IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON COMMUNICATION AND FOG/EDGE COMPUTING TOWARDS INTELLIGENT CONNECTED VEHICLES (ICVS)

Received February 6, 2020, accepted February 17, 2020, date of publication March 12, 2020, date of current version April 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978525

A Lightweight Privacy-Preserving Communication Protocol for Heterogeneous IoT Environment

XI LUO[1,2], LIHUA YIN[1], CHAO LI[1], CHONGHUA WANG[3], FUYANG FANG[4], CHUNSHENG ZHU[5,6], AND ZHIHONG TIAN[1], (Member, IEEE)

[1]Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
[2]PCL Research Center of Cyberspace Security, Peng Cheng Laboratory, Shenzhen 518066, China
[3] Department of Security Technology, China Industrial Control Systems Cyber Emergency Response Team, Beijing 100082, China
[4]Information Science Academy of China Electronics Technology Group Corporation, Beijing 100082, China
[5]SUSTech Institute of Future Networks, Southern University of Science and Technology, Shenzhen 518055, China
[6]PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518066, China

Corresponding author: Chao Li (lichao@gzhu.edu.cn)

This work was supported in part by the Key Research and Development Program for Guangdong Province under Grant 2019B010136001, in part by the National Natural Science Foundation of China under Grant 61872100, in part by the project ''PCL Future Regional Network Facilities for Large-scale Experiments and Applications under Grant PCL2018KP001, and in part by the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019).

ABSTRACT While Internet-of-Things (IoT) significantly facilitates the convenience of people's daily life, the lack of security practice raises the risk of privacy-sensitive user data leakage. Securing data transmission among IoT devices is therefore a critical capability of IoT environments such as Intelligent Connected Vehicles, Smart Home, Intelligent City and so forth. However, cryptographic communication scheme is challenged by the limited resource of low-cost IoT devices, even negligible extra CPU usage of battery-powered sensors would result in dramatical decrease of the battery life. In this paper, to minimize the resource consumption, we propose a communication protocol involving only the symmetric key-based scheme, which provides ultra-lightweight yet effective encryptions to protect the data transmissions. Symmetric keys generated in this protocol are delegated based on a chaotic system, i.e., Logistic Map, to resist against the key reset and device capture attacks. We semantically model such protocol and analyze the security properties. Moreover, the resource consumption is also evaluated to guarantee runtime efficacy.

INDEX TERMS Internet of Things (IoT), key delegation, lightweight protocol, secure communication, symmetric encryption.

I. INTRODUCTION

IoT devices used in smart homes [1], smart city [2], Intelligent Connected Vehicles (ICVs) [3] and so forth have become a fundamental part of our modern life. It is estimated that there will be 25 billion ''connected'' IoT devices by 2020, and the global economy will be impacted by the IoT sector's expansion by more than $6 trillion by 2025 [4]. Such devices facilitate the automation, adaptability, efficiency, and convenience of our living space.

In most cases, devices or sensors in IoT environment are connected using wireless channels. Such channels are usually unreliable and render users' privacy-sensitive data exposed to eavesdroppers. For example, the Electronic Control Units (ECUs) that operate the logic of ICVs are susceptible to adversarial manipulations, as automotive communication standards generally implement no authenticity between such software interaction [5]. A natural solution to secure the privacy-sensitive data is to implement end-to-end encryption to protect the insecure communication. Traditional protocols designed for IoT environment usually utilize the asymmetric key-based schemes to initialize communications. Such techniques are mainly based on RSA algorithm [6], Diffie-Hellman Key Exchange (DHKE) algorithm [7] or Elliptic Curve Cryptography (ECC) algorithm [8], which require extensive computation resource. However, only 5% extra CPU usage of battery-powered devices can lead to less than a year of battery life [9].

Symmetric cryptography is a better choice for IoT network due to the negligible depletion, but designing such a

The associate editor coordinating the review of this manuscript and approving it for publication was Changqing Luo.

communication protocol is challengeable for the following reasons. First, since symmetric keys can be used to both encryption and decryption, the secrecy of pre-distribution process have to be guaranteed. Second, devices will share the keys with others for authentication and thus are vulnerable to device capture attack [10]. Third, when attackers pilfer the long term symmetric keys, he can eavesdropper any message at any time he want. Existing symmetric key-based approaches such as [11]–[18] dedicate in improving the probability of establishing a direct pairwise key between two neighbor sensing nodes. These approaches are either vulnerable to the device capture attack or only feasible to small networks due to the high complexity of these distributed algorithms. Some other symmetric key-based works for IoT environment like [1], [19] can just be applied in simple IoT environment such as smart homes where only authorized entities have access to it. In contrast, most types of IoT systems such as ICVs work in outdoor environment can be easily approached by attackers. Unlike above works, our goal is to design a generic symmetric key-based secure protocol applicable for heterogeneous IoT environment, especially the easily-approached and low powered devices (e.g., sensors in ICVs).

In this paper, we present a lightweight yet secure protocol based solely on symmetric cryptography. We utilize a chaotic system, i.e., Logistic Map, to implement key delegation scheme. Chaotic system which is indeterminate, unpredictable and unrepeatable has been widely adopted in various cryptography applications in recent years [1], [20]–[23]. In pre-distribution phase, a parameter and an initial value of Logistic Map are random generated and assigned as default configuration to each device. Such parameter and initial value (i.e., the key) are used for authentication when device firstly connect to a control center, e.g., cloud platform. Afterwards, this control center can assigned another pair of parameter and initial value for device-to-device communication. All the keys are updated synchronously by iterating the Logistic Map with specific parameters and initial values. Since the parameters are kept secret and would never be transmitted through public channels, adversaries can never calculate the communication key unless they capture the devices. Meanwhile, since the parameters are random selected and not shared with any others, attackers can only obtain the data associated with compromised devices. This means our protocol is resistant to device capture attack. In summary, the contributions of our research are as following.

1. We present a secure communication protocol based solely on symmetric cryptography scheme. Since it works independent on asymmetric cryptography, the resource cost is extremely low. Moreover, it is a general solution for secure end-to-end data exchange among devices in heterogeneous IoT environment.
2. We design an effective key delegation mechanism based on a chaotic system, i.e., Logistic Map. This mechanism helps to resist the device capture attack by random selecting the parameters and initial values. Besides, we implements a synchronous rekey scheme to prevent key reset attack.
3. We comprehensively evaluate the security properties and resource cost of our protocol. The result illustrates that the safety and efficacy. Moreover, in the comparison analysis, our protocol outperforms existing chaotic system-based approach [1] for smart home systems.

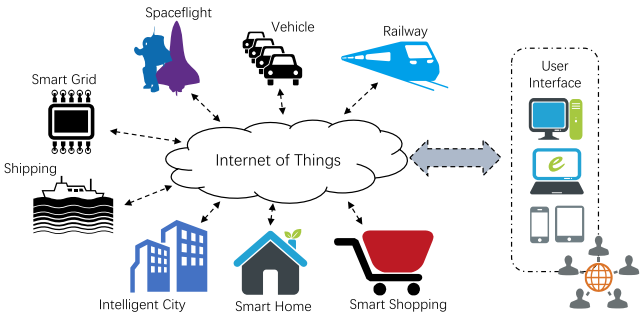


FIGURE 1. Overview of IoT network architecture.

Organization. The rest of the paper is organized as follows. In Section II, we illustrate the background and motivation. In Section III we provide our threat model and proposed protocol. In Section IV, we evaluate our protocol. In Section V, we discuss our work. In Section VI, we briefly analyze the previous works. At last, in Section VII, we conclude our work.

II. BACKGROUND

A. IoT NETWORK

The Internet of Things consists of a large number of devices that are interconnected through the Internet. In Fig. 1, we present a generic IoT network architecture in which eight different applications are depicted, i.e., railway, vehicle, spaceflight, smart grid, shipping, intelligent city, smart home and smart shopping. Several IoT devices, such as sensors and actuators, are deployed to carry out these applications. All these devices can connect to the Internet via traditional cable or wireless networks. Users can control the smart systems as well as access the information collected by such devices through the user interface.

The interior network structures of smart applications differ from each other. For example, smart home systems which can be accessed solely by the authorized users usually contain simple centralized networks [1], while large scale industrial IoT systems like intelligent city implement hierarchy communication infrastructure [9] serving for a large number of devices and users. Given the different architectures of such network structures, distinct vulnerability would be engendered. Smart home users primarily suffer the risks of privacy leakage, whereas the intelligent city designer should concern about the device capture attack as well. It is challengeable to design such a generic communication protocol for the devices deployed in different IoT environment.

