An Attribute Credential Based Public Key Scheme for Fog Computing in Digital Manufacturing

Xuanxia Yao, Huafeng Kong, Hong Liu, Tie Qiu, *Senior Member IEEE*, and Huansheng Ning, *Senior Member, IEEE*

***Abstract*—In order to meet low latency, service sensitive and location awareness requirements of digital manufacturing, fog computing is introduced to be an intermediate layer between industrial Internet of Things (IoT) and cloud. The distributed, dynamic characteristics and the collaboration requirement make it face many new security and privacy issues that cannot be solved by the traditional public key or symmetric cryptosystem. For addressing them, a registered but anonymous attribute credential is designed to manage the network entities. Based on it, an attribute credential based public key cryptography (AC-PKC) is constructed to provide flexible key management by taking the advantage of the certificate-less public key cryptography (CL-PKC) and the combination property of the elliptic curve cryptography(ECC). Encryption, authentication and access control with privacy preserving can be realized on the basic operations of AC-PKC, which can meet various security requirements of fog computing based digital manufacturing. The performance analyses and comparison with the existing public key schemes and attribute based encryption(ABE) solutions show that the proposed scheme can work flexibly at a relative low cost.**

***Index Terms*—Access Control, Authentication, Digital Manufacturing, Fog Computing, Key Management**

I. Introduction

DIGITAL manufacturing requires high quality machines connection and collaboration, real time data collection and process, intelligent decision making[1], and location awareness as well, which make the centralized industrial applications and IoT-Cloud based digital manufacturing can't work well[2]. Fog computing appears as an intermediate layer between IoT and cloud to support geographically distributed, location awareness, real-time or low latency and service sensitive industrial IoT applications [2]. Unfortunately, the distributed, collaboration and dynamic characteristics of fog computing make it face many new security and privacy issues. On the one hand, strange nodes or entities always need to interact with each other，which may lead to data or privacy leakage, impersonation and manipulation as well [3][4]. On the other, the distributed and dynamic environment and resource-constraint nodes introduce limitations on security mechanisms. Consequently, the existing public-key infrastructure based cryptography (PKI-PKC), identity based public key cryptography (IB-PKC) and symmetric cryptography can't meet the security and privacy requirements of fog computing based digital manufacturing. To deal with it, we try to take the advantages of the certificate-less public key cryptography(CL-PKC) and the combination property of the elliptic curve cryptography(ECC) [5] to construct an public key scheme to realize flexible key management and meet the dynamic and distributed security and privacy requirements.

The main contributions include four aspects. 1) A registered but anonymous attribute credential and its consummation algorithm are designed; 2) An attribute credential based public key cryptography(AC-PKC) and its basic operation algorithm are defined; 3) The fast one-way accumulator is employed to verify the blinded attribute; 4) Two instances of the security mechanism with privacy preserving are achieved in fog computing based digital manufacturing.

The rest of the paper is organized in 7 sections. Section II gives a description of the related work. Section III describes the preliminaries. Section IV constructs the registered but anonymous attribute credential. Section V designs the attribute credential based public key scheme and its basic operations. two security mechanisms with privacy preserving for fog based digital computing are carried out in section VI,. Section VII makes performance and comparison analysis for the proposed scheme. Section VIII draws a conclusion.

II. Related Work

At this stage, the researches related to security and privacy issues in fog computing mainly concentrate on analyzing the security and privacy challenge or exploring the potential

The manuscript is received on Nov. 30, 2017. This work was supported by the Key Lab of Information Network Security Ministry of Public Security under Grant "C16601", the Chinese Fundamental Research Funds for the Central Universities under Grant "06105031", National Natural Science Foundation of China under Grant "61672131" and "61601129".

X. Yao and H. Ning are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China. (e-mail: yaoxuanxia@163.com, ninghuansheng@ustb.edu.cn).

H. Kong is with the Third Research Institute of Ministry of Public Security,Shanghai,200031, China. (e-mail: robink74@stars.org.cn).

H. Liu is with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, 200062, China.(e-mail: liuhongler@foxmail.com).

T. Qiu is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, 300350, China. (e-mail: qiutie@ieee.org).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2891079, IEEE Transactions on Industrial Informatics

security and privacy vulnerabilities and how to solve a specific security issue or secure a specific fog computing environment.

A. Security and Privacy Analysis

The most authoritative security and privacy analysis should be the argument of OpenFog Consortium, they point out in the reference architecture (RA)[6] that any security compromise in the fog network can result in severe consequences. Based on the RA, M. MUKHERJEE et al. analyze the security and privacy challenges in fog computing [2] and figure out some typical security and privacy issues. For instance, the traditional authentication mechanism faces the challenges from resource constraint devices; the fog nodes' dynamicity and large scale devices make the traditional symmetric and asymmetric cryptosystem can't work well. Meanwhile, they give some open questions and research challenges. A. O de Sà analyze the attacks in CPS[7]. P.G. Lopez et al. summarize the problems relevant to the security and privacy in fog computing [8].

In addition, there are also some analysis on the status of security and privacy issues in fog computing. For instance, A. Alrawais et al. [9] point out that the research on the security and privacy issues of fog computing for IoT is still at the early stage and illustrate the specificity and importance of authentication, access control and privacy preserving in IoT. Y. Wang and et al. point out that the location sensitivity, wireless connectivity, and geographical accessibility as well may result in some new security and forensics challenges for fog computing [10].

B. Security and Privacy Solutions

Most of the security and privacy solutions for fog computing are application-oriented. For instance, H. Hamid et al. [11] construct a security model for privacy preserving of medical big data in fog computing based healthcare. But the heavy overheads of the pairing-based cryptography makes it not suitable well for fog computing. H. Kim et al. [12] conduct a research on authentication and authorization for Internet of Things. They think that the overheads of authentication and authorization should be distributed to fog nodes. Y. W. Law et al.[13] proposed a wide-area measurement system key management model for the smart grid, which can enhance the system security to certain degree. M. H. Ibrahim presents a mutual authentication scheme for fog nodes[14]. In addition, Y Wang and et al. [7] propose an Software Defined Network based security architecture to leverage a centralized controller on the cloud and the distributed controllers in fog network.

III. PRELIMINARIES

A. System Model

In general, a fog computing based digital manufacturing system includes three layers as shown in Fig 1.

The perception and executive layer consists of all kinds of machines equipped with smart sensors and actuators. Their main tasks are to collect data, receive and execute instructions. These smart machines usually do not trust each other before their communication or collaboration.

The fog layer can be seen as a distributed network of fog nodes. Fog nodes may be industrial robot, network devices, smart mobile ends, AGV (Automated Guided Vehicle) and so on. They are usually on move, except for network devices. Although they don't trust with each other, they often need to communicate and collaborate mutually.

The cloud layer includes a variety of servers to provide services. For instance, the data server is responsible for data analysis, processing and decision making; the management server is in charge of managing various affairs, such as staff, equipment, and production management; the security server takes charge of the security related issues in the system.

B. Security and Privacy Requirements

In order to clarify the security and privacy requirements, we make a detailed investigation on the existing research achievements, and 4 aspects are summarized.

1) Since fog nodes, smart sensors and actuators may belong to different departments or stakeholders and have no prior knowledge about each other, a trust relationship should be established before they collaborate so as to identify malicious nodes. The trust is usually established by authentication, while traditional authentication mechanisms can't address the authentication between strange entities.

