Draft

Technical Report TR2: ISIF ASIA Funded Project

Lightweight Authentication Mechanism and OAuth Protocol for IoT Devices

Prof Vijay Varadharajan, Dr Uday Tupakula and Kallol Karmakar

Advanced Cyber Security Engineering Research Centre (ACSRC)
Faculty of Engineering and Built Environment
The University of Newcastle

Oct 2018

Acknowledgements:

The authors would like to thank ISIF ASIA for their financial contribution to the Project

Executive Summary

The report presents project summary on the “Software Defined Networks based Security Architecture for IoT Infrastructures” project funded by the ISIF group. There are three milestones for the project with specific deliverables for each milestone. As part of first milestone, we have conducted detailed survey of attacks related to IoT and proposed the design and development of feature distributed malware attacks for IoT. We have also analysed previously proposed techniques to deal with the attacks and developed some security requirements that need to be considered for designing and developing security architecture for IoT Applications. Please refer to Milestone 1 report for more details on the outcomes. In this report we present a summary of the outcome of second milestone “Lightweight Authentication Mechanism and OAuth Protocol for IoT Devices” which consists of the following tasks: i) Design of the Lightweight Authentication Protocol for IoT and ii) Security Analysis of the Authentication Protocol for IoT.

The Internet of Things (IoT) provides transparent and seamless incorporation of heterogeneous and different end systems. It has been widely used in many applications including smart cities such as public water system, power grid, water management, and vehicle traffic control system. In these smart city applications, a large number of IoT devices are deployed that can sense, communicate, compute, and potentially actuate. The uninterrupted and accurate functioning of these devices are critical to smart city applications as crucial decisions will be made based on the data received. For example, a traffic monitoring system utilizes received information to make decisions, i.e., control the traffic. However, such systems are vulnerable to cyber-attacks as the millions of sensors deployed in an unprotected space introduces the greater attack space, and are beyond the control of traditional perimeter defence mechanisms. A malicious user, who intends to create chaos on the road, can take over certain sensors and may report false data to the system. One of the challenging tasks is to assure the authenticity of the devices so that we can rely on the decision making process with a very high confidence. One of the characteristics of IoT devices deployed in such applications is that they have limited battery power. A challenge is to design a secure mutual authentication protocol, which is affordable to resource constrained devices.

As part of this milestone, we developed a lightweight mutual authentication protocol based on a novel public key encryption scheme for smart city applications. The proposed protocol takes a balance between the efficiency and communication cost without sacrificing the security. We evaluate the performance of our protocol in software and hardware environments. On the same security level, our protocol performance is significantly better than existing RSA and ECC based protocols. We also provide security analysis of the proposed encryption scheme and the mutual authentication protocol. The proposed protocol is an n-pass lightweight mutual authentication protocol. The value of n is related to the desired security level of the protocol and the system parameters of the encryption scheme. Our lightweight mutual authentication protocol applies the proposed encryption scheme as a building block. The security of the proposed n-pass mutual authentication is guaranteed by the security of the Needham-Schroeder protocol. We will show that our protocol can resists attacks such as man-in-the-middle attack and impersonation attack. We evaluated the protocol on Contiki OS and CC2538 evaluation modules. The experimental evaluations show that our protocol is respectively 88 and 7 times faster than RSA and ECC on the security level of 112 bits. The mutual authentication time can be further reduced if online/offline technique is enabled.

As part of Milestone 3, we are working on the development of a security architecture for IoT networks by leveraging the underlying features supported by Software Defined Networks (SDN). We aim to use Lightweight Elliptic Curve Cryptography (ECC) to achieve authentication and OAuth Protocol for authorisation in our security architecture. A more detailed report on the security architecture will be provided as part of Milestone 3 deliverable

The authors would like to thank ISIF ASIA for their financial contribution to the Project.

Contents

1. Introduction

The report presents project summary on the “Software Defined Networks based Security Architecture for IoT Infrastructures” project funded by the ISIF group. There are three milestones for the project with specific deliverables for each milestone. In this report we present a summary of the outcome of second milestone. The aims of this milestone are to develop Lightweight Authentication Mechanism and OAuth Protocol for IoT Devices.

The Internet of Things (IoT) comprises of billions of devices that can sense, communicate, compute and potentially actuate. In IoT, a physical entity like a sensor is a piece of a network and it is given abilities to gather and exchange data with other entities. With diverse information collected from devices, the IoT can be used to implement many important systems such as public water management system, power grid, water management, and vehicle traffic control system. These systems improve the understanding, controlling and interacting with our environment. The sensed data also help to drive further innovation of novel applications. Our focus in this report is on the smart city applications [1].

