, 2014.

[30] G. Casolla, S. Cuomo, V. S. Di Cola, and F. Piccialli, "Exploring unsupervised learning techniques for the internet of things," IEEE Transactions on Industrial Informatics, 2019.

[31] X. Wang, L. T. Yang, H. Li, M. Lin, J. Han, and B. O. Apduhan, "Nqa: A nested anti-collision algorithm for rfid systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 18, no. 4, pp. 1–21, 2019.

[32] X. Wang, L. T. Yang, Y. Wang, X. Liu, Q. Zhang, and M. J. Deen, "A distributed tensor-train decomposition method for cyber-physical-social services," ACM Transactions on Cyber-Physical Systems, vol. 3, no. 4, pp. 1–15, 2019.

[33] K. Emura, A. Kanaoka, S. Ohta, and T. Takahashi, "Building secure and anonymous communication channel:formal model and its prototype implementation," Acm Symposium on Applied Computing, pp. 1641–1648, 2014.

[34] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlated graph," EURASIP Journal on Wireless Communications and Networking, pp. 1123–1345, 2019.