2) Smart actuators must ensure that the received instructions do come from the legitimate nodes with required attributes and keep fresh and integrity. And the existing authentication mechanism can't meet these security requirements well.

3) IoT devices need to reject the unauthorized access to its data for avoiding data leakage, and Fog nodes need to refuse the unauthorized access to its service for saving resources. Both situations require distributed access control mechanism rather than the existing centralized access control mechanism.

4) Fog nodes are close to IoT devices and can collect sensitive data, e.g. user's identity, location and etc.. Since these data are directly associated with the users in the local context, the existing privacy preserving mechanisms cannot work well.

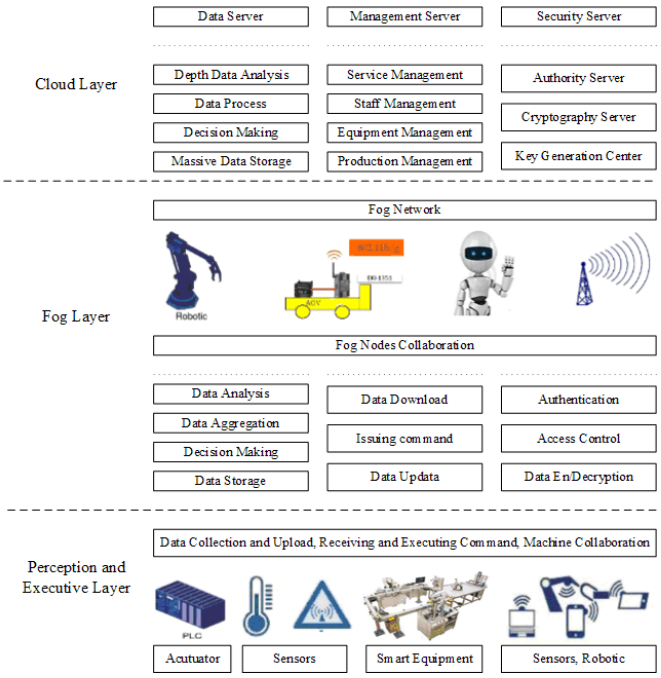

Fig. 1. A fog computing based digital manufacturing system

C. Assumptions

For reading and understanding, some assumptions and explanations are made as following.

1) The attribute servers are assumed to be trusted, manage attributes, construct and issue the registered but anonymous attribute credentials. Key generation center(KGC) is the unique security server, which is completely trusted and in charge of generating the system security parameters and partial key pair.

2) Each entity has a unique identifier assigned by one of its registered attribute servers. Server's unique identifier is public, and the unique identifier of a non-service node is only known to itself and the registered server. In addition, every entity has a secret shared key with the registered attribute server.

3) An elliptic curve E defined over the finite field F_p with a subgroup of prime order q is used to be the system's basic algorithm. The default encryption algorithm is Elliptic Curve Integrated Encryption Standard (ECIES)[15]. And the default signature algorithm is Elliptic Curve Digital Signature Algorithm (ECDSA)[16] .

4) The private key and public key's material of each server are generated by the algorithm in literature [17] in advance. Any an entity can derive the public key of a server according to its identifier and public key's material.

D. Notations

For description, the main notations used in the scheme are illustrated in Table I.

IV. Attribute Credential

Similar to the real life, an credential in digital world is also used to certify that the owner has some attributes or privileges. In the existing credential, the attributes or privileges are all in plain text, which may lead to unnecessary privacy or data leakage. To deal with it, we try to construct a registered but anonymous attribute credential to solve the security and privacy issues in fog computing based digital manufacturing.

For practicability, simplicity and privacy preserving, the attribute credential to be designed is only to manifest that its owner has some attributes or meets some attributes requirements, and does not reveal any plaintext information about the owner and its attributes. In addition, considering that different attributes may be administrated by different attribute servers, one entity's multiple attribute credentials issued whether by the same attribute server or different ones should be used together to meet complex attributes requirements.

TABLE I
Main Notations in the Scheme

Symbol	Meaning
n	A security parameter to indicate the size in bits of a key
$E(k,d)$	Using the shared secret key or public key k to encrypt data d.
$D(k,e)$	Using the shared secret key or private key k to decrypt e
$h()$	A cryptographic hash function.
$H()$	The fast one way accumulative function in literature [18]
$\mathrm{Sig}(k,m)$	Using the private key k to make signature on m.
u_i	The unique identifier of entity i.
$a_{i,j}$	The j^{th} attribute of entity i.
$C_{i,j}$	The j^{th} attribute credential of entity i.
$K_{i,j}$	The shared key between entity i and j.
P_{pub}	The master public key of the system.
s	The master private key of the system.
F_p	A finite field with p elements $\{0, p-1\}$, and p is a large prime.
q	a large prime.
G	The base point on the elliptic curve E that generates the subgroup of order q.
$\times$	The scalar multiplication in ECC.

A. Attribute Credential Construction

The registered but anonymous attribute credential consists of 7 items. For description, it is assumed that there are j attributes of entity u_i to be issued in one attribute credential by attribute server A. The construction process is illustrated by algorithm1.

Algorithm 1 Attribute Credential Construction

Begin

Step 1. For all the attributes to be issued for user u_i, the attribute server A computes $I_1=h(h(K_{A,i}||u_i||a_{i,1}||a_{i,2}||\ldots||a_{i,j}))$.

Step 2. Computes $I_2=h\,(I_1||h(u_i))$.

Step 3. For all attributes to be issued in the credential, A encrypts it with the shared key between it and u_i, which is denoted by $I_3=\mathrm{E}(K_{A,i}, a_{i,1}||a_{i,2}||\ldots||a_{i,j})$.

Step 4. Sets I_4 to be the validity period of the credential.

Step 5. Let I_5 to be the attribute server's unique identifier ID_A.

Step 6. Let I_6 to be the public key material of ID_A.

Step 7. For each attribute $a_{i,j}$, computes its hash value, and calculates the one-way accumulated value for them according to the fast one-way accumulator in [18], which is denoted by $I_3^*=H(H(\ldots(H(K_{A,i},h(a_{i,1})),h(a_{i,2})),\ldots,h(a_{i,j-1})),h(a_{i,j}))$.

Step 8. Makes signature on the hash value of I_1, I_2, I_3^*, I_4, I_5 and I_6, which is denoted by $I_7=\mathrm{Sig}(s_A, \mathrm{H}(I_1||I_2||I_3^*||I_4||I_5||I_6))$.

End.

It should be stated that item I_1 is used to identify the attribute credential and help the trusted third part to verify whether the credential is used by its owner before it has a corresponding key pair; item I_2 is designed to help the owner consummate the credential and the verifier identify the collusion attack; item I_3 is designed for the credential owner to get the attribute value assigned by the attribute server. And I_3 will be replaced with I_3^* or the one way accumulated value of the attributes hash when the owner consummates the credential.

It can be seen that both the identity and the attributes of the attribute credential owner are blinded by encryption, hash and hash based one-way accumulated value. The aim is to make the attribute credential be a registered but anonymous one, realize privacy preserving and avoid collusion attacks.

B. Attribute Credential Consummation

When the entity u_i receives an attribute credential from the attribute server A, it needs to verify and consummate the attribute credential. The process is described in algorithm 2.

After consummating the attribute credential, entity u_i can use it to apply for its partial key pair to KGC and generate a key pair for it according to the proposed AC-PKC.