The Smart City [2] provides a vision of urban development. It collects a variety of information from different sources; for example, environmental information such as pollution types, and noise level or transport information such as parking vacancy, peak hour traffic and live traffic report. The authorities then can use the obtained information for urban planning. Also, people can access public data to understand their living environment. As an example, a pollution monitoring system provides the pollution type and level in a particular area. The information allows people to carry out protections like wearing a breathing mask. It is more important to people who have specific medical diseases. Similarly, the transport management system controls the traffic lights based on the traffic flows at a particular time.

In IoT applications, the things (sensors, tags, etc.) can collect sensitive information from environment and provide the information to authorized users or applications. The information is then analysed and decisions are made to take certain actions including actuating sensors. The decisions will have serious consequences. For example, a traffic monitoring system utilizes received information to make decisions, i.e., control the traffic. Such systems are vulnerable to cyber-attacks as the millions of sensors deployed in an unprotected space introduces the greater attack space, and are beyond the control of traditional perimeter defence mechanisms. A malicious user, who intends to create chaos on the road, can take over certain sensors and may report false data to the system. It is this important that such systems satisfy at least the following properties: confidentiality and integrity of data and devices, authentication of devices, and access control to data. In this report, our focus is on the authentication of devices.

There could be potential attacks on the IoT systems either by taking over the sensors or the server. That is, a malicious attacker may take over a sensor and report the false data or the attacker may take over the server and wrongly actuate the sensor. It is thus important to support mutual authentication to the Smart City applications. This means both sensors and a server also should be authenticated prior to data exchange. In addition, mutual authentication protocol is desired to be lightweight due to the resource constrained IoT devices. In the following, we first provide a brief summary of existing authentication protocol for IoT systems and their shortcomings. It is then followed by a brief summary of our proposed technique.

Algebraic Eraser ™ (AE) [3] is a concept introduced by SecureRF Corporation. The goal is to develop lightweight public key based algorithms and protocols which can be used in Radio Frequency Identification (RFID), Near Field Communication (NFC), IoT, etc. The proposed key agreement protocol [3] is a public key based cryptosystem using braid groups [4]. AE is claimed to achieve high performance on lightweight devices [5]. During the system setup phase, a Trusted Third Party (TTP) is needed to generate system parameters. In other words, TTP controls the master secret so that it should be fully trusted.

NTRU [6] is a lattice based public key cryptosystem proposed in 1996. It has recently received more attentions because of the high security level, moderate key size and asymptotic performance. Unlike conventional public key schemes, such as RSA and ElGamal, NTRU is based on finding small solutions to a system of linear equations over rings. NTRU encryption scheme can be used on lightweight devices. The security and performance are better than the existing RSA and ECC based solutions. However, NTRU requires larger key size and ciphertext size. The communication cost is an issue of NTRU in low power networks.

Elliptic Curve Cryptography (ECC) is an alternative approach to Public Key Cryptography (PKC). An advantage of ECC-based cryptographic algorithms is the short key size. Hence, they are inherently suitable for IoT applications. For example, an ECC encryption algorithm with 160-bit key size can achieve the same security level to that of 1,024-bit RSA encryption [7]. Many ECC based authentication protocols [8], [9], [10], [11] were proposed in RFID research. Note that RFID is an important element of IoT. In practice, Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie-Hellman (ECDH) key exchange are two standard ECC schemes. They can be used in lightweight mutual authentication protocols and provide high level security. Comparing with some solutions like AE and NTRU, ECC based protocols are inefficient because scalar point multiplication is heavy to lightweight devices.

In IoT applications, a lightweight device interacts with either other lightweight devices or a powerful server. The communication capabilities like bandwidth restrict low-cost devices to transmit large messages. High efficiency is another requirement to lightweight protocols. The computational cost to a lightweight device should be affordable and a protocol run shall be performed efficiently. It is a reason that many traditional authentication protocols cannot be used on IoT devices [12]. Existing solutions based on AE, ECC and NTRU have their advantages and disadvantages. In this report, the primary objective is to propose a lightweight mutual authentication protocol which provides a balanced result on security and performance. Note that the solution should not sacrifice the security to achieve high performance.