B. MOTIVATION

Secure end-to-end communication can be categorized into two types, i.e., asymmetric and symmetric, based on the type of cryptosystem. In a general example of asymmetric type, when an entity A tries to communicate to another entity B, A utilizes the public key of B to encrypt messages while B leverages the private key for decryption. Such messages can contain a temporal symmetric session key. Then, the further communication is enciphered and deciphered using this symmetric key shared by both entities.

The public key can be transfered through unreliable channel because it can't be used to decrypt the ciphertext. This significantly facilitates the key delegation. However, the encryption and decryption processes based on asymmetric algorithm are computationally intensive and thus consume a large amount of energy and computational resources that are limited in IoT devices. In earlier work [24], the authors illustrate that the asymmetric key schemes is at least 100 times more costly than symmetric key schemes. Such energy and time consumption is disastrous when serving for large scale data analytics in a timely manner [25].

The main challenge of implementing a symmetric key-based protocol is designing a robust key delegation mechanism. The keys can't be distributed through public networks and require effective renewing and revocation strategies to maintain freshness. In our work, the symmetric keys are managed in a synchronous mode based on chaotic system which can guarantee the confusion and diffusion of our cryptosystem.

C. CHAOTIC SYSTEM

Chaotic systems have been used to solve cryptography issues since the sensitivity to parameters and initial conditions, ergodicity, and pseudorandom behavior conform to the analogous requirements for a good cryptosystem. One of the most famous chaotic systems, i.e., Logistic Map, is pretty suitable for IoT environment because its simplicity of computation. Logistic Map is a nonlinear map given by

$$x_{i+1} = \mu x_i(1 - x_i) \tag{1}$$

where $\mu \in (1, 4]$, $x_i \in (0, 1)$ and $i \in N^+$. When $3.5699\ldots < \mu \leqslant 4$, the system is chaotic. Meanwhile, such chaotic system is deterministic since the random output is completely determined by x_0 and μ. Since Logistic Map is pseudorandom, and the result values fall in an infinite space in $(0, 1)$, it can be utilized to compute private keys. The chaotic behavior with $x_0 = 0.7648$ and $\mu = 3.987$ in 500 iterations is exemplified in Fig. 2.

III. PROTOCOL DESIGN

In this section, we present threat model as well as design details of our lightweight yet secure communications protocol.

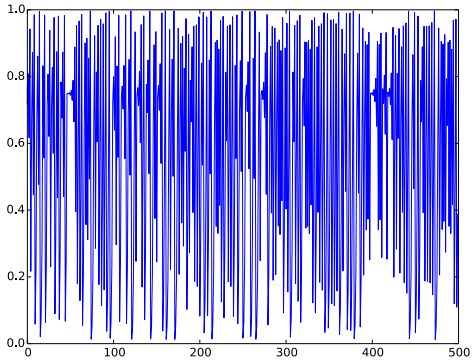


FIGURE 2. Chaotic behavior of Logistic Map with $x_0 = 0.7648$ and $\mu = 3.987$ in 500 iterations.

TABLE 1. Notations.

Notation	Description
A, B, S	Device A, Device B and Control Center S
M_*	Plaintext message
ID_*	Device Identifier
μ_*, x_*^0, T_*^j	Pre-distributed parameters and initial value and timestamp
$h(-), [-]_x$	Hash and encryption function
$MAC_*, <-,-,->$	Message Authentication Code and pair function

A. THREAT MODEL

We consider integrity and confidentiality violations caused by a Dolev-Yao adversary [26] in an IoT environment. With respect to the behavior of adversaries, we will focus attention on "aggressive" eavesdroppers. That means someone who first taps the communication line to obtain messages and then tries everything he can in order to discover the plaintext. According to Dolev-Yao model, under the premise that the encryption algorithm can't be brute cracked, we assume the following abilities about a adversary.

a. He can obtain, intercept and replay any message transferring through the network. Moreover, he can utilize his knowledge to insert new messages in the network.
b. He is a legitimate user/device in the network environment, i.e., possess a legitimate identity, and therefore can connect to any other user/device.
c. He has the opportunity to receive the message from any user/device and extents his knowledge.

In an IoT environment, the second property of the adversary can be achieved by capturing a smart device in wild. This is also called the device capture attack. We assume that the adversary has ability to extract any data he wants in such a captured device, and thus can work as a legitimate user/device in the IoT network. Additionally, in general, an IoT environment implements a control center to schedule the entire communications. Such control center is assumed uncompromising in our work.

B. OVERVIEW OF OUR PROPOSED PROTOCOL

Prior to introducing the protocol, we first summarize the notations that will be used in the following content in Table 1.

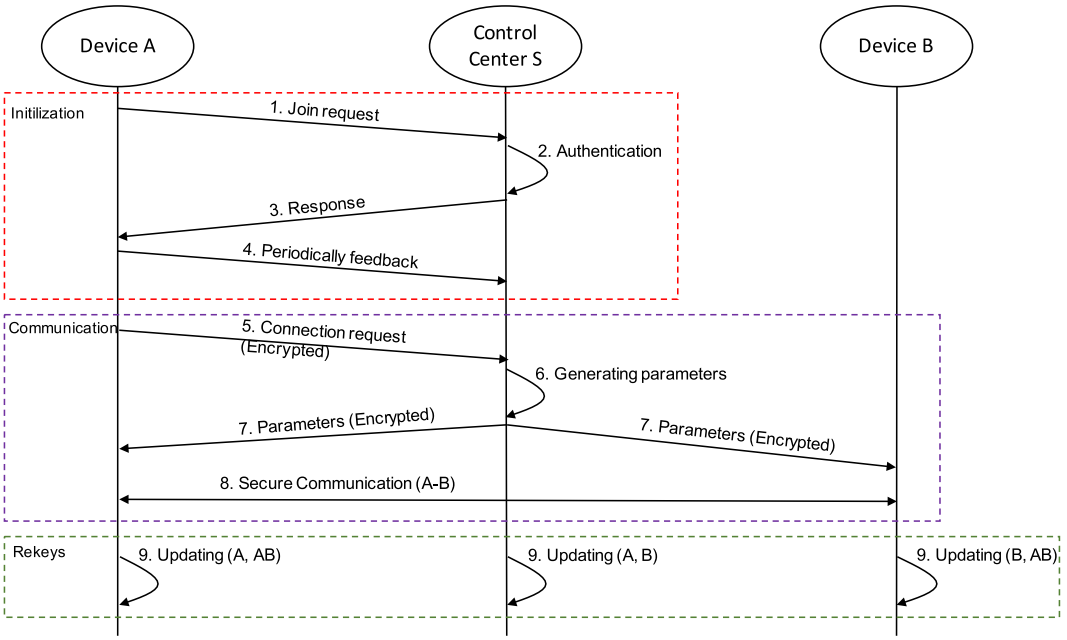


FIGURE 3. Overview of the proposed protocol.

There are three entities in our protocol, i.e., devices A, B and control center S. Symbol M_* denotes the plaintext message. ID_* denotes unique identifier of a device or control center. For example, ID_A represents the identifier of device A. μ_*, x_*^0 and T_*^j represent the random selected parameter, initial value (according to Logistic Map) and timestamp, respectively. In this work, we assume the parameters are hardcoded in the firmware of devices, and the manufacturer stores them in where can be accessed by S. $h(-)$ and $[-]_x$ denote hash and encryption function, respectively. MAC_* denotes Message Authentication Code while $< -, - >$ denotes pair function.

In our protocol, as shown in Fig. 3, when device A attempts to join a IoT network with a control center S, an initialization process is launched by A to send a message containing the identity and parameter of A to S. If S successfully authenticate this message, it will return a response that indicates A successfully joining the network. Next, A will periodically connects to S for either submitting collected data or making heartbeat check. Messages exchanged in initialization process are all encrypted based on symmetric key generated using the built-in Logistic Map parameters.

After device A successfully joins in the network, when it attempts to communicate to another device B, it should first request control center S to generate a new pair of parameter and initial value. Then, S transmits them to both A and B based on symmetric communication channel built previously in initialization process. The new pair of parameter and initial value is utilized to calculate and update a temporal session key between A and B. As a result, our protocol builds secure communication channels both for D2C (device to control center) and D2D (device to device) scenario.

In the key updating phase, all the entities iterate their initial values, i.e., the keys, in a synchronous mode based on Logistic Map. The timestamps of devices are calibrated when they communicate to control center S. Hence, our protocol is resistant against key reset attacks that can be launched in asynchronous scenario. In the ensuing content, we will introduce the design details of our protocol.