V. Attribute Credential based Public Key Scheme

The attribute credential based public key system is designed to solve the issues of key management and privacy preserving in fog computing based digital manufacturing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2891079, IEEE Transactions on Industrial Informatics

Algorithm 2 Attribute Credential Consummation

Begin

Step 1. Compute h ($I_1||h(u_i)$) and check it is equal to I_2 or not. If yes, it means that the credential is for u_i, else, discard it and go to step 9.

Step 2. Check whether the credential is overdue or not according to I_4, if yes, discard it and go to step 9.

Step 3. Decrypt I_3 with the shared key $K_{A,i}$ between A and u_i to get the attributes $\tilde{a}_{i,j}$ in the credential.

Step 4. Compute $h(h(K_{A,i}||u_i||\tilde{a}_{i,1}||\tilde{a}_{i,2}||\ldots||\tilde{a}_{i,j}))$ and check whether it equals to I_1 or not. If yes, it indicates that the attributes are integrity, else, discard it and go to step 9.

Step 5. For all the attributes, compute their hashes' one-way accumulated value according to the fast one-way accumulator in literature [18], which is denoted by $I_3^*=H(H(\ldots(H(K_{A,i}, h(\tilde{a}_{i,1})), h(\tilde{a}_{i,2})),\ldots, h(\tilde{a}_{i,m-1})), h(\tilde{a}_{i,m}))$.

Step 6. Derive the public key P_A of attribute server A according to I_5 and I_6.

Step 7. Verify the signature I_7 with P_A according to the I_1, I_2, I_3^*, I_4, I_5 and I_6. If it can't pass the verification, discard the credential and go to step 9.

Step 8. Replace I_3 with I_3^*.

Step 9. Stop.

End.

A. Conception and Principle

The attribute credential based public key scheme is based on CL-PKC[17] so as to simplify key management and avoid key escrow. Since item I_1 binds the attributes and the owner securely and can uniquely identify the attribute credential, the partial key and the corresponding key pair are derived from it rather than the identity in CL-PKC, which can prevent privacy leakage. Considering the characteristics of distributed, resources constraint, and low latency in the industrial field, the ECC based lightweight CL-PKC scheme [17] is employed in AC-PKC. At the same time, the idea of Combinatorial Public Key (CPK) [5] system is also introduced in AC-PKC to realize flexible key management.

It should be stated, although the AC-PKC is closely related to attributes, it is completely different from the well-known ABE (Attribute Based Encryption). It is essentially an ECC based CL-PKC. And ABE is actually an extension of IB-PKC, which is usually based on bilinear mapping and secret sharing.

B. Key Generation

Similar to the CL-PKC, there are 3 phrases to generate a key pair for one attribute credential.

1) System Setup

The system setup phase should be done by KGC, including five steps.

Step 1, Choose an elliptic curve with the parameter of (p, a, b, G, q) according to the system security parameter n.

Step 2, Select $\boldsymbol{s}$ from the interval [1,q-1] randomly as the system's master key, which should be kept secret by the KGC.

Step 3, Compute the system's master public key $\boldsymbol{P}_{\mathbf{pub}}=\boldsymbol{s}\times G$.

Step 4, Chooses a cryptographic hash function h to map the bit string with any finite length to $\{0,1\}^n$.

Step 5, Publish the system's parameter $\{E_p(a, b,), h, \boldsymbol{P}_{\mathbf{pub}}\}$.

2) Partial Key Extraction

A partial key pair is corresponding to an attribute credential. When an entity u_i applies a partial key pair to KGC for its attributes credential $C_{i,j}$, KGC must verify $C_{i,j}$ before extracting a partial key pair for it. The verification includes two phases. The first phase is to verify the validity, authenticity and integrity of the attribute credential, which is similar to the traditional credential verification. The second phase is to check whether the credential is used by its owner so as to avoid the credential embezzlement and collusion attack, which should begin after the first phase succeeds. Since there is still no public and private keys for the attribute credential, the second phase can't be accomplished by public key cryptography. To address it, u_i has to provide the related information to help KGC verify $C_{i,j}$. There are 3 steps should be done by user u_i.

Step 1, Select a random k_s from [1..p-1] as the session key to encrypt the hash value for the concatenation of $K_{A,i}$, u_i, its attributes in t $C_{i,j}$ and the timestamp, which is denoted by $E(k_s, h(K_{A,i}||u_i ||a_{i,1}||a_{i,2}||\ldots||a_{i,j})||T)$, where T is a timestamp.

Step 2, Encrypt k_s with the KGC's public key P_K, which is denoted by $E(P_K, k_s)$.

Step 3, Let M="$C_{i,j}||E(k_s,h(K_{A,i}||u_i||a_{i,1}||a_{i,2}||\ldots||a_{i,j})||T)||E(P_K, k_s)$ and send M to KGC.

After receiving M from u_i, KGC makes the first phase verification for $C_{i,j}$. If it succeeds, KGC can begin the second phase verification, which is described by Algorithm 3.

Algorithm 3 The Second Phase Verification

Begin

Step 1. Decrypt $E(P_K, k_s)$ with its private key S_K to get k_s.

Step 2. Decrypt $E(k_s, h(K_{A,i}||u_i||a_{i,1}||a_{i,2}||\ldots||a_{i,j})||T)$ with k_s to get $h(K_{A,i}||u_i||a_{i,1}||a_{i,2}||\ldots||a_{i,j})||T$.

Step 3. If T is within the reasonable time window, compute $I_1^*=h(h(K_{A,i}||u_i ||a_{i,1}||a_{i,2}||\ldots||a_{i,j}))$.

Step 4. Check whether I_1^* is identical to $C_{i,j}.I_1$ or not, if yes, it indicates that u_i is the credential owner and return true, otherwise, return false.

End.

It should be stated that $h(K_{A,i}||u_i||a_{i,1}||a_{i,2}||\ldots||a_{i,j})$ can be known only to u_i, A and KGC, both the attribute server A and the security server KGC are trusted, so it can be used to check whether the attribute credential holder is its owner or not.

If $C_{i,j}$ passes the second phase verification, KGC will extract a partial key pair for it according to algorithm 4, otherwise, the partial key pair extracting request will be denied.

Algorithm 4 Partial Key Extraction

Begin

Step 1. Choose $r_{i,j}$ randomly from the interval [1, q-1].

Step 2. Compute partial private key $d_{i,j}=(s+r_{i,j}\times C_{i,j}.I_1) \bmod q$.

Step 3. If $d_{i,j}=0$, go back to step 1.

Step 4. Compute partial public key $R_{i,j} = r_{i,j}\times G$.

Step 5. Encrypt $d_{i,j}$ and $R_{i,j}$ with the session key k_s and send $E(k_s, d_{i,j}||R_{i,j}.x||R_{i,j}.y)$ to the applier u_i.

End.

In algorithm 4, $d_{i,j}$ and $R_{i,j}$ constitute the partial key pair of $C_{i,j}$; It is obvious that different attribute credentials have different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2891079, IEEE Transactions on Industrial Informatics

partial key pairs even their attributes values are identical, because the $C_{i,j}.I_1$ is not only a unique identifier of $C_{i,j}$ but also a binding code of u_i and its attributes in $C_{i,j}$.

3) Key Generation

When u_i receives a partial key pair of $C_{i,j}$ from KGC, it can derive the corresponding private key $\boldsymbol{s_{i,j}}$ and the intermediate public key $\boldsymbol{X_{i,j}}$ from it. The process is described by algorithm 5.