The contribution of this report is two-folds: First, we propose a new public key encryption scheme and use it to construct a lightweight mutual authentication protocol. The proposed scheme and protocol are suitable to lightweight devices and low power networks that are deployed in Smart City applications. Our encryption scheme does not require a large number of computations and it allows lightweight devices to perform calculations efficiently. The proposed mutual authentication protocol takes a balance among the efficiency, ciphertext size and usability, in comparison with ECC, AE and NTRU. As demonstrated in Section 4.3, our protocol takes about 1,401 ms while ECC based protocol needs 6,160 ms at the 80 bits security level. A ciphertext of our protocol is in size of 272 bits which is 897 bits shorter than NTRU at the message security level of 80 bits [6]. Compared with AE, our scheme does not need TTP during IoT device setup period, making our scheme more practical.

Second, we analyze the security of proposed scheme and protocol, then implement a prototype in Contiki platform. This prototype is evaluated in various environments including Cooja based emulation

environment and Texas Instruments CC2538 hardware platform. With correct configurations, IoT nodes can be mutually authenticated efficiently; in particular, it takes around 125ms on CC2538. In addition, we show some optimizations to the scheme and tested the message retransmission rate which can be used to understand the underlying network behaviours.

The remainder of this report is organized as follows. Chapter 2 reviews some related work. In Chapter 3, we first present definitions, notifications of symbols, security models, adversaries and attacks. Then we propose our lightweight mutual authentication protocol and a novel public key encryption scheme and present the security analysis of the proposed schemes. Chapter 4 shows the implementation and emulation results of our protocol and it compares the performance with other solutions. Chapter 5 describes the current work on the development of SDN based security architecture for IoT, which makes use of ECC for authentication, and OAuth for Authorisation of network services. Chapter 6 concludes the report.

2. Related Work

In this chapter Although IoT has huge potential, such an IoT environment with heterogeneous devices with different operating systems and connectivity ranging from Zigbee to wireless to mobile networks poses significant challenges in security and privacy.

Anshel et al. [3] proposed a lightweight key agreement protocol (AAGL for short) based on AE. The protocol achieves high performance which is better than existing public key based solutions to low-cost platforms. AE introduces a new operation called E-multiplication. It is a one-way function that an adversary cannot reveal the input from a given output. The complexity of E-multiplication increases linearly with the security level. It allows AAGL protocol to significantly improve the efficiency. The underlining mathematical foundations of AE is different from that of traditional public key cryptosystems. AAGL protocol utilizes the braid group and its properties to construct an algebraic erase concept. Braid group was introduced to public key cryptosystem by Ko, Lee, Cheon, Han, Kang, and Park [13]. Some braid group based protocols and algorithms (e.g., [14], [15], [16]) were proposed later. However, potential problems in braid groups were found [17], [18]. Note that AE is claimed not braid group cryptography. Nevertheless, some potential attacks against AE were proposed recently in [19], [20], [21]. The defense of these attacks were provided in [22], [23], while the security of AE is still debated. In addition, AE based protocols need a TTP to generate system parameters. Furthermore, no public implementation is available for AE.

Hoffstein, Pipher and Silverman [6] proposed a public-key encryption scheme called NTRU encryption. The scheme is a mixing system which is based on elementary probability theory and polynomial algebra. In recent years, the NTRU encryption has received more attention because it can be a choice of post-quantum public-key encryption schemes [24]. This scheme is also included in the IEEE P1363 standard [25]. NTRU provides features such as high speed and low memory requirements. On the same security level, NTRU can be 10-to-100 times faster than the conventional public-key encryption schemes. It takes about 100 microseconds on contemporary computing platforms [24]. The speed record of NTRU is achieved by implementing this scheme on a GPU using the CUDA platform [26]. NTRU can be used to lightweight devices. However, the large size of keys and that of ciphertext are issues to low power network communication. The security of NTRU encryption scheme is not formally proved, but so far there is no serious attacks found. A variant of NTRU encryption with formal security proof is proposed by Stehlé and Steinfeld [27]. The proposed scheme is not implemented and the performance is unknown to lightweight devices.

Peeters, Hermans and Fan [28] proposed a wide-strong private mutual RFID authentication protocol called IBIHOP. The proposed protocols can resist most of the identified passive and active attacks. The proposed protocol is provably secure against a wide-strong adversary [29] who knows the internal state of tags. The protocol provides tag untraceability and anonymity even if the tag is compromised. Side-channel attacks are not considered. The IBIHOP protocol requires at least 3 scalar multiplications to tag. The performance (on tag) is close to standard ECDSA based mutual authentication protocols. Recently, a mutual authentication protocol was introduced by Lee and Chien [30] . It is also claimed to be secure against strong adversaries and achieve the wide-strong privacy. Unfortunately, the proposed protocol requires many scalar multiplications, hash computations and point additions.