C. INITIALIZATION

In initialization process, device A holds three items, i.e., a unique identifier ID_A, Logistic Map parameter μ_A and initial value x_A^0, for authentication. Time stamp T_A^0 is also required to verify the aliveness. Details of the initialization process are presented in the following, as shown in Alg. 1.

1) Device A initiates a join message M_A and encrypts it with current timestamp T_A^0 based on key x_A^0. Then we have ciphertext $C_A = [M_A \parallel T_A^0]_{x_A^0}$. Next, A generates the message authentication code (MAC) $MAC_A = h(C_A \parallel \mu_A)$. Afterwards, A sends the ciphertext C_A and MAC_A as well as identifier ID_A, i.e., $ID_A \parallel C_A \parallel MAC_A$ to control center S.
2) When control center S receives the message containing $ID_A \parallel C_A \parallel MAC_A$, it first looks up the database to find the key x_A^0 and parameter μ_A indexed by ID_A. If it can not find such a record, an non-existent device error code is responded to A. Otherwise, S verifies the integrity by comparing $h(C_A \parallel \mu_A)$ to MAC_A. If $h(C_A \parallel \mu_A) == MAC_A$, S decrypts the ciphertext C_A using x_A^0 and obtains M_A and T_A. Otherwise, S rejects such join request. Next, it certifies the timestamp T_A. if $|T_S^i - T_A^i| > \lambda$, where λ is the time limit threshold, a list L_{id} of device identifiers

Algorithm 1 Initialization

For device A***:***
Generating M_A
Loading ID_A, μ_A, x_A^0, T_A^0
Encrypting $C_A = [M_A \parallel T_A^0]_{x_A^0}$
Computing $MAC_A = h(C_A \parallel \mu_A)$
Sending $ID_A \parallel C_A \parallel MAC_A$

For control center S***:***
Receiving $ID_A \parallel C_A \parallel MAC_A$
Searching μ_A and x_A^0 via ID_A
Computing $MAC = h(C_A \parallel \mu_A)$
if $MAC \neq MAC_A$ **then**
    Authentication failed
    Return
end if
Loading current timestamp T_S^0
Decrypting $C_A = [M_A \parallel T_A^0]_{x_A^0}$
Reading M_A and T_A^0
if $|T_S^0 - T_A^0| > \lambda$ **then**
    Authentication failed
    Return
end if
Sending $[L_{id}, T_S^0]_{x_A^0}$ to A

Algorithm 2 Establish Session Key

For device A***:***
Loading ID_A, ID_B, μ_A, x_A^i, T_A^i
Encrypting $C_A = [ID_B \parallel T_A^i]_{x_A^i}$
Computing $MAC_A = h(C_A \parallel \mu_A)$
Sending $ID_A \parallel C_A \parallel MAC_A$

For control center S***:***
Receiving $ID_A \parallel C_A \parallel MAC_A$
Searching μ_A and x_A^i via ID_A
Computing $MAC = h(C_A \parallel \mu_A)$
if $MAC \neq MAC_A$ **then**
    Authentication failed
    Return
end if
Loading current timestamp T_S^i
Decrypting $C_A = [ID_B \parallel T_A^i]_{x_A^i}$
Reading ID_B and T_A^i
if $|T_S^i - T_A^i| > \lambda$ **then**
    Authentication failed
    Return
end if
Generating $< x_{AB}^0, \mu_{AB} >$
Loading x_B^i,μ_b
Encrypting $C_{AB} = [< x_{AB}^0, \mu_{AB} > \parallel T_S^i]_{x_A^i}$
Encrypting $C_{BA} = [< x_{AB}^0, \mu_{AB} > \parallel T_S^i]_{x_B^i}$
Computing $MAC_{AB} = h(C_{AB} \parallel \mu_A)$
Computing $MAC_{BA} = h(C_{BA} \parallel \mu_B)$
Sending $ID_S \parallel C_{AB} \parallel MAC_{AB}$ to A
Sending $ID_S \parallel C_{BA} \parallel MAC_{BA}$ to B

For device B***:***
Receiving $ID_S \parallel C_{BA} \parallel MAC_{BA}$
Loading μ_B, x_B^i and current timestamp T_B^i
if $MAC \neq MAC_B$ **then**
    Authentication failed
    Return
end if
Decrypting $c_{BA}[< x_{AB}^0, \mu_{AB} > \parallel T_S^i]_{x_B^i}$
Reading x_{AB}^0, $\mu_A B$ and T_S^i
if $|T_B^i - T_S^i| > \lambda$ **then**
    Authentication failed
    Return
end if
Computing $MAC = h(< x_{AB}^0, \mu_{AB} > \parallel \mu_B)$
Admit x_{AB}^0 and μ_{AB}

of reachable devices is encrypted with T_S^0 based on x_A^0 and sent to A. Or else, S reject such join request.

3) After above process, device A builds a secure communication channel with control center S. Generally, devices, e.g., A, in this IoT network periodically connect to S to upload the monitored data, e.g., D_A. Similar to the first step, D_A is encrypted along with the timestamp and a MAC is computed based on μ. The device ID, ciphertext and MAC are sent to S for authentication.

The timestamp of A may be inconsistent with that of S due to different timing instrument. To solve this problem, the user can manually configure the timestamp or synchronize it via short-distance communication channels (e.g., bluetooth) to ensure the consistence when deploying such devices. Therefore, adversaries is not able to manipulate the timestamps during initialization process. Note that, the third phase is more like communication between device and control center (D2C) rather than the initialization process. We state it here just for a better understanding of D2C interaction.

D. DEVICE-TO-DEVICE COMMUNICATION

In our protocol, each device holds a persistent key only for D2C communication. As shown in Fig. 3, when a device attempts to connect to another, it should first request the control center to establish a session key for secure communication. Such key can be reserved by both devices for either long term or temporal use. The device-to-device communication process is separately stated in Alg. 2 and Alg. 3, including key establishing and communication phases. The details are as following.

1) As shown in Alg. 2, device A reads the identifier of device B, i.e., ID_B, from L_{id}. Then A encrypts ID_B with timestamp T_A^i using key x_A^i and obtains

Algorithm 3 Communication

For device B:
Loading ID_B, x_{AB}^0, μ_{AB}, T_B^i
Generating M_B="Hello A, I am B."
Encrypting $C_{AB} = [M_B \parallel T_B^i]_{x_{AB}^0}$
Computing $MAC_{AB} = h(C_{AB} \parallel \mu_{AB})$
Sending $ID_B \parallel C_{AB} \parallel MAC_{AB}$ to A

For device A:
Receiving $ID_B \parallel C_{AB} \parallel MAC_{AB}$
Loading μ_{AB}, x_{AB}^0 and current timestamp T_A^i
Computing $MAC = h(C_{AB} \parallel \mu_{AB})$
if $MAC \neq MAC_b$ **then**
    Authentication failed
    Return
end if
Decrypting $C_{AB} = [M_B \parallel T_B^i]_{x_{AB}^0}$
Reading M_B and T_B^i
if $|T_A^i - T_B^i| > \lambda$ **then**
    Authentication failed
    Return
end if
Success and starting data transmission

$C_A = [ID_B \parallel T_A^i]_{x_A^i}$. Next, A computes $MAC_A = h(C_A \parallel \mu_A)$ for integrity verification. ID_A, C_A and MAC_A are then sent to S to initialize the D2D communication with B.

2) When control center S receives the message $ID_A \parallel C_A \parallel MAC_A$, it first loads μ_A, x_A^i and timestamp T_S^i. Then, it compares $h(C_A \parallel \mu_A)$ to MAC_A to verify integrity and authenticate the identity of A. If success, then, it decrypts the ciphertext C_A and certifies the timestamp T_A. If T_A is available, a new pair of parameter μ_{AB} and initial value x_{AB}^0 are generated by S. S then generate C_{AB} and C_{BA} by computing $C_{AB} = [< x_{AB}^0, \mu_{AB} > \parallel T_S^i]_{x_A^i}$ and $C_{BA} = [< x_{AB}^0, \mu_{AB} > \parallel T_S^i]_{x_B^i}$, respectively. $ID_S \parallel C_{AB} \parallel MAC_{AB}$ and $ID_S \parallel C_{BA} \parallel MAC_{BA}$ are sent to A and B to assign the pair $< x_{AB}^0, \mu_{AB} >$.
3) As shown in Alg. 3, when devices A and B receive the above messages, respectively, they conduct similar authentication processes and we just present that conducted by B in Alg. 2 for space. B first verifies the integrity by examining whether $h(C_{BA} \parallel \mu_B)$ equals to MAC_{BA}. Then, B decrypts C_{BA} and extracts x_{AB}^0, μ_{AB} and T_S^i. After successfully verifying T_S^i, B admits x_{AB}^0 and μ_{AB} which are further used to communicate to A.
4) After both devices A and B admit x_{AB}^0 and μ_{AB}, they can conduct secure communications. In this case, A waits for B to initialize the session since B works as a receiver and is entitled to reject the connection request. Rejection may caused by the access control strategies. For example, if computation resources of B is exhausted, it will reject communication request from A. Such access control strategies are not concerned in this paper and will be briefly discussed in Section V. As shown in Alg. 3, B initializes a greeting message containing $C_{AB} = [M_B \parallel T_B^i]_{x_{AB}^0}$, $MAC_{AB} = h(C_{AB} \parallel \mu_{AB})$ and ID_B and sends it to A. After A receives this message and verify time availability of timestamp and authenticate the identity of B, the secure communication channel is accomplished.