Algorithm 5 Key pair generation

Begin

Step 1. Compute and check whether the equation $\boldsymbol{d_{i,j}}\times\boldsymbol{G}=\boldsymbol{P_{pub}}+C_{i,j}.I_1\times\boldsymbol{R_{i,j}}$ can hold or not. If it holds, it indicates that the partial key pair ($d_{i,j}$, $R_{i,j}$) is correct and generated from $C_{i,j}$ by the KGC, otherwise, go to step 7.

Step 2. Choose an integer $z_{i,j}$ randomly from interval [1, q-1].

Step 3. Compute the private key of $C_{i,j}$ by $\boldsymbol{s_{i,j}}=(\boldsymbol{d_{i,j}}+ z_{i,j}\times C_{i,j}.I_1)$ mod q.

Step 4. If $\boldsymbol{s_{i,j}}=0$, go to step 2, otherwise keep $\boldsymbol{s_{i,j}}$ secretly.

Step 5. Compute the intermediate public key $\boldsymbol{X_{i,j}}$ of $C_{i,j}$ by $\boldsymbol{X_{i,j}}=\boldsymbol{R_{i,j}}+z_{i,j}\times\boldsymbol{G}$.

Step 6. If $\boldsymbol{X_{i,j}}=\boldsymbol{O}$, go back to step 2, otherwise, store the executive public key $\boldsymbol{X_{i,j}}$ along with $C_{i,j}$.

Step 7. Stop.

End.

It should be noted although $\boldsymbol{X_{i,j}}$ is derived from the random $r_{i,j}$ in algorithm 4 and $z_{i,j}$ in algorithm 5, it is not really independent of the entity's unique identifier u_i and its attributes in $C_{i,j}$. For one thing, $C_{i,j}.I_1$ is the binding code of u_i and its attributes in $C_{i,j}$. For the other, $C_{i,j}.I_1$ is used together with $r_{i,j}$ and $z_{i,j}$ to generate the partial private key $\boldsymbol{d_{i,j}}$ and the private key $\boldsymbol{s_{i,j}}$ for $C_{i,j}$, which makes the $\boldsymbol{X_{i,j}}$ associate with u_i and its attributes in $C_{i,j}$ indirectly. The relation between $C_{i,j}$'s actual public key $\boldsymbol{P_{i,j}}$ and its intermediate public key $\boldsymbol{X_{i,j}}$ can be described by $\boldsymbol{P_{i,j}}=\boldsymbol{P_{pub}}+C_{i,j}.I_1\times\boldsymbol{X_{i,j}}$, which can be further deduced to be $\boldsymbol{P_{i,j}}=\boldsymbol{s_{i,j}}\times G$. It indicates that $\boldsymbol{X_{i,j}}$ is the bridge between the $C_{i,j}$ and its actual public key $\boldsymbol{P_{i,j}}$. The binding relationship between $\boldsymbol{P_{i,j}}$ and $C_{i,j}$ makes $\boldsymbol{P_{i,j}}$ can certify itself.

C. Basic Operations of AC-PKC

In practice, a consummated attribute credential should be used with its intermediate public key so as to simplify the public key management. Unlike the existing PKI-PKC,IB-PKC and CL-PKC, AC-PKC has to process the attribute credential verification, attribute verification and key combination 3 basic operations to get a key pair.

1) Attribute Credential Verification

Before deriving a key pair for an attribute credential, it must be verified so as to ensure it is a legal one and being used by its owner. There are 3 levels for attribute credential verification. The first and second level verification are prerequisite and the third is required only when multiple attribute credentials are used together. The first is legitimacy verification, which is similar to the existing certificate validation and can be made by the attribute server's public key. The second level is to verify whether the current user is the credential owner or not, which can be done by a time or random dependent signature because the attribute credential's key pair is available now.

The success of the first level verification is the premise of the second level verification, so they can be described together. For clarity, we take a scenario to illustrate the first and second level verification. When the entity u_i tries to prove to anyone v that it has some attributes in the attribute credential $C_{i,j}$, it uses the ECDSA to make a signature on the hash of $C_{i,j}.I_7$ and the current timestamp T (or a random R) by the private key $\boldsymbol{s_{i,j}}$ of $C_{i,j}$, which is denoted by $S^T_{i,j}=\mathrm{Sig}(\boldsymbol{s_{i,j}}, h(C_{i,j}.I_7||T))$. And then sends message m="$C_{i,j}||X_{i,j}||S^T_{i,j}||T$" to v. The verifier v verifies $C_{i,j}$ according to algorithm 6 after receiving message m.

Algorithm 6 The First and Second Level Verification

Begin

Step 1. Check whether the $C_{i,j}$ is within its validity period and T is in a reasonable time window or not. If not, return false.

Step 2. Derive the attribute server's public P_A according to $C_{i,j}.I_5$ and $C_{i,j}.I_6$.

Step 3. Verify $C_{i,j}.I_7$ with P_A. If it fails, return false.

Step 4. Compute $C_{i,j}$'s public key $\boldsymbol{P_{i,j}}=(\boldsymbol{P_{pub}}+ C_{i,j}.I_1\times X_{i,j})$.

Step 5. Verify the signature $S^T_{i,j}$ with $\boldsymbol{P_{i,j}}$ and T according to the verification process in ECDSA. If it succeeds, return true, else return false.

End.

Obviously, if algorithm 6 returns true, it indicates that $C_{i,j}$ is legal and being used by its owner.

The third level verification is also called correlation verification, which is designed to verify whether the attribute credentials used together belong to the same owner or not. It can be realized by an associated code $h(u_i)$ provided by the credential holder. For each received attribute credential $C_{i,j}$, v checks whether the equation $h(C_{i,j}.I_1||h(u_i))=C_{i,j}.I_2$ holds or not. If it holds for all the attribute credentials, they pass the third level verification, else, the third level verification fails.

2) Attribute Verification

In general, an entity's attributes may be certified by different attribute credentials. When an entity u_i wants to prove that it has some attributes or a verifier v wants to verify whether u_i has some attributes, u_i should show the attributes related credentials to v. Unfortunately, it is difficult for v to verify whether u_i has the required attributes only according to its attribute credentials, because all attributes in a credential are blinded for privacy preserving. To address it, an attribute verification algorithm is designed on item I_3 (the one way accumulated value for the hash of each attribute) of the attribute credential.

For description, it is assumed that the attributes to be verified are denoted by set A_a; the number of the attribute credentials provided by the entity u_i is N; the verification result is denoted by V. The attribute verification is illustrated by algorithm 7.

Algorithm 7 Attribute Verification

Begin

Step 1. Let j and k be 1, and V be True.

Step 2. While V and $j\leqslant|A_a|$ do

Step 2.1 For attribute $a_{i,j}$ required to be verified, compute $h(a_{i,j})$.

Step 2.2 Check whether $h(C_{i,k}.I_3,h(a_{i,j}))$ is equal to $C_{i,k}.I_3$ or not. If yes, let j be (j+1) and k be 1, else let k be (k+1).

Step 2.3 If ($k>N$), let V be false.

Step 3. Return V.