Hardware implementation can significantly improve the performance of ECC based protocols. Lee et al. introduced an architecture of an elliptic curve based security processor for RFID tags. The implementation is suitable to lightweight ECC based authentication algorithms. It shows the high performance over GF (2163). For an EC scalar multiplication, the proposed processor takes 243.17 ms with 15,365 gates and 78,544 cycles. They also demonstrated the performance of Schnorr

identification protocol using the proposed security processor. Note that the processor cannot be used in our work and it is beyond the scope of this report. Apart from the public key based protocols, many recent authentication protocols like [31] achieve high performance by using efficient symmetric key based cryptography.

3. Lightweight Mutual Authentication for IoT and Its Applications

The Internet of Things (IoT) provides transparent and seamless incorporation of heterogeneous and different end systems. It has been widely used in many applications including smart cities such as public water system, power grid, water management, and vehicle traffic control system. In these smart city applications, a large number of IoT devices are deployed that can sense, communicate, compute, and potentially actuate. The uninterrupted and accurate functioning of these devices are critical to smart city applications as crucial decisions will be made based on the data received. One of the challenging tasks is to assure the authenticity of the devices so that we can rely on the decision making process with a very high confidence. One of the characteristics of IoT devices deployed in such applications is that they have limited battery power. A challenge is to design a secure mutual authentication protocol which is affordable to resource constrained devices. In this Chapter, we propose a lightweight mutual authentication protocol based on a novel public key encryption scheme for smart city applications. The proposed protocol takes a balance between the efficiency and communication cost without sacrificing the security. We also provide security analysis of the proposed encryption scheme and the mutual authentication protocol.

3.1 Preliminaries

3.1.1 Smart City Applications

IoT enabled Smart City applications are based on diverse type of devices and technologies. Fig. 1 shows an example of Smart Citiy scenario where the system observes environmental information, such as air quality, noise level and traffic congestion, for urban planning and public services. For example, vehicle network based system (e.g., [32], [33]) is attractive in Smart Cities. A typical Smart City system consists of four components: the things (sensors, tags, etc.), intermediate nodes, cloud server and users.

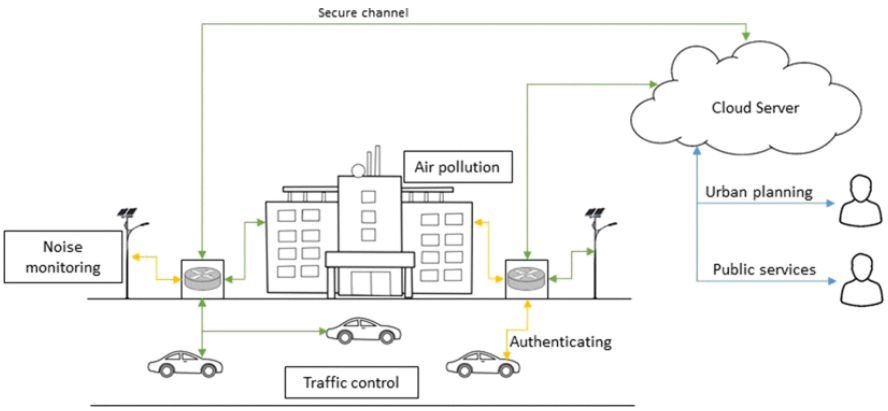

Figure 1: Smart City Applications

- The things: are resource constrained devices which report to cloud server the data collected from the environment. Note that the things need to be authenticated to establish a secure channel to the cloud server; otherwise the provided data cannot be trusted.
- Intermediate nodes: are powerful devices, such as routers and gateways, which connect between the cloud server and the things.
- Cloud server: receives data from the things and conducts data processing, and then provides services for authorities and the public. Note that the server must be authenticated to the things for preventing unauthorized users from observing sensitive information.
- Users: are people who use the services provided by the cloud server for understanding the environment and possibly making decisions.

3.1.2 Notations and Definitions

We provide the notations of symbols (Table 1) used in this report for references.