Note that the index i used in above symbols like x_A^i corresponds to the number of iterations of Logistic Map, i.e., the updating times of x_*^0. In the following content, we will introduce the key updating mechanism in detail.

E. KEY DELEGATION

Unlike the asymmetric encryption, once the symmetric key used in our protocol is exposed adversaries can persistently eavesdrop the communications. Hence, the keys have to be updated frequently to guarantee the privacy of user data. As stated previously, we implement a synchronous key updating scheme to maintain freshness of the persistent symmetric keys.

The main challenge to develop a synchronous scheme is the consistence of timestamps recorded on different devices. In IoT environment, not all the devices are necessary and have enough resource to access the Internet, and thus their clock qualities only depend on the build-in time counters. Hence, inconsistencies will exhibit among different devices over time.

As shown in Alg. 1, Alg. 2 and Alg. 3, all the communication messages contain timestamp of the sender. Since we assume control center S is uncompromising, only the timestamp of S is considered as official and calibration is conducted every time a device sending the heartbeat feedback. Assuming the feedback interval t_f and the rekey period t_r, we set $t_r > t_f$ to ensure the time consistence. For instance, if the key is updated per an hour, the feedback interval could be 1min and the timestamp is calibrated sixty times per update.

When rekey time is up, the devices compute $x_{i+1} = \mu x_i(1 - x_i)$ to update the keys. It is valuable to mention that the session key x_{AB}^i is also updated until the expire time assigned by control center. The expire time is determined by the mission A requests. This determination strategy is not concerned in this paper and will be briefly discussed in Section V like the access strategy mentioned above.

IV. EVALUATION

In this section, we formalize the proposed protocol and analyze the security properties including secrecy and non-injective synchronization [27] using scyther proof tool [28]. Next, we discuss the resistance against several easily-conducted attacks. Then, we detail the implementation and examine the performance of this protocol. At last, we compare our protocol to the similar chaotic-based schemes [1] designed for smart home system.

A. SECURITY ANALYSIS

According to the protocol specification language presented in paper [28], we briefly define our protocol $LT \stackrel{def}{=} \{C, S\}$ as following.

$$\begin{aligned} I &\stackrel{def}{=} \langle I_1, I_2 \rangle \\ R &\stackrel{def}{=} \langle R_1, R_2 \rangle \\ I_1 &= Send_1(\{|m_1|\}_{x^i}, h(\{|m_1|\}_{x^i}, \mu))) \\ R_1 &= Recv_1(\{|m_2|\}_{x^i}, h(\{|m_2|\}_{x^i}, \mu))) \\ R_2 &= Send_2(\{|m_3|\}_{x^i}, h(\{|m_3|\}_{x^i}, \mu))) \\ I_2 &= Recv_2(\{|m_4|\}_{x^i}, h(\{|m_4|\}_{x^i}, \mu))) \end{aligned} \tag{2}$$

Role I and R represent the initiator and responder in a communication, respectively. Both of them contain two role steps. Each role step sends or receives a message consisting of a pair of ciphertext $\{|m|\}_{x^i}$ and MAC $h(\{|m|\}_{x^i}, \mu))$. For device to center (D2C) communication, x^i and μ are hardcoded by the manufacturer. For device to device (D2D) communication, they are assigned by the control center.

Given a initiator device who completes its role with an uncompromising server, LT protocol guarantees the secrecy of text m and non-injective synchronization with a server. Non-injective synchronization expresses that the messages are transmitted exactly as prescribed by the protocol description. In this paper, we only proof the two properties for D2C communication. This is because that the two properties guarantee the security of the assigned x^i and μ for D2D communication, such that the security of D2D scenarios can be proofed in a similar way.

According to scyther proof tool, secrecy ϕ_{sec} and non-injective synchronization ϕ_{auth} can be formalized as following.

$$\begin{aligned} \phi_{sec}(tr, th, \sigma) &\stackrel{def}{=} \forall i \in TID. \\ role_{th}(i) &= I \wedge I, R \notin Compr \\ &\Rightarrow m\#i \notin knows(tr) \end{aligned} \tag{3}$$

$$\begin{aligned} \phi_{auth}(tr, th, \sigma) &\stackrel{def}{=} \forall i \in TID. \\ role_{th}(i) &= I \wedge \sigma(r, i) \notin Compr \wedge (i, I_2) \in steps(tr) \\ &\Rightarrow (\exists j \in TID. role_{th}(j) = R \\ &\quad \wedge \sigma(x^i, i) = \sigma(x^i, j) \\ &\quad \wedge \sigma(\mu, i) = \sigma(\mu, j) m_1\#i = \sigma(m3, j) \\ &\quad \wedge (i, I_1) \prec_{tr} (j, R_1) \wedge (j, R_2) \prec_{tr} (i, I_2)) \end{aligned} \tag{4}$$

tr contains the event traces ordered in our protocol, th contains the $role \times rolesteps$ corresponding to all threads which denote instances of roles, σ contains a variable store storing for each variable and thread identifier ($tid \in TID$). (tr, th, σ) maintains the state of our protocol. $steps(tr)$ contains all the step event (tid, s) in tr, while $knows(tr)$ consists of the messages that adversary learns in tr. $Compr$ denotes the set of compromised devices and $\prec_{tr}$ represents preceding order in tr. In this work, we assume the adversary has ability to eavesdrop and intercept any message transfered in the network and obtain any data he wants in compromised devices. In the following, we utilize the rules to prove such properties and all these rules are referenced from [28].

1) PROOF OF SECRECY

Suppose the secrecy ϕ_{sec} does not hold for some states (tr, th, σ). Then there is a thread i such that $role_{th}(i) = I, \sigma(r, i) \notin Compr$, and $m\#i \in knows(tr)$. Scyther proof tool determines the possible ways that $m\#i \in knows(tr)$ within the *CHAIN* rule, which converges into the following conclusions.

$$\begin{aligned} &1). \quad ((m\#i) \in IK_0) \vee \\ &2). \quad (\exists x. m\#i = h(x) \wedge x \prec_{tr} h(x)) \vee \\ &3). \quad (\exists x. m\#i = \{|x|\}_k \wedge x \prec_{tr} \{|x|\}_k \wedge k \prec_{tr} \{|x|\}_k) \vee \\ &4). \quad (\exists x, y. m\#i = (x, y) \wedge x \prec_{tr} (x, y) \wedge y \prec_{tr} (x, y)) \vee \\ &5). \quad (\exists Send_l(pt). \exists role_{th}(tid) \\ &\qquad \wedge chain_{tr}((tid, Send_l(pt)), inst_{\sigma,tid}(pt), m\#i)) \end{aligned} \tag{5}$$

IK_0 denotes the initial knowledge of adversary, which, in this work, includes the public data like device ID and any information reserved in compromised devices in $Compr$. Case 5) states that there are devices initializing threads and sending message pt which can be used to infer m by conducting zero or more decryption and projection. $Chain_{tr}(E, u, u')$ is defined as following.

$$\begin{aligned} Chain_{tr}(E, u, u') \stackrel{def}{=} &(u' = u \wedge (\forall e \in E. e \prec_{tr} u)) \\ &\vee (\exists x, k. u' = \{|x|\}_k \\ &\wedge (\forall e \in E. e \prec_{tr} \{|x|\}_k) \\ &\wedge chain_{tr}(\{\{|x|\}_k, k^{-1}\})) \\ &\vee (\exists x, y. u' = (x, y) \\ &\wedge chain_{tr}(E, x, u) \\ &\vee chain_{tr}(E, y, m)) \end{aligned} \tag{6}$$

Case 1), 2), 3) and 4) in Equation 5 infer that either the key x^i, plaintext m or parameter μ are included in adversary's initial knowledge IK_0. This is contrary to the assumption $role_{th}(i) = I, I \notin Compr$. Case 5) states that either x^i or μ are send by at least one device. However, in our protocol, neither x^i nor μ is send through the public channel. Hence, we conclude that our protocol conforms to the secrecy property ϕ_{sec}.

2) PROOF OF NON-INJECTIVE SYNCHRONIZATION

For every state (tr, th, σ) and every thread i such that $role_{th}(i) = I, \sigma(s, i) \notin Compr$, and $(i, I_2) \in step(tr)$, the non-injective synchronization guarantees that there is a thread j holds the following conclusion.

$$\begin{aligned} role_{th}(j) = R \\ \wedge \sigma(x^i, i) = \sigma(x^i, j) \\ \wedge \sigma(\mu, i) = \sigma(\mu, j) \\ \wedge (i, I_1) \prec_{tr} (j, R_1) \\ (j, R_2) \prec_{tr} (i, I_2) \end{aligned} \tag{7}$$

Since $(i, I_2) \in tr$, we have $(\{m4\#i\}_{x^i}, h((\{m4\#i\}_{x^i}, \mu\#i))) \prec_{tr} (i, I_2)$ using the rule *INPUT*. Then, according to the rule *KNOWN* and *PAIR*, $\{m4\#i\}_{x^i} \in knows(tr)$ and $h((\{m4\#i\}_{x^i}, \mu\#i)) \in knows(tr)$. When applying the *CHAIN* rule and removing trivial cases, considering $(\{m4\#i\}_{x^i}, \mu\#i)$ as a whole, i.e., $m_w\#i$, we yields the following result.

$$\begin{aligned} &1)(m_w\#i \prec_{tr} h(m_w\#i)) \vee \\ &2)(\exists j.role_{th}(j) = R \wedge (j, R_2) \prec_{tr} h(\sigma(m3, j)) \\ &\quad \wedge h(\sigma(m3, j)) = h(m_w\#i) \end{aligned} \tag{8}$$

Case 1) contradicts the secrecy property we proved previously. From $h(\sigma(v, j)) = h(m_w\#i)$ and the injectivity of hash algorithm, we have $m_w\#i = \sigma(m_w, j)$ and thus have $\mu\#i = \mu\#j$, i.e., $\sigma(\mu, i) = \sigma(\mu, j)$. Then, we have $\sigma(x^i, i) = \sigma(x^i, j)$ since μ and x_0 are one-to-one correspondence and can uniquely determine x^i. From $(j, R_2) \prec_{tr} h(\sigma(m_w, j))$ (rule *INPUT*), $h(\sigma(m_w, j)) = h(m_w\#i)$ and $m_w\#i = \sigma(m3, j)$, it follows that $(j, R_2) \prec_{tr} (i, I_2)$.

To establish Equation 7, it remains to be shown $\sigma(x^i, i) = \sigma(x^i, j) \wedge (i, I_1) \prec_{tr} (i, R_1)$. From $(j, R_2) \prec_{tr} (i, I_2)$, we get $(j, R_1) \prec_{tr} (j, R_2)$ (rules *EXEC* and *ROLE*). Hence, $\{m2\#i\}_{x^i} \prec_{tr} (j, R_1) \wedge h((\{m2\#i\}_{x^i}, \mu\#j)) \prec_{tr} (j, R_1)$ (rule *EXEC*). Using rule *KNOWN* and rule *PAIR*, we have $\{m_1\#i\}_{x^i} \in knows(tr)$ and yields

$$\begin{aligned} &1)(m_2\#i \prec_{tr} \{m_2\#i\}_{x^i} \wedge x^i \prec tr\{m_2\#i\}_{x^i}) \vee \\ &2)(\exists i.role(tr)(i) = I \wedge (i, C_1) \prec_{tr} \{m\#i\}_{x^i\#i} \wedge \\ &\quad \{m_1\#i\}_{x^i\#i} = \{m_2\#j\}_{x^i\#j}) \end{aligned} \tag{9}$$

Case 1) again contradicts the secrecy property we proved due to $m_2\#i \in knows(tr)$ (rule *KNOWN*). Case 2) implies $\{m_1\#i\}_{x^i\#i} = \{m_2\#j\}_{x^i\#j}$, so we get

$$(i, I_1) \prec_{tr} \{m_1\#i\}_{x^i\#i} = \{m_2\#j\}_{x^i\#j} \prec_{tr} (j, R_1) \tag{10}$$

where $x^i\#i = x^i\#j$ (i.e., $\sigma(x^i, i) = \sigma(x^i, j)$ proved above) and $(i, I_1) \prec_{tr} (j, R_1)$.

B. RESISTANCE AGAINST TYPICAL ATTACKS

To limit the resource consumption of our protocol, we chose Advanced Encryption Standard (AES) algorithm and Secure Hash Algorithm (SHA) as the encryption and hash function, respectively. AES, also known by its original name Rijndael, is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [29], while the SHA are a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) [30]–[32].

More precisely, AES-128 and SHA-256 are selected in this work, since they have been proven the outperformed energy efficacy and safety in the previous works [24] and [33]. Considering both the key security and resource cost, we empirically set the rekey interval 10mins and heartbeat period 1min. Note that, both values can be set according to the exact environment where the protocol is deployed.

1) BRUTE FORCE ATTACK

Cracking AES-128 session key through brute-force manner is impractical (with $\frac{1}{2^{128}}$ probability) as illustrated in previous work [1]. Even if the adversaries extract AES keys, it is impossible for them to predict the next updated keys due to the chaotic behavior of Logistic Map. In our protocol, a key used in a device is updated in 10mins, such that the next round communication will be initialized using a new key. In this case, the keys cracked by the adversaries will be expired soon and therefore have negligible validity.

2) DEVICE CAPTURE ATTACK

In IoT environment, the devices are usually not physically protected. Hence, an adversary is able to physically capture the devices, and then extracts information stored in those captured sensing devices to compromise communication between other devices. The resistance against such attack is measured by the proportion of compromised communications in which the compromised sensing devices are not directly involved [10].

In our protocol, the secret information of D2C communication will never been exchanged by other entities. Therefore, the D2C communication is absolutely resistant against device capture attack. On the other hand, the session keys utilized in D2D communication is assigned through the D2C channel, a device only knows the keys used in self-involved communications, which also represents 100% resistance.

3) KEY RESET ATTACK

The key reset attack abuses design or implementation flaws in cryptographic protocols to reinstall an already-in-use key [34]. The reseting key's associated parameters such as transmit nonces and replay counters are then received by the adversary.

Such attack is especially harmful to the long-term symmetric key-based communication in which a key can be leveraged to decrypt all the messages. We stress that the asynchronous rekey scheme is essentially vulnerable to this attack. This is because asynchronous scheme contains a key updating order in which an endpoint updates the key and then notices another to update. Once the last confirm message intercepted, the key will be inconsistent between these two endpoints.

In our protocol, we implement a synchronous update scheme and the timestamp is frequently calibrated through reliable D2C channel rather than the clock servers in Internet. Even though an adversary is capable of tampering the time of control center by interpolating the communication message sent by unreliable clock servers, he can't impact time consistence between devices and the control center. Hence, our protocol is resistant against key reset attack.

C. PERFORMANCE

1) TIME CONSUMPTION

Given the process shown in Alg. 1, Alg. 2 and Alg. 3, we define the asymptotic time costs of key generation,

symmetric encryption and decryption, hash function as T_g, T_{en}, T_{de} and T_h, respectively. Moreover, we let T_{wr} and T_{re} to represent the time costs of the I/O operations conducted by control center.

For the D2C communication, a device conducts one key generation, one MAC hash computation and one symmetric encryption. The entire time cost is $T_g + T_{en} + T_h$. When receiving a message, the control center also generates a key and a MAC, and decrypts the ciphertext, such that the time cost is $T_g+T_{de}+T_h$. Besides, it has to search the Logistic Map parameters in database and generates time cost T_{wr} and T_{re}. In total, the time cost is $2T_g + 2T_h + T_{de} + T_{wr} + T_{re}$.

For the D2D communication, the sender first connect to control center which conducts two request to assign the temporal session key. This process consists of three message transmissions with time cost $3(2T_g + 2T_h + T_{de} + T_{wr} + T_{re})$. Then, the sender and receiver devices transferring ρ messages. Therefore, the total time cost is $3(2T_g + 2T_h + T_{de} + T_{wr} + T_{re}) + \rho(2T_g + 2T_h + T_{de})$.

Considering the public key scheme generates more than 100 times consumption, our protocol extremely saves the computation resources of such resource-restraint IoT devices. Moreover, if control center implements cache scheme, the entire time cost is significantly decreased since the I/O cost is much larger than that of symmetric encryption [24].