End.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2891079, IEEE Transactions on Industrial Informatics

3) Key Combination

A key pair is corresponding to an attribute credential. Since one attribute credential usually only includes partial attributes of an entity, more than one attribute credentials are often required in many cases to meet the attribute requirements. Accordingly, the pair key should be derived from all the related attribute credentials. A simple and practical solution is to take the advantage of the combination property of ECC to derive a combination key pair for the required attributes.

For illustration, we take the case of two attribute credentials as example, which can be used in the occasion of multiple attribute credentials. It is assumed that the entity u_i has two attribute credentials $C_{i,j}$, $C_{i,k}$ issued by attribute server A_1 and A_2 respectively, and their intermediate public keys are $\boldsymbol{X_{i,j}}$ and $\boldsymbol{X_{i,k}}$. The private key $\boldsymbol{s_{i,c}}$ corresponding to $C_{i,j}$ and $C_{i,k}$ can be derived by u_i, which is $\boldsymbol{s_{i,c}}=(\boldsymbol{s_{i,j}}+\boldsymbol{s_{i,k}})$ mod q. The public key $\boldsymbol{P_{i,c}}$ corresponding to $C_{i,j}$ and $C_{i,k}$ can be derived by anyone w that is able to get and verify the two attribute credentials. Firstly, w computes the two attribute credential's public key $\boldsymbol{P_{i,j}} =\boldsymbol{P_{pub}}+C_{i,j}.I_1\times\boldsymbol{X_{i,j}}$ and $\boldsymbol{P_{i,k}} =\boldsymbol{P_{pub}}+C_{i,k}.I_1\times\boldsymbol{X_{i,k}}$ respectively. And then, w can compute $\boldsymbol{P_{i,c}}= \boldsymbol{P_{i,j}} +\boldsymbol{P_{i,k}}$.

Since only scalar multiplications, hash and arithmetic operations are involved in the three basic operations, AC-PKC is suitable for resource constraint scenarios.

VI. Using AC-PKC in Fog Computing based Digital Manufacturing

In fog computing based digital manufacturing, there are 4 types of entities. They are servers, fog nodes, smart sensors and smart actuators. As the description in the system model, the servers can be further classified into data servers, management servers and security servers. For clarity, we make an agreement that management servers refer to attribute servers and KGC is the only security server. Using AC-PKC, the 4 security issues in sub-section B of section III can be addressed from the aspects of authentication, encryption and access control.

For description, the attribute server, the fog node, the smart sensor, the smart actuator, the data and the command are denoted by A, f, δ, α, d and c respectively.

A. Authentication

The authentication can be divided into fuzzy and accurate authentication. The fuzzy authentication just depends on the attribute credential, while the accurate authentication relies on the signature generated by AC-PKC.

1) Fuzzy Authentication

The fuzzy authentication is used to certify that an entity has some required attributes. The entity just shows the related attribute credentials to the verifier, and the verifier can verify whether the credential holder has the expected attributes or not. To be more precise, the fuzzy authentication is achieved by the attribute credential verification and the attribute verification, which are described in subsection *C* of section V. Based on the required attributes, two strange entities can establish the initial or basic trust relationship.

2) Accurate Authentication

The accurate authentication is used to certify that a signature is generated by the private key of the attribute credentials, which include the required attributes and belong to the same owner. Since the signature and the verification key are derived according to the required attributes, it has good flexibility.

Like the fuzzy authentication, the accurate authentication also needs attribute and credential verification so as to avoid the embezzlement and collusion attack. For clarity, we take an actuator α authenticating command c from fog node f for an example to illustrate how to realize accurate authentication.

The fog node f is required to generate a signature for the command c before issuing it to the smart actuator α. The signature should be made according to the required attributes for issuing command c. It is assumed that the required attributes are in k attribute credentials and denoted by $C_{f,1}$, $C_{f,2}$,.., $C_{f,k}$, f needs to compute the signature private key $\boldsymbol{s_f} = (\sum_{j=1}^{k} \boldsymbol{s_{f,j}}) mod\ q$, use $\boldsymbol{s_f}$ to make the signature on c and send "$hash(f)||c||\mathrm{Sig}(\boldsymbol{s_f},c)||C_{f,1}||\boldsymbol{X_{f,1}}||C_{f,2}||\boldsymbol{X_{f,2}}...||C_{f,k}||\boldsymbol{X_{f,k}}$" to α.

When α receives the message, it verifies each attribute credential according to the part 1 of the subsection *C* in section V. If all the attribute credentials pass the verification, α makes attribute verification for the required attributes according to the part 2 of the subsection *C* of section V. If all the required attributes pass the verification, α firstly computes the public key $\boldsymbol{P_{f,j}} =\boldsymbol{P_{pub}}+C_{f,j}.I_1\times\boldsymbol{X_{f,j}}$ for each attribute credential $C_{f,j}$. And then, it the verification public key $\boldsymbol{P_f} = \sum_{j=1}^{k} \boldsymbol{P_{f,j}}$ and verifies the signature $\mathrm{Sig}(\boldsymbol{s_f},c)$ with $\boldsymbol{P_f}$. If $\mathrm{Sig}(\boldsymbol{s_f},c)$ passes the verification, it indicates that the command c does come from f, f has the required attributes and c is intact and fresh.

It should be stated that the privacy can also be preserved in the accurate authentication, because credential owner's identity and all the attributes in the credentials are blinded.

B. Encryption and Access Control

The flexible key management of the proposed AC-PKC makes it easy to derive a public key according to the need of security, which makes the asymmetric or symmetric based encryption can be realized as required and is helpful to solve the issue of sensitive data leakage.

Access control is usually based on authentication and the access control policy can always be described by attributes. So the AC-PKC is very suitable for access control in various environments. Furthermore, the fuzzy authentication is fully capable of undertaking the task. It is illustrated as following.

In the fog based digital manufacturing, when a smart sensor δ subscribe data processing service to a fog node f or f requires data from a smart sensor δ, access control is indispensable, which can be accomplished just by showing attribute credentials to δ or f. It is essentially the fuzzy authentication. According to the fuzzy authentication results, f can decide whether to provide the corresponding data processing service or not; and δ can decide whether to send data and send what data to f. Obviously, the privacy of δ and f can be preserved in the process of access control. In addition, δ should encrypt the data with a key protected by the public key of the attribute credentials what f shows to it so as to avoid data leakage.

C. Experiment

In order to validate the correctness and the usability of the proposed AC-PKC, we construct a plain simulation environment for fog computing based digital manufacturing, and design 2 scenarios to do encryption, authentication and access control experiments. For simplicity, there are only one attribute server, one KGC, one fog node, one smart actuator and one smart sensor deployed in the simulation environment. The attribute server and KGC are realized in Eclipse 4.5.2 and JDK 1.7 on a computer with Windows 10, i5-6300HQ processor and 8G DDR4 memory. The fog node, actuator and smart sensor are realized in Android Studio 1.4 and JDK 1.7 on the Android virtual machine. The size of the security parameter n is 160 bits. The secp160r1 in [19] is adopted as ECC's parameter. Blowfish is employed as the symmetric cryptographic algorithm in ECIES. And SHA1 is used as the hash function.

The first scenario is the simplification of the example in accurate authentication. Only 2 attribute credentials from the same attribute server are used by actuator α to authenticate fog node f. The purpose is to validate the correctness and efficiency of AC-PKC. For objectivity, the average values of 1000 experiments are used to be the experiment results. The experiment results show that the key pair can be generated from the attribute credential correctly and the accurate authentication can be accomplished. The average time for key generation, signature and verification are listed in Table II.

TABLE II
THE AVERAGE TIME FOR THE MAIN OPERATIONS IN ACCURATE AUTHENTICATION

Operation	Average Time
Generating a partial key pair for one attribute credential	7ms
Generating a key pair for one attribute credential	22ms
Signature	7ms
Signature verification	15ms

The second scenario is that a fog node wants to gather data from a smart sensor, which is designed to conduct encryption and access control experiment. The fuzzy authentication is realized to make the access control. And the results show that access control can be implemented correctly. The experiment also indicates that the encryption and decryption based on the AC-PKC are correct. And the average time for asymmetric and symmetric cryptographies are listed in Table III.

It can be seen from Table II and Table III that the average time for the main operations is roughly consistent with their complexity and can basically meet the low latency requirements of the digital manufacturing.

TABLE III
THE AVERAGE TIME FOR THE MAIN OPERATIONS IN SCENARIO 2

Operation	Average Time
Encryption with the attribute credential's public key	24ms
Decryption with the attribute credential's private key	8ms4
Encryption based on Blowfish	123μs
Decryption based on Blowfish	102μs

VII. EVALUATION

To evaluate the proposed AC-PKC objectively, performance analysis is done from the aspects of security, privacy preserving, computation overhead, communication overhead and flexibility. Meanwhile, comparison analysis is also made between AC-PKC and the existing public key schemes.

A. Security

The security of AC-PKC depends on the security of the attribute credential. The attacks that may be launched on attribute credentials and the related security mechanism mainly include credential forgery, collusion attack and credential embezzlement as well.

For credential forgery, it is basically impossible. Because the attribute credential is always issued by the authority attribute servers, and anyone can verify its authenticity and integrity according to the attribute server's public key.

For collusion attack, the basic idea is based on the combinable characteristic of keys. The premise that the attacks can succeed is that the service or data provider will combine the public keys of the attribute credentials from different owners to encrypt the secret information, or verify the attacker's signature. In fact, the service or data providers must verify the attribute credentials in three levels before deriving and using a public key from them. If the attribute credentials don't belong to the same owner, the verification will fail, and the provider will not derive the public key at all. So it is impossible for attackers to launch a collusion attack.

For credential embezzlement, there are two cases. One is that an attribute credential is used directly by an attack to apply the partial key pair to KGC, which can be avoided by item I2 in the attribute credential, because only the credential owner can help KGC verify item I2 and the KGC is trusted. The other is that an attack may replay an attribute credential, which can be defeated by the time dependent signature.

B. Privacy Preserving

Privacy preserving is achieved not only by blinding and hiding the attribute credential owner's identity but also by keeping both the attributes and their values secretly.

Firstly, the attribute credential is registered but anonymous. Since there is no plaintexts of the owner or the attributes in it, there is no chance to leak the identity or attributes' information from an attribute credential.

Secondly, although item I_2 in the attribute credential may be utilized by an attacker to associate different credentials with one owner, the identity of the credentials' owner can't be recognized just according to its hash values. And even the attacker can trace the credential owner's behaviors, it can't conclude who is the owner because too many users have the same behaviors.

Thirdly, all the attributes data are encrypted, hashed or accumulated. Except for the owner and the attribute server, no one knows what attributes are in an attribute credential. Even the verifier can only checks whether a required attribute is in it or not, and not know how many and what attributes in it.

C. Computation Overhead

For simplicity, the computation overhead is measured only by the most time-consuming operation, and the operation like hash, the encryption/decryption of the symmetric cryptographic algorithm, and the arithmetic operations are ignored.

In AC-PKC, the most time-consuming operation is the point addition or scalar multiplication of ECC. In most existing CL-PKCs, the bilinear mapping is the most expensive operation and the modular exponential operation is the second expensive one. So the scalar multiplication, bilinear mapping and the modular exponential are used to measure computation overheads. According to the results corroborated in literature [20], the overhead of one bilinear pairing is about that of 20 scalar multiplications, and one modular exponential operation is about 2 scalar multiplications. Based on this, the metric for computation overhead is the number of the scalar multiplication. And the computation overheads of AC-PKC and the typical CL-PKCs are shown in Table IV.

TABLE IV
COMPUTATION OVERHEADS

	AC-PKC	EC-PKC	CL-PKC	A-CLE	NP-CLS	DL-CLS	BP-CLS
System Master Keys	1	1	1	1	1	2	1
Partial Key pair	1	1	1	1	1	4	1
Key pair	5	**2**	43	3	4/**1**	16	25
Encryption	4	4	61	22	N	N	N
Decryption	1	1	20/21	22	N	N	N
Signature	1	1	23	N	1	2	1
Verification	5	5	80	N	6/7	10	22

It should be stated that the overheads of EC-PKC (ECC based Certificate-less Public Key Cryptography), A-CLE (Authenticated Certificate-less Encryption), DL-CLS (DL based Certificate-less Signature) and BP-CLS (Bilinear Pairing based Certificateless Signature) are from literature[17],[22],[25] and [26] respectively. And the data of CL-PKC[21] are the computation overheads of the basic CL-PKE(Certificate-less Encryption), Full CL-PKE and CL-PKS(Certificate-less Signature). The column of NP-CLS(No Pairing Certificate-less Signature) shows the computation overheads of the schemes in literature [23] and [24] respectively.

Although EC-PKC, A-CLE and NP-CLS have the relative lower computation overheads in key pair generation than our AC-PKC, EC-PKC and NP-CLS don't verify the correctness and the authenticity of the partial key pair, which needs 3 scalar multiplications. Therefore, EC-PKC has no advantage and NP-CLS has only one scalar multiplication advantage over AC-PKC. Nevertheless, the signature verification overhead in NP-CLS overwhelms their advantage. In addition, A-CLE's two scalar multiplication advantage in key pair generation is overtaken by its heavy computation overhead in encryption and decryption. In summary, our AC-PKC scheme has relative low computation overhead.

D. Communication Overhead

The communication overhead is often denoted by the size of the message need to be transmitted. In fog based digital manufacturing, the security related communication is mainly in the process of authentication and access control, including the system public key, required attributes, the ciphertext and signature as well.

In order to estimate the communication overhead objectively, typical ABEs are used as comparison objects, because they can also achieve authentication and access control with privacy preserving. In ABEs, both authentication and access control are accomplished by decrypting the ciphertext. And the public parameters and the required attributes are indispensable in encryption and decryption. So the communication overheads in ABEs can be estimated by the size of the public parameters, the required attributes and the ciphertext related information.

Currently, the basic operations of ABEs are either scalar multiplication or modular exponentiation. The corresponding schemes are denoted by ABE-SM and ABE-ME respectively. For comparison, it is assumed that the length of ABE-SM's security parameter is l (l =160bits). And the length of ABE-ME's security parameter should be 6.4l, because the security strength of the 1024 bits modular exponentiation is equivalent to that of ECC-160. Meanwhile, we also assumed that the number of the required attributes is k, the size of the attributes space is n, the number of the involved attribute credentials is L and the length of each attribute is l.

For simplicity, the algorithm parameters and the length of the message are not taken into account. In fact, the algorithm parameters of all ABE are greater than those of AC-PKC. The size of the required attributes in all the ABE schemes can be expressed by kl. But the attributes in AC-PKC are included in attribute credentials, which makes the size of the required attributes depend on the number of the related attribute credentials and the size of an attribute credential. In a consummated attribute credential, the lengths of the I_1,I_2,I_3,I_5 and I_7 are all l. If each part of the validity period is represented in binary, the maximum length of I_4 is 160 bits, which is identical to l. I_6 is the intermediate public key of the attribute server, as a point on the elliptic curve, its length is 2l. So, the total length of a consummated attribute credential is 8l.

The ciphertext related data are essentially the data used for authentication or access control. In AC-PKC, they are the ciphertext or signature, whose lengths are 3l and 2l respectively. In ABE, they are the data issued with the ciphertext for decrypting. For objectivity, 4 typical ABE schemes are selected to be comparison objects. One is the lightweight ABE scheme in literature[27], which is based on ECC and with low communication overhead. The modular exponentiation based ABE can be further classified into variable size ciphertext and constant size ciphertext two categories. The former is denoted by ABE-MEV and the latter is denoted by ABE-MEC. Since ABE-MEV schemes always have heavy communication overhead, only one efficient scheme in literature[28] is made as an instance. While ABE-MEC schemes are usually have relative low overhead, we choose two relative efficient schemes in literature[29] and [30] as the comparison instances, which are denoted by ABE-MEC1 and ABE-MEC2 respectively.

TABLE V
COMMUNICATION OVERHEADS

	Public Parameters Size	Required Attributes Size	Ciphertext/Signature Related Data Size
AC-PKC	2*l*	≤8*kl*	*E*:3*l*, *S*:2*l*
ABE-EC	2(*n*+1)*l*	*kl*	(2*k*+1) *l*
ABE-MEV	6.4(*n*+3)*l*	*kl*	6.4(2*k*+1)*l*
ABE-MEC1	25.6*nl*	*kl*	12.8*l*
ABE-MEC2	6.4(*n*+3)*l*	*kl*	12.8*l*

For clarity, the communication overhead comparison is illustrated in Table V. It can be seen that our AC-PKC has distinct advantage over ABE-MEV, ABE-MEC1 and ABE-MEC2. Unfortunately, it is difficult to compare AC-PKC with ABE-EC, because the relations among *L*, *k* and *n* are complicated and uncertain, which depend not only on the security policy and the attribute management strategy but also on the attribute space. In the worst case of $L=k$, if $k \leq 0.4(n-1)$, AC-PKC's overhead is less than or equal to ABE-EC's. In practice, there are usually multiple required attributes in the same attribute credential, and *L* is usually less than or much less than *k* when *k* is far great than 2. To be fair, it can be assumed that $L = k/2$ when *k* is an even number and $L=(k+1)/2$ when *k* is an odd number. In this way, AC-PKC's communication overhead should be (4*k*+5) or (4*k*+9). On average, it has a relative advantage over ABE-EC.

E. Flexibility

The flexibility is a distinct advantage of the proposed AC-PKC. Since all the attribute credentials' key pair of one entity can be combined arbitrarily, AC-PKC can be the foundation of constructing flexible security mechanism and meet various security requirements.

In contrast, the ABE schemes always need to fix the required attributes set in advance, which makes them can only meet the predefined security policy. While the existing CL-PKC schemes are always based on the user's identity, which makes them can only meet the identity based security requirements and can't preserve privacy.

F. Comprehensive Comparison

In order to highlight the traits of the proposed AC-PKC for fog computing based digital manufacture and distinguish it from the existing public key cryptography and ABE schemes, a comprehensive comparison is made from five aspects, which are key management (denoted by KM), applicability to distributed, dynamic fog computing environments(denoted by ADDFC), overhead, flexibility and privacy preserving(denoted by PP). There are 4 typical types of public key cryptography and 3 categories of ABE schemes analyzed here. The comparison results are shown in Table VI.

It should be stated that "Attr" is the abbreviation of attribute and "Credt" is the abbreviation of credential. It can be seen from Table VI that AC-PKC can not only be applied in distributed, dynamic fog computing environments but also provide the flexibility, privacy preserving and the relative low costs in computation and communication simultaneously. It can be said that AC-PKC has better overall performance than the existing public key cryptography and ABE schemes.

TABLE VI
COMPARISON WITH THE EXISTING PUBLIC KEY SCHEME

	KM	ADDFC	PP	Overhead	Flexibility
PKI-PKC	Certificate	No	No	High	No
IB-PKC	User ID	Yes	No	High	No
BP-PKC	User ID	Yes	No	High	No
NP-PKC	User ID	Yes	No	Low	No
ABE-EC	Attr Set	Yes	Yes	Low	No
ABE-MEV	Attr Set	Yes	Yes	High	No
ABE-MEC	Attr Set	Yes	Yes	Low	No
AC-PKC	Attr Credt	Yes	Yes	Low	Yes

VIII. CONCLUSION

To deal with the security and privacy issues in the fog computing based digital manufacturing, AC-PKC is proposed to provide flexible key management and achieve authentication and access control with privacy preserving. Although the performance analysis and comparison show that the proposed scheme can meet the requirement of dynamic security mechanism for fog computing and is more efficient and flexible than the existing schemes, there are still some points for improvement. In the future, we will further refine the attribute description, simplify and optimize the attribute credential to decrease the complexity of the attribute credential verification and low the system overheads.

REFERENCES

[1] M. MUKHERJEE *et al.*, "Security and Privacy in Fog Computing: Challenges," *IEEE Access*, vol. 5, DOI 10.1109/ACCESS.2017.2749422, pp. 19293–19304, Sept. 2017.

[2] Z. Wu, Z. Meng, and J. Gray, "IoT-Based Techniques for Online M2M-Interactive Itemized Data Registration and Offline Information Traceability in a Digital Manufacturing System," *IEEE Trans. Industrial Informatics*, vol. 13, no. 5, DOI 10.1109/TII.2017.2704613,pp. 2397–2405, Oct. 2017.

[3] M. S. de Brito, S. Hoque, R. Steinke, A. Willner, "Towards Programmable Fog Nodes in Smart Factories," in *Foundations and Applications of Self* Systems, IEEE 1st International Workshops on*, DOI 10.1109/FAS-W.2016.57, pp. 236–241,2016.

[4] R. Rios, R. Roman, J. A. Onieva and J. Lopez, "From Smog to Fog: A Security Perspective," in *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference of the Proceeding*. DOI 10.1109/FMEC.2017.7946408, pp. 56–61, 2017.

[5] X. Nan and H. Chen, "Combined Public Key (CPK) System Standard", *Information Security and Communications Privacy*, 2008, no.08, DOI 10.3969/j.issn.1009-8054.2008.08.009, pp. 21–22, Aug. 2008.

[6] OpenFog Consortium, "OpenFog Reference Architecture," Jul. 23, 2017. [Online]. Available: https://www.openfogconsortium.org/ra/.

[7] A. O de Sà, L. F. R. da C. Carmo, and R. C. S. Machado ,"Covert Attacks in Cyber-Physical Control Systems," *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, vol.13, no.4, DOI 10.1109/TII.2017.2676005, pp.1641-1651, Aug. 2017.

[8] P.G. Lopez *et al.*, "Edge-Centric Computing: Vision and Challenges," *ACM SIGCOMM Computer Communication Rev*, vol. 45, DOI 10.1145/2831347.2831354, pp. 37–42, Oct. 2015.

[9] A. Alrawais, A.n Alhothaily, C. Hu, and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," *IEEE Computer Society, IEEE Internet Computing*, vol.21, DOI 10.1109/MIC.2017.37, pp.34–42, Mar. 2017.

[10] Y. Wang, T.o Uehara, R. Sasaki, "Fog Computing: Issues and Challenges in Security and Forensics," in *2015 IEEE 39th Annual International Computers, Software & Applications Conference (COMPSAC) of the Proceeding*, vol. 3, DOI 10.1109/COMPSAC.2015.173, July. 2015, pp. 53–59.

[11] H. Hamid, S. Rahman, M. Hossain, A. Almogren, and A. Alamri, "A Security Model for Preserving the Privacy of Medical Big Data in a Healthcare Cloud Using a Fog Computing Facility with Pairing-Based

Cryptography," *IEEE Access*, vol. 5, DOI 10.1109/ACCESS.2017.2757844, pp. 22313–22328, Sept. 2017.

[12] H. Kim and E. A. Lee, "Authentication and Authorization for the Internet of Things," *Trusting the Internet of Things. IEEE Computer Society*, vol. 19, DOI 10.1109/MITP.2017.3680960, pp. 27–33, Oct. 2017.

[13] Y.W. Law *et al.*, "Wake: Key Management Scheme for Wide-Area Measurement Systems in Smart Grid," *IEEE Communications Magazine*, vol. 51, no. 1, DOI 10.1109/MCOM.2013.6400436, pp. 31–41, Jan. 2013.

[14] M. H. Ibrahim, "Octopus: An edge-fog mutual authentication scheme", *Int. J. Netw. Security*, vol. 18, no. 6, pp. 1089–1101, Nov. 2016. [Online].Available:https://pdfs.semanticscholar.org/8e88/dbc4359940366af820a3e0eb42a15e8a65aa.pdf.

[15] V. G. Martínez, L. H. Encinas, C. S. Ávila, "A survey of the elliptic curve integrated encryption scheme", *Journal of Computer Science and Engineering*, vol. 2, no. 2, AUGUST 2010, pp7-13. [Online].Available: https://www.researchgate.net/profile/Carmen_Sanchez_Avila/publication/255970113.

[16] D. Johnson, A. Menezes, S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)", vol. 1, no.1, DOI 10.1007/s102070100,*International Journal of Information Security*, pp. 36-63, Aug. 2001.

[17] X. Yao, X. Han and X. Du. "A light-weight certificate-less public key cryptography scheme based on ECC," in *The 23rd International Conference on Computer Communication and Networks (ICCCN) of the Proceeding*, DOI 10.1109/ICCCN.2014.6911773, pp. 1–8, Aug. 2014.

[18] X. Yao,X. Han, X. Du, X. Zhou. "A Lightweight Multicast Authentication Mechanism for Small Scale IoT Applications". *IEEE Sensors Journal*. vol.13, no.10, DOI 10.1109/JSEN.2013.2266116, pp.3693–3701,2013.

[19] Standards for Efficient Cryptography Group in Certicom Research, "SEC2: Recommended Elliptic Curve Domain Parameters, Version1.0," pp.10-11,Sept.20,2000.[Online].Available:http://www.secg.org/SEC2-Ver-1.0.pdf.

[20] X. Cao, W. Kou, X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges". *Information Sciences*, 2010, vol. 180, no. 15, DOI10.1016/j.ins.2010.04.002, pp.2895-2903. 2010.

[21] S.S. Al-Riyami, , K.G. Paterson, "Certificateless public key cryptography". In Laih, C.-S. (ed.): ASIACRYPT 2003. LNCS, vol. 2894, DOI 10.1007/978-3-540-40061-529, pp. 452–473. Springer, Heidelberg (2003)

[22] Y. R. Lee, H. S. Lee, "An authenticated certificate-less public key encryption scheme", *Trends in Mathematics Information Center for Mathematical Sciences*, vol. 8, no. 1, June, 2005, pp.177-187.

[23] D. He, J. Chen, R. Zhang. "An efficient and provably-secure certificate-less signature scheme without bilinear pairings". *International Journal of Communication Systems*, 2012, vol. 25, no. 11, DOI 10.1002/dac.1330, pp.1432-1442,2012.

[24] Y. Wang, R. Zhang, "Strongly secure certificate-less signature scheme without pairings", *Chinese Journal on Communications*, vol.34, DOI 10.3969/j.issn.1000-436x.2013.02.011 , no.2, pp 94-99,108, Feb 2013.

[25] L. Harn, J. Ren, C. Lin. "Design of DL-based certificate-less digital signatures". *Journal of Systems and Software*, 2009, vol 82, DOI 10.1016/j.jss.2008.11.844, no.5, pp.789-793,2008.

[26] H. Du, Q. Wen, "Efficient and provably-secure certificate-less short signature scheme from bilinear pairings". *Computer Standards and Interfaces*, vol.31, DOI 10.1016/j.csi.2008.05.013, no.2, pp. 390-394, 2009.

[27] X. Yao, Z. Chen, Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things". *Future Generation Computer Systems*. vol.49, DOI 10.1016/j.future.2014.10.010, pp.104-112,2014.

[28] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization", in: D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi (Eds.), PKC 2011, in: *LNCS*, vol. 6571, Springer, Heidelberg, DOI 10.1007/978-3-642-19379-8_4 pp. 53-70,2011.

[29] C. Wang, J. Luo, "A key-policy attribute-based encryption scheme with constant size ciphertext". in *2012 Eighth International Conference on Computational Intelligence and Security*, DOI 10.1109/CIS.2012.106, pp. 447-451. 2012.

[30] C. Chen, Z. Zhang, D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost", in *International Conference on Provable Security, Lecture Notes in Computer Science*, vol. 6980, DOI 10.1007/978-3-642-24316-5_8, Springer-Verlag, Berlin, Heidelberg, pp. 84-101, 2011.

Xuanxia Yao received her M.S. and Ph.D. degree from University of Science and Technology Beijing (USTB) in 2002 and 2009 respectively. She is an associate professor at the School of Computer and Communication Engineering, USTB. Her research interests include the security issues in IoT, industrial control system, cloud computing, and blockchain as well. Dr. Yao is a member of China Computer Federation (CCF) and a senior member of China Institute of Communications.

Huafeng Kong received his Ph.D , M.S. and B.S. degree from Huazhong University of Science and Technology (HUST), Wuhan, China. He is a research professor in the Third Research Institute of the Ministry of Public Security. His current research interests include network security, Internet of Things and cloud computing. He is the author of more than 20 articles.

Hong Liu received her Ph.D. degree from the School of Electronic and Information Engineering, Beihang University in 2014. She is an associate professor at the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. Her research interests include security in edge computing and industrial control system.

Tie Qiu (M'12-SM'16) received M.S and Ph.D degree from Dalian University of Technology in 2005 and 2012 respectively. He is a Professor at the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University. His research interests include the IoT, Mobile Computing and so on. Prof. Qiu is a senior member of China Computer Federation (CCF) and a Senior Member of ACM.

Huansheng Ning received Ph.D. degree from Beihang University in 2001. He is a Professor at the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include IoT, cyber-physical modeling and privacy preserving. Prof. Ning is a senior member of China Institute of Communications. He serves as an associate editor of IEEE System Journal and IEEE Internet of Things Journal, and gained the IEEE Computer Society Meritorious Service Award in 2013, IEEE Computer Society Golden Core Award in 2014.