Table 1: Notations of Symbols

$\times$:	An integer multiplication.
$+$:	An addition of two integers or two vectors.
A, B:	Identities of two devices.
$N_{i,j}$:	The jth nonce of user i.
PK:	A user's public key.
SK:	A user's private key.
$\mathbf{pk}$:	A vector of a public key.
$pk_{i,j}$:	The jth element of the public key vector $\mathbf{pk}_i$.
$m_{\max}$:	The maximum length of a message.
C:	A ciphertext.
$E_{PK}(\cdot)$:	An encryption under the public key PK.
$D_{SK}(\cdot)$:	Ad decryption using the private key SK.
ℓ:	The bit length of a nonce.
n:	The number of passes of a protocol.
λ:	The security parameter of a system.

A public key encryption scheme consist of four algorithms: Setup, KeyGen, Encrypt and Decrypt:

- Setup: The setup algorithm takes a security parameter λ as an input. It outputs system parameters *params*.
- KeyGen: The key generation algorithm takes the system parameters *params* as an input. It outputs a pair of public and private keys (PK, SK).
- Encrypt: The encryption algorithm takes a message m and a public key PK as inputs. It outputs a ciphertext C.
- Decrypt: The decryption algorithm takes a ciphertext C and a private key SK as inputs. It outputs a message m.

A public key based mutual authentication protocol is composed of four algorithms: Setup, KeyGen, Init and Auth.

- Setup: On input a security parameter λ, it outputs system parameters *params*.
- KeyGen: On input system parameters *params*, it outputs a key pair (PK,SK).
- Init: On input identities and the corresponding public keys, the initialization algorithm sets the information on devices for a specific session. Note that this algorithm is optional if a session configuration is not needed.
- Auth: The authentication algorithm is an interactive algorithm runs between two users. The algorithm takes as input two identities A,B and the corresponding key pairs (PK_A,SK_A) and (PK_B,SK_B), respectively. It outputs 1 if two users are mutually authenticated, otherwise outputs 0.

Needham-Schroeder protocol was proposed by Needham and Schroeder [34]. They presented a public key based mutual authentication protocol. However, the protocol is insecure against the man-in-the-middle attack [35]. We review the fixed Needham-Schroeder protocol as follows.

A	$\rightarrow$	AS:	A, B
AS	$\rightarrow$	A:	$S_{SK_S}(PK_B, B)$
A	$\rightarrow$	B:	$E_{PK_B}(N_A, A)$
B	$\rightarrow$	AS:	B, A
AS	$\rightarrow$	B:	$S_{SK_S}(PK_A, A)$
B	$\rightarrow$	A:	$E_{PK_A}(N_A, N_B, B)$
A	$\rightarrow$	B:	$E_{PK_B}(N_B)$

AS denotes an authentication server which is a TTP. SK_S is the private key of AS and $S_{SKS}(\cdot)$ is a signature.

3.1.3 Complexity Assumptions

Definition 1 (Learning with Errors (LWE) [36]).

Let $s \in Z^n_q$ be an n-dimensional vector. Taking $a \in Z^n_q$ as a sample, it outputs $(a,\langle a,s\rangle+e)$, where $e \in Zq$ is based on an error distribution D_e (e.g., discrete Gaussian distribution). Given arbitrary number of independent samples $\{(a_i,\langle a_i,s\rangle+e_i)\}$, the LWE problem is to find the secret vector s. The LWE problem is hard if there is no probabilistic polynomial time (PPT) adversary who can solve the problem with non-negligible probability.

3.2 Security Models

The security of our lightweight mutual authentication protocol relies on the proposed encryption scheme. It is needed to discuss both of the schemes for their security. This section defines different types of adversaries, attacks and security models.

3.2.1 Adversaries and Attacks

The abilities of an adversary are bounded by the actions that he is allowed to perform during attacks. A security model captures the specific attacks and adversaries. Now, we define some attacks and adversaries.

3.2.1.1 Attacks against Encryption Schemes

In security of the proposed encryption scheme, we consider two types of attacks. Given a public key, Type I attack attempts to recover the corresponding secret key. It aims to fully break the system rather than distinguish ciphertexts. Type II attack is a traditional ciphertext indistinguishability attack. These two attacks are related to key security and message security, respectively. Note that if a public encryption scheme is secure against Type II attack, it is also secure against Type I attack. Nevertheless, the security of private key will be analysed individually in this report. We give the definition of two adversaries for these attacks respectively.