2) MEMORY USAGE

In our protocol, each device maintains a parameter μ and a initial value x_i, both in float type with 32 bits. A symmetric key is 128 bits and a MAC is 256 bits. Besides, it may receive or send a π bits message (AES-128 algorithm outputs a ciphertext with the same length). In total, memory usage of a device is $448 + \pi$ bits. The control center has the same memory usage for each device. Assuming there are γ devices connect to the network, entire memory usage of control center is about $\gamma(448 + \pi)$ bits.

Additionally, both the devices and control center maintain a list of identifiers of reachable devices. Assume an identifier is 32 bits and all the devices are mutually reachable, the list is 32γ bits. We stress that, as devices have no need to connect to all the others, access strategies can be used to reduce the size of their lists according to the resource usage. For example, the control center can only assign the identifier of thermometer to an air conditioner, or send only the identifier of motion sensor to a door locker.

It is noticeable that above memory usage values are benchmarks since all those devices and control center may communicate with $n_{con} \leqslant \gamma$ other devices.

D. COMPARISON ANALYSIS

Our proposed protocol is partially inspired by the previous approach developed by Song *et al.* [1]. They introduced a chaotic system-based symmetric protocol for Smart Home System (SHS). In that work, they solely concern about the D2C communication. The process of such protocol is briefly presented as follows.

1) In their initialization phase, control center selects two pairs of parameters and initial values to setup two chaotic maps. One is used to generate encryption key and another for computing MAC. Then, the two pairs are transferred to a specific device through an unreliable channel. Since the adversary of SHS environment is hard to intercept the message, they suppose such transmission is secure.
2) In their D2C communication phase, devices periodically connect to control center and each connection generates two keys based on above two chaotic maps, respectively. Then, plaintext message is encrypted using one key and MAC is computed based on another. The control center also generates two keys for decryption and authentication. In their case, ***as each device maintains two distinct chaotic maps, their memory and computation consumption is about twice of our protocol.***
3) When a device accomplishes the encryption and MAC computation, it updates keys and sends the ciphertext to control center. Afterwards, when control center accepts such message, it updates the corresponding keys also. This is a asynchronous rekey scheme. As mentioned before, such scheme vulnerable to key reset attack in which an adversary can intercepts the message sent to control center and resulting in inconsistent keys between the two endpoints. Even adding a verifying message from control center, one can still intercept this message and cause the device canceling key update. Even worse, ***their initialization is absolutely unreliable, which makes the reset keys exposed to the adversary.***

Exactly, the above vulnerabilities of their work can be extremely mitigated by the simple SHS environment. However, this significantly restrains the adoption of their protocol. In contrast, ***our protocol is generic for heterogeneous IoT environment containing powerful interceptors***. Moreover, we also present a D2D communication scheme rather than only the D2C one. In summary, our approach is more lightweight and effective than that in [1].

V. DISCUSSION

Application. Our protocol is suitable for any IoT systems that apply a central architecture. For example, ECUs implemented in ICVs can be used as control center in our protocol. Symmetric cryptography helps to minimize power cost of the sensors so that can prolong service life of car battery, especially facilitates the electric vehicles [35]. Besides, smart home systems and wearables are also require a robust lightweight protocol to reduce resource consumption of power-restraint devices such as wireless camera, intelligent door lock and smartbands.

Limitations. Our protocol should works with upon some other techniques like key storage and access control to ensure the entire security of IoT environment. On one hand, when dealing with a large number of devices, the control center should request a database storing large scale

device information. Also, a cache mechanism may also be required to improve the performance. For instance, Kumar *et al.* [9] introduced an effective hierarchical key management scheme to maintain the keys used in large scale environment and Yin and Liu [36] developed an effective data indexing mechanism. Such data or key management methods can facilitates our approach.

On another hand, a comprehensive access control scheme is helpful to reduce the memory used for the reachable identifier list, i.e., it can define what kind of resource (collected by some types of devices) should be accessed by a certain device. Under such consideration, most devices will only maintain a small number of identifiers of the collaborative devices. Therefore, the receiver of a connection request can determine whether deny it or not.

Though our protocol relies on such techniques, we stress that our purpose is to design a lightweight secure communication protocol, and those techniques are out of the scope of this paper and left in our future work.

VI. RELATED WORK

In this section, we summary the related work presented in recent years for IoT environment and category them into three types, i.e., key delegation and authentication, access control, and secure communication.

A. KEY DELEGATION AND AUTHENTICATION

Traditional key delegation mechanisms like [11]–[16], [18] mainly focused on designing a comprehensive distribution algorithm to maintain the connectivity of entire network. Such algorithms are too complex to be deployed in real world IoT environment.

In recent, several IoT-oriented approaches have been introduced. Ambrosin *et al.* [37] proposed SCIoT, a Secure and sCalable framework for IoT management. SCIoT guarantees low complexity in terms of communication, storage and computation on both managed devices and the management entity. Thomas *et al.* [38] presented a new Key Management and Distribution Scheme for use in the European Rail Traffic Management System (ERTMS). Its aim is to simplify key management and improve cross-border operations through hierarchical partitioning. Kumar *et al.* [9] proposes JEDI (Joining Encryption and Delegation for IoT), a many-to-many device-to-device encryption protocol for IoT. JEDI encrypts and signs messages device-to-device, while conforming to the decou- pled communication model typical of IoT systems. Wang *et al.* [39] proposed an approach to utilize the power of distributed caching and explores the feasibility of using the cache spaces on all IoT devices as a large pool to store validated certificates. Roeschlin *et al.* [40] proposed a novel approach to device pairing that applies whenever a user wants to pair two devices that can be physically touched at the same time. Han *et al.* [41] developed a new context-based pairing mechanism called Perceptio that uses time as the common factor across differing sensor types.

B. ACCESS CONTROL

Etigowni *et al.* [42] presented CPAC, a cyber-physical access control solution to manage complexity and mitigate threats in cyber-physical environments, with a focus on the electrical smart grid. Zhou *et al.* [43] conducted an in-depth analysis of devices interaction on five widely-used smart home platforms to illustrate the vulnerabilities caused by anomaly state transitions. Celik *et al.* [44] presented IOTGUARD, a dynamic, policy-based enforcement system for IoT, which protects users from unsafe and insecure device states by monitoring the behavior of IoT and trigger- action platform apps. Zhou *et al.* [45] proposed Heracles, an IoT access control system that achieves robust, fine-grained access control at enterprise scale. Heracles adopts a capability-based approach using secure, unforgeable tokens that describe the authorizations of subjects, to either individual or collections of objects in single or bulk operations. Ding and Hu [46] proposed a framework called IoTMon that discovers any possible physical interactions and generates all potential interaction chains across applications in the IoT environment. IoTMon also includes an as- sessment of the safety risk of each discovered inter-app interaction chain based on its physical influence. He *et al.* [47] reenvisioned access control and authentication for the home IoT. They proposed that access control focus on IoT capabilities (i.e., certain actions that devices can perform), rather than on a per-device granularity. Schuster *et al.* [48] designed and implemented a novel approach to IoT access control. Our key innovation is to introduce ''environmental situation oracles'' (ESOs) as first-class objects in the IoT ecosystem.

C. SECURE COMMUNICATION

Yang *et al.* [49] proposed two multi-cloud-based outsourced-ABE schemes, namely the parallel-cloud ABE and the chain-cloud ABE, which enable the receivers to partially outsource the computationally expensive decryption operations to the clouds, while preventing user attributes from being disclosed. Shi *et al.* [50] introduced an ultra-lightweight white-box encryption scheme, which requires a relatively small amount of static data, for securing resource-constrained IoT devices. Song *et al.* [19] presented a D2S secure communication protocol for SHS environment based on symmetric key scheme, in which the key is updated using a one-way hash function to maintain the freshness. In the next year, they improved their method using chaotic system in [1]. Li *et al.* [51] identified the design requirements of a smart shopping system, built a prototype system to test functionality, and designed a secure communication protocol to make the system practical. Yin *et al.* [52] present a lightweight intrusion detection approach to identify the compromised host for secure communication. Zhu *et al.* [53]–[56] presented some techniques to improve or protect the communication between sensors and clouds.

Summary. Most of the previous works target on specific IoT environment, e.g., smart home, intelligent city or power

grid, which restrains the adoption and feasibility in real world deployment. Differing from them, our work presents a generic and ultra-lightweight secure communication protocol for heterogeneous IoT environment under the existence of a powerful adversary. Moreover, such key delegation and access control techniques as well as some optimization approaches like [57], [58] are compatible with our protocol as discussed in Section V.

VII. CONCLUSION

In this paper, we present an ultra-lightweight device-to-device secure protocol solely based on the symmetric key-based scheme. Our protocol provides generic protection for heterogeneous IoT environment. In this protocol, the synchronous key delegation mechanism is designed using a chaotic system, i.e., Logistic Map, which ensures the unpredictable, unrepeatable and determinate properties of the symmetric keys. We comprehensively evaluate the security and efficacy of the proposed protocol, and examine the resistance against some harmful vulnerabilities. The result shows that our protocol outperforms the previous symmetric key-based work for smart home systems.

REFERENCES

[1] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for IoT applications in smart homes," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1844–1852, Dec. 2017, doi: 10.1109/JIOT.2017.2707489.

[2] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Trans Ind. Informat.*, vol. 16, no. 4, pp. 2659–2666, Apr. 2020.

[3] Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: A novel reputation framework for identifying denial of traffic service in Internet of connected vehicles," *IEEE Internet Things J.*, to be published.

[4] *Sumsung Smartthings Developers Documentation*. [Online]. Available: https://smartthings.developer.samsung.com/blog/en-us/2019/01/17/Shape-the-Future-of-IoT-with-SmartThings

[5] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "VulCAN: Efficient component authentication and software isolation for automotive control networks," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Orlando, FL, USA, Dec. 2017, pp. 225–237, doi: 10.1145/3134600.3134623.

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, doi: 10.1145/359340.359342.

[7] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976, doi: 10.1109/TIT.1976.1055638.

[8] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Elliptic curve lightweight cryptography: A survey," *IEEE Access*, vol. 6, pp. 72514–72550, 2018, doi: 10.1109/ACCESS.2018.2881444.

[9] S. Kumar, Y. Hu, M. P. Andersen, R. A. Popa, and D. E. Culler, "JEDI: Many-to-many end-to-end encryption and key delegation for IoT," in *Proc. 28th USENIX Secur. Symp., USENIX Secur.*, Santa Clara, CA, USA, Aug. 2019, pp. 1519–1536. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/kumar-sam

[10] A. K. Das, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for Internet of Things," *Future Gener. Comput. Syst.*, vol. 89, pp. 110–125, Dec. 2018, doi: 10.1016/j.future.2018.06.027.

[11] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. Comput. Commun. Secur. (CCS)*, Washington, DC, USA, Nov. 2002, pp. 41–47, doi: 10.1145/586110.586117.

[12] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. Symp. Secur. Privacy*, Berkeley, CA, USA, May 2003, pp. 197–213, doi: 10.1109/SECPRI.2003.1199337.

[13] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proc. 12th Annu. Int. Cryptol. Conf., Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, Aug. 1992, pp. 471–486, doi: 10.1007/3-540-48071-4_33.

[14] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 41–77, Feb. 2005, doi: 10.1145/1053283.1053287.

[15] A. K. Das, "A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks," *Int. J. Inf. Secur.*, vol. 11, no. 3, pp. 189–211, Apr. 2012, doi: 10.1007/s10207-012-0162-9.

[16] F. Hendaoui, H. Eltaief, and H. Youssef, "A collaborative key management scheme for distributed smart objects," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 6, Jun. 2018, Art. no. e3198, doi: 10.1002/ett.3198.

[17] A. K. Das, "ECPKS: An improved location-aware key management scheme in static sensor networks," *Int. J. Netw. Secur.*, vol. 7, no. 3, pp. 358–369, 2008. [Online]. Available: http://ijns.femto.com.tw/contents/ijns-v7-n3/ijns-2008-v7-n3-p358-369.pdf

[18] I.-C. Tsai, C.-M. Yu, H. Yokota, and S.-Y. Kuo, "Key management in Internet of Things via kronecker product," in *Proc. IEEE 22nd Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Christchurch, South Island, Jan. 2017, pp. 118–124, doi: 10.1109/PRDC.2017.25.

[19] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for IoT applications in smart homes," in *Proc. Int. Conf. Identificat., Inf. Knowl. Internet Things (IIKI)*, Beijing, China, Oct. 2016, pp. 519–524, doi: 10.1109/IIKI.2016.3.

[20] W. S. Sayed, A. G. Radwan, and H. A. H. Fahmy, "Design of a generalized bidirectional tent map suitable for encryption applications," in *Proc. 11th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2015, pp. 207–211.

[21] T. S. Chaware and B. K. Mishra, "Secure communication using TPC and chaotic encryption," in *Proc. Int. Conf. Inf. Process. (ICIP)*, Dec. 2015, pp. 615–620.

[22] P. Tobin, L. Tobin, M. McKeever, and J. Blackledge, "Chaos-based cryptography for cloud computing," in *Proc. 27th Irish Signals Syst. Conf. (ISSC)*, Jun. 2016, pp. 1–6.

[23] C. Hu, A. Alhothaily, A. Alrawais, X. Cheng, C. Sturtivant, and H. Liu, "A secure and verifiable outsourcing scheme for matrix inverse computation," in *Proc. IEEE IEEE Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, May 2017, pp. 1–9, doi: 10.1109/INFOCOM.2017.8057199.

[24] S. A. Hirani, "Energy consumption of encryption schemes in wireless devices," Ph.D. dissertation, Univ. Pittsburgh, Pittsburgh, PA, USA, 2003.

[25] W. Liao, C. Luo, S. Salinas, and P. Li, "Efficient secure outsourcing of large-scale convex separable programming for big data," *IEEE Trans. Big Data*, vol. 5, no. 3, pp. 368–378, Sep. 2019, doi: 10.1109/TBDATA.2017.2787198.

[26] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983, doi: 10.1109/TIT.1983.1056650.

[27] C. J. F. Cremers, S. Mauw, and E. P. de Vink, "Injective synchronisation: An extension of the authentication hierarchy," *Theor. Comput. Sci.*, vol. 367, nos. 1–2, pp. 139–161, Nov. 2006, doi: 10.1016/j.tcs.2006.08.034.

[28] S. Meier, C. Cremers, and D. Basin, "Strong invariants for the efficient construction of machine-checked protocol security proofs," in *Proc. 23rd IEEE Comput. Secur. Found. Symp.*, Edinburgh, U.K., Jul. 2010, pp. 231–245, doi: 10.1109/CSF.2010.23.

[29] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard* (Information Security and Cryptography). Springer, 2002, doi: 10.1007/978-3-662-04722-4.

[30] D. Eastlake, III, and P. E. Jones, *US Secure Hash Algorithm 1 (SHA1)*, document RFC 3174, 2001, pp. 1–22, doi: 10.17487/RFC3174.

[31] D. Eastlake, III, and T. Hansen, *US Secure Hash Algorithms (SHA and HMAC-SHA)*, document RFC 4634, 2006, pp. 1–108, doi: 10.17487/RFC4634.

[32] D. Eastlake, III, and T. Hansen, *US Secure Hash Algorithms (SHA and SHA-Based HMAC and HKDF)*, document RFC 6234, 2011, pp. 1–127, doi: 10.17487/RFC6234.

[33] J. Balasch, B. Ege, T. Eisenbarth, B. Gérard, Z. Gong, T. Güneysu, S. Heyse, S. Kerckhof, F. Koeune, T. Plos, T. Pöppelmann, F. Regazzoni, F. Standaert, G. V. Assche, R. V. Keer, L. van Oldeneel tot Oldenzeel, and I. von Maurich, "Compact implementation and performance evaluation of hash functions in attiny devices," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, Graz, Austria, Nov. 2012, pp. 158–172, doi: 10.1007/978-3-642-37288-9_11.

[34] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Dallas, TX, USA, Oct./Nov. 2017, pp. 1313–1328, doi: 10.1145/3133956.3134027.

[35] Y. Cao, T. Jiang, O. Kaiwartya, H. Sun, H. Zhou, and R. Wang, "Toward pre-empted EV charging recommendation through V2V-based reservation system," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published.

[36] L.-H. Yin and H. Liu, "Searching activity trajectories with semantics," *J. Comput. Sci. Technol.*, vol. 34, no. 4, pp. 775–794, Jul. 2019.

[37] M. Ambrosin, M. Conti, A. Ibrahim, A. Sadeghi, and M. Schunter, "SCIoT: A secure and scalable end-to-end management framework for IoT devices," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)*, Barcelona, Spain, Sep. 2018, pp. 595–617, doi: 10.1007/978-3-319-99073-6_29.

[38] R. J. Thomas, M. Ordean, T. Chothia, and J. de Ruiter, "TRAKS: A universal key management scheme for ERTMS," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Orlando, FL, USA, Dec. 2017, pp. 327–338, doi: 10.1145/3134600.3134631.

[39] M. Wang, C. Qian, X. Li, and S. Shi, "Collaborative validation of public-key certificates for IoT by distributed caching," in *Proc. IEEE IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr./May 2019, pp. 847–855, doi: 10.1109/INFOCOM.2019.8737423.

[40] M. Roeschlin, I. Martinovic, and K. B. Rasmussen, "Device pairing at the touch of an electrode," in *Proc. 25th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, Feb. 2018, pp. 1–15. [Online]. Available: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03B-4_Roeschlin_paper.pdf

[41] J. Han, A. J. Chung, M. K. Sinha, M. Harishankar, S. Pan, H. Y. Noh, P. Zhang, and P. Tague, "Do you feel what I hear? Enabling autonomous IoT device pairing using different sensor types," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2018, pp. 836–852, doi: 10.1109/SP.2018.00041.

[42] S. Etigowni, D. J. Tian, G. Hernandez, S. Zonouz, and K. Butler, "CPAC: Securing critical infrastructure with cyber-physical access control," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl. (ACSAC)*, Los Angeles, CA, USA, Dec. 2016, pp. 139–152. [Online]. Available: http://dl.acm.org/citation.cfm?id=2991126

[43] W. Zhou, Y. Jia, Y. Yao, L. Zhu, L. Guan, Y. Mao, P. Liu, and Y. Zhang, "Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms," in *Proc. 28th USENIX Secur. Symp., USENIX Secur.*, Santa Clara, CA, USA, Aug. 2019, pp. 1133–1150. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/zhou

[44] Z. B. Celik, G. Tan, and P. McDaniel, "IoTGuard: Dynamic enforcement of security and safety policy in commodity IoT," in *Proc. Netw. Distrib. Syst. Secur. Symp. NDSS*, San Diego, CA, USA, Feb. 2019, pp. 1–15.

[45] Q. Zhou, M. Elbadry, F. Ye, and Y. Yang, "Heracles: Scalable, fine-grained access control for Internet-of-Things in enterprise environments," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1772–1780, doi: 10.1109/INFOCOM.2018.8485944.

[46] W. Ding and H. Hu, "On the safety of IoT device physical interaction control," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 832–846, doi: 10.1145/3243734.3243865.

[47] W. He, M. Golla, R. Padhi, J. Ofek, M. Dürmuth, E. Fernandes, and B. Ur, "Rethinking access control and authentication for the home Internet of Things (IoT)," in *Proc. 27th USENIX Secur. Symp., USENIX Secur.*, Baltimore, MD, USA, Aug. 2018, pp. 255–272. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/he

[48] R. Schuster, V. Shmatikov, and E. Tromer, "Situational access control in the Internet of Things," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 1056–1073, doi: 10.1145/3243734.3243817.

[49] L. Yang, A. Humayed, and F. Li, "A multi-cloud based privacy-preserving data publishing scheme for the Internet of Things," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl. (ACSAC)*, Los Angeles, CA, USA, Dec. 2016, pp. 30–39. [Online]. Available: http://dl.acm.org/citation.cfm?id=2991127

[50] Y. Shi, W. Wei, Z. He, and H. Fan, "An ultra-lightweight white-box encryption scheme for securing resource-constrained IoT devices," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl. (ACSAC)*, Los Angeles, CA, USA, Dec. 2016, pp. 16–29. [Online]. Available: http://dl.acm.org/citation.cfm?id=2991086

[51] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "IoT applications on secure smart shopping system," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1945–1954, Dec. 2017, doi: 10.1109/JIOT.2017.2706698.

[52] L. Yin, X. Luo, C. Zhu, L. Wang, Z. Xu, and H. Lu, "ConnSpoiler: Disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries," *IEEE Trans Ind. Informat.*, vol. 16, no. 2, pp. 1373–1384, Feb. 2020.

[53] C. Zhu, X. Li, V. C. M. Leung, L. T. Yang, E. C.-H. Ngai, and L. Shu, "Towards pricing for sensor-cloud," *IEEE Trans. Cloud Comput.*, to be published.

[54] C. Zhu, V. C. M. Leung, K. Wang, L. T. Yang, and Y. Zhang, "Multi-method data delivery for green sensor-cloud," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 176–182, May 2017.

[55] C. Zhu, V. C. M. Leung, J. J. P. C. Rodrigues, L. Shu, L. Wang, and H. Zhou, "Social sensor cloud: Framework, greenness, issues, and outlook," *IEEE Netw.*, vol. 32, no. 5, pp. 100–105, Sep. 2018.

[56] C. Zhu, L. Shu, V. C. M. Leung, S. Guo, Y. Zhang, and L. T. Yang, "Secure multimedia big data in trust-assisted sensor-cloud for smart city," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 24–30, Dec. 2017.

[57] H. Zhou, X. Chen, S. He, J. Chen, and J. Wu, "DRAIM: A novel delayconstraint and reverse auction-based incentive mechanism for WiFi offloading," *IEEE J. Sel. Areas Commun.*, to be published.

[58] X. Li, C. Luo, H. Ji, Y. Zhuang, H. Zhang, and V. C. M. Leung, "Energy consumption optimization for self-powered IoT networks with non-orthogonal multiple access," *Int. J. Commun. Syst.*, vol. 33, no. 1, Sep. 2019, Art. no. e4174, doi: 10.1002/dac.4174.

XI LUO received the Ph.D. degree in computer science and technology from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. He is currently conducting his Postdoctoral Research at the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China. His current research interests include network security and the IoT security.

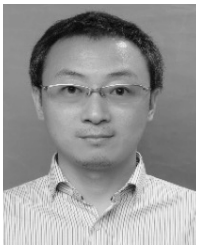
LIHUA YIN received the Ph.D. degree from the Harbin Institute of Technology. She is currently a Professor and also the Associate Dean of the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong, China. She has authored over 100 articles published by refereed international conferences and journals. Her research interests include computer networks, the Internet of Things, and cyberspace security. Her research has been supported in part by the National Natural Science Foundation of China, the National Key research and Development Plan of China, the Major State Basic Research Development Program of China (973 Program), and the National High-tech Research and Development Program of China (863 Program). She has also served as a member of the China Computer Federation.

CHAO LI received the B.S. and M.S. degrees in computer science and technology from the Guilin University of Electronic Technology, Guilin, China, in 2007, and the Ph.D. degree in information security from the University of Chinese Academy of Sciences, Beijing, China, in 2013. From 2013 to 2017, he was a Research Assistant with the Institute of Information Engineering, Chinese Academy of Sciences. He is currently an Associate Professor with the Cyberspace Institute of Advanced Technology, Guangzhou University. His current research interests include privacy preserving and access control in the Internet of Things, and information leakage protecting.

CHONGHUA WANG received the Ph.D. degree from the Institute of information Engineering, Chinese Academy of Sciences. He was a visiting joint Ph.D. student with Purdue University. He is currently a Researcher with the China Industrial Control Systems Cyber Emergency Response Team. His research interests include attack and defense technology of network and systems, cloud security, the Industrial Internet Security, and ICS security.

FUYANG FANG received the Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences. He is currently a Researcher with the Information Science Academy, China Electronics Technology Group Corporation. His research interests include cryptography, Blockchain, cyberspace situational awareness, artificial intelligence security, and the Industrial Internet Security.

CHUNSHENG ZHU received the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Canada. He is currently an Associate Professor with the SUSTech Institute of Future Networks, Southern University of Science and Technology, China. He is also an Associate Researcher with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, China. He has authored more than 100 publications published by refereed international journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the *ACM Transactions on Embedded Computing Systems*, and the *ACM Transactions on Cyber-Physical Systems*, and the magazines such as the *IEEE Communications Magazine*, the *IEEE Wireless Communications Magazine*, and the *IEEE Network Magazine*, as well as conferences such as the IEEE INFOCOM, IEEE IECON, IEEE SECON, IEEE DCOSS, IEEE ICC, and IEEE GLOBECOM. His research interests mainly include the Internet of Things, wireless sensor networks, cloud computing, big data, social networks, and security.



ZHIHONG TIAN (Member, IEEE) is currently a Professor and the Dean of the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong, China. He is a Guangdong Province Universities and Colleges Pearl River Scholar (Distinguished Professor). He is also a part-time Professor at Carlton University, Ottawa, Canada. He has served in different academic and administrative positions at the Harbin Institute of Technology. He has authored over 200 journal and conference papers in these areas. His research interests include computer networks and cyberspace security. His research has been supported in part by the National Natural Science Foundation of China, the National Key research and Development Plan of China, and the National High-tech Research and Development Program of China (863 Program). He has also served as a member, Chair, and General Chair for a number of international conferences. He is a Senior Member of the China Computer Federation.