Definition 2 (Type I and Type II adversaries). The adversaries are defined by the allowed operations and the goal of the game.

- Type I Adversary (A_I): A_I is allowed to query the algorithm KeyGen. It aims to output the secret key of the given public key.
- Type II Adversary (A_{II}): A_{II} is allowed to query the algorithm Encrypt. It aims to distinguish two ciphertext generated by using the same public key.

3.2.1.2 Attacks against Mutual Authentication Protocols

A mutual authentication aims to check the validity of two users at the end of a protocol run. An adversary against a mutual authentication protocol has different goals. This section defines some assumptions and attacks.

3.2.1.3 Assumptions

- Identities of two users and the corresponding public keys are known to each other before an authentication.
- An adversary is allowed to perform various actions on messages, such as interception and modification [37].
- Private keys are securely stored and are unknown to an adversary.

3.2.1.4 Attack Definitions

- Replay attack: An adversary eavesdrops and records protocol messages between two users and uses the captured messages to play the protocol with the target user.
- Impersonation attack: An adversary attempts to impersonate the target user to run an mutual authentication. It aims to convince the authenticity to the recipient without knowing the target user's secret.
- Man-in-the-middle attack: An adversary tampers the messages between two users. There are two goals: 1) it aims to convince the authenticity to the target recipient; 2) it aims to cheat the target recipient by a false successful mutual authentication (i.e., the recipient accepts the sender's identity, while the sender rejects the recipient).

3.2.2 Security Models

The security of public key encryption schemes usually considers the ciphertext indistinguishability. Fig. 2 shows the Indistinguishability under Chosen Plaintext Attack (IND-CPA) model.

Experiment $\mathbf{Exp}^{IND\text{-}CPA}_{\mathcal{A},\mathcal{C}}[\lambda, q, b]$:

- **Setup**: The challenger $\mathcal{C}$ runs the algorithm **Setup** to generate public parameters $params$ and runs the algorithm **KeyGen** to generate public and private key pair (PK, SK). It gives PK to $\mathcal{A}$.
- **Learning Phase**: $\mathcal{A}$ can query $Encrypt$ operation to the challenger $\mathcal{C}$ at most q times. $\mathcal{A}$ outputs two messages m_0 and m_1. It gives $\{m_0, m_1\}$ to $\mathcal{C}$.
- **Challenge Phase**: $\mathcal{C}$ randomly flips a coin $b \in \{0, 1\}$ and returns $C^* \leftarrow \mathbf{Encrypt}_{PK}(m_0)$ if $b = 0$, otherwise outputs $C^* \leftarrow \mathbf{Encrypt}_{PK}(m_1)$.
- $\mathcal{A}$ can continue to query $Encrypt$ operation.
- **Guess**: $\mathcal{A}$ outputs a bit b'.

Exp outputs *success* if $b = b'$.

Figure 2: Indistinguishability under chosen plaintext attack

Definition 3. A public key encryption scheme is IND-CPA secure if any PPT adversary A who succeeds the experiment $Exp^{IND-CPA}_{A,C}[\lambda,q,b]$ has advantage

$$Adv(A)=|Pr[b=b']-1/2|\leq\epsilon, \text{ where } \epsilon \text{ is negligible.}$$

3.3 Lightweight Mutual Authentication Protocol

The proposed protocol composes of a novel lightweight public key encryption scheme and a sequence of interactions between participants. We now present the encryption scheme and the mutual authentication protocol.

3.3.1 A Lightweight Encryption Scheme

The proposed encryption scheme consists of four algorithms: System Setup (Setup), Key Generation (KeyGen), Encryption (Encypt) and Decryption (Decrypt).

Setup. Taking as input a security parameter λ , it selects a tuple (p,t,r,w,d,q,m_{max}) which satisfies the following conditions.

1. All elements are positive integers.
2. The elements p and t are co-prime.
3. The element m_{max} denotes the maximum length of plaintext

$$m_{max} < t,\ m_{max} + wrt < p.$$

Let d and q be the number of dimensions and the number of public key vectors, respectively. It returns the system parameter $params=(p,t,r,w,d,q, m_{max})$.

KeyGen. Taking as input system parameters params, it generates a pair of public and private keys (PK,SK) as follows.

Step1: It randomly chooses a d-dimensional vector $(k_1,\ldots,k_d)$, such that $k_i \in \{1,\ldots,p-1\}$ where $i = 1,2,\ldots,d$. There is at least one k_i satisfies $\gcd(k_i,p)=1$. For the simplicity, we let $\gcd(k_d,p)=1$ and find the inverse k^{-1}_d modulo p.

Step2: A public key PK consists of q public key vectors pk, such that $pk_j = (pk_{j,1},\ldots, pk_{j,d})$, where $j=1,2,\ldots,q$. The first $d-1$ elements of pk_j are uniformly sampled from $\{1,\ldots,p-1\}$, while $pk_{j,d}$ is computed in two cases below.

For j=1, it computes $pk_{1,d}$ as

$$Pk_{1,d} = k^{-1}_d(1-(k_1pk_{1,1}+\cdots+k_{d-1}pk_{1,d-1}))$$

For $2\le j\le q$, it randomly chooses $r_j \in \{0,\ldots,r\}$ and calculates

$$pk_{j,d} = k^{-1}_d(pk_{j-1,d}+r_jt-(k_1pk_{j,1}+\cdots+k_{d-1}pk_{j,d-1})).$$

Let the public and private key pair (PK,SK) be $(\{pk_1,\ldots,pk_q\},(k_1,\ldots,k_d))$ and return (PK,SK). Note that all additions, subtractions and multiplication of two integers are followed by the modulo operation with p.

Encrypt. Let $PK=\{pk_1,\ldots,pk_q\}$ be a user's public key. To encrypt a message m, such that $m\le m_{max}$, it computes as follows.

Step1: Randomly select an integer j, where $2\le j\le q$, set $C=(pk_{j,1},\ldots,pk_{j,d})$ and $\alpha=pk_{j-1,d}$.

Step2: Randomly select an integer j again, where $2\le j\le q$, and compute $C=C+pk_j$, $\alpha=\alpha+pk_{j-1,d}(\text{mod } p)$.

Repeat this step for $w-1$ times. Note that the value of j can be repeated in different selections.

Step3: Calculate and return the ciphertext $C=C+(m-\alpha)pk_1(\text{mod } p)$.

Decrypt. Let $C=(c_1,\ldots,c_d)$ be a ciphertext generated using the public key $PK=(pk_1,\ldots,pk_q)$. To decrypt C by using the private key $SK=(k_1\ldots,k_d)$, compute

$$x=k_1c_1+\cdots+k_dc_d(\text{mod } p),$$

$$m=x(\text{mod } t).$$

3.3.2 Correctness

We now show that the proposed encryption scheme is correct. Suppose a given ciphertext $C=(c_1,\ldots,c_d)$ is generated by using the subset $S=\{pk_1,pk_i,\ldots,pk_j\}$ of the public key PK. There are w public key vectors, such that from pk_i to pk_j are randomly selected. The corresponding message m can be computed using the private key $(k_1,\ldots,k_d)$ as follows:

$$\begin{aligned}
x &= k_1c_1+\cdots+k_dc_d \\
&= k_1(\underbrace{pk_{i,1}+\cdots+pk_{j,1}}_{w} + (m-\alpha)pk_{1,1})+\cdots+k_dc_d \\
&= k_1pk_{i,1}+\cdots+k_dpk_{i,d}+(m-\alpha)k_1pk_{1,1}+\cdots+k_1pk_{j,1} \\
&\quad +\cdots+k_dpk_{j,d}+(m-\alpha)k_dpk_{1,d} \\
&= \underbrace{r_it+pk_{i-1,d}+\cdots+r_jt+pk_{j-1,d}}_{w} + (m-\alpha)\sum_{z=1}^{d}k_zpk_{1,z} \\
&= (r_it+\cdots+r_jt)+(pk_{i-1,d}+\cdots+pk_{j-1,d})+m-\alpha \\
&= (r_i+\cdots+r_j)t+m \pmod{p} \\
&= m \pmod{t}.
\end{aligned}$$

3.3.3 Our Protocol

Our mutual authentication protocol is based on the above lightweight public key encryption scheme. It is to provide the mutual authentication between two lightweight devices. Basically, the authentication protocol uses the encryption scheme to transmit challenges and check whether the recipient can respond accordingly. The proposed mutual authentication protocol is an n-pass protocol. The number of round depends on two elements: the security parameter of the protocol (i.e., the security level) and the system parameters of the encryption scheme. For example, if the challenges can be encrypted and sent in a single message, only two challenge-response rounds (3 passes) are needed. Note that we assume the identities and public keys are known to participants.

The proposed encryption scheme is adaptive to various parameter settings. However, the larger parameter values requires stronger computational and storage capabilities. The maximum length mmax of a message is bounded by the parameters t and p. To encrypt a large block of message, the value of t and p need to be increased. That is, condition 3) (in Section 3.3.1) of Setup of encryption scheme shall hold.

As motivated by lightweight Smart City applications, we consider that the end devices are resource constrained. A major challenge of lightweight authentication protocols is to avoid heavy computations. On constrained devices, such as Tmote Sky, large number computation is inefficient. It is desired to use relatively small system parameters without compromising the security of the protocol.

3.3.3.1 Proposed Protocol

To address the issue, an idea is to cut a challenge into small pieces. Each piece is encrypted by using the proposed encryption scheme under the recipient's public key. Fig. 3 shows our lightweight mutual authentication protocol. The protocol consists of four algorithms: System Setup (Setup), Key Generation (KeyGen), Initialization (Init) and Mutual Authentication (Auth).

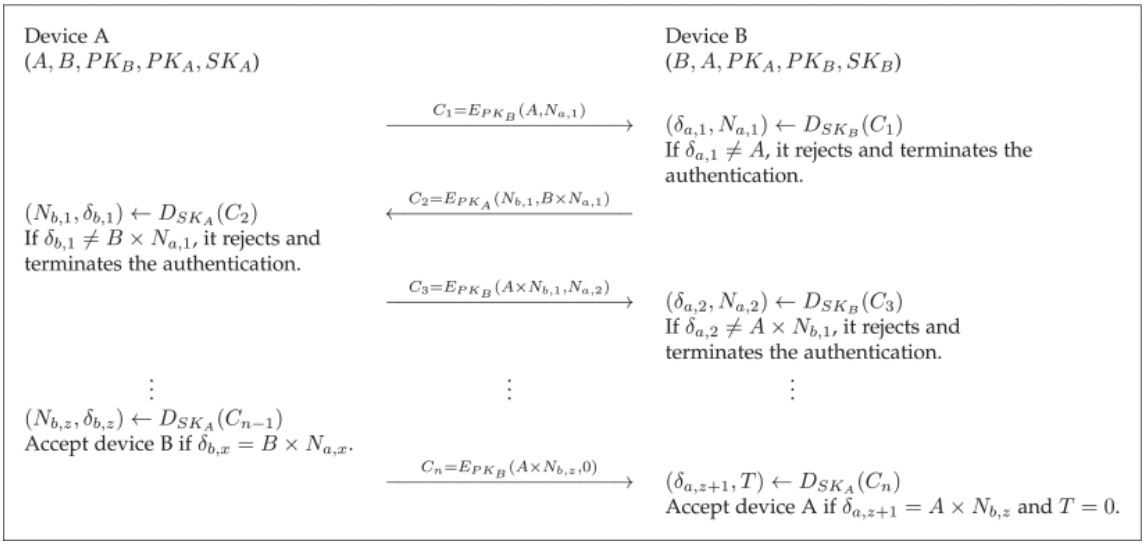

Figure 3: Lightweight mutual authentication protocol

Setup. Taking as input a security parameter λ, it runs the system setup algorithm Setup of the proposed encryption scheme. It outputs the system parameters

$$\text{params}=(p,t,r,w,d,q,m_{max},n,z),$$

where *n* and *z* are the number of passes and nonces, respectively.

KeyGen. Taking as input the system parameters params, it runs the key generation algorithm KeyGen of the proposed encryption scheme. It outputs a pair of public and private keys

$$(PK,SK)=(\{pk_1,\ldots,pk_q\},(k_1,\ldots,k_d)).$$

Init. Taking as input the identities of two devices, say A and B, and their corresponding public keys (PK_A,PK_B), they securely exchange the identities and public keys. For example, the information has been pre-configured to devices by an authorized user.

Auth. Suppose that the length of an identity is $\lfloor 1/2\log m_{max} \rfloor$ bits. Two devices A and B perform the mutual authentication as follows.

- A→B: A randomly chooses a nonce $N_{a,1} \in \{0,\ldots,\lfloor mmax/2 \rfloor\}$. It encrypts the message $(A,N_{a,1})$ using B's public key and outputs the ciphertext $C1 \leftarrow E_{PKB}(A,N_{a,1})$. It sends C_1 to device B.
- B→A: Upon receiving the message C1, device B decrypts the message and obtains $(\delta_{a,1},N_{a1}) \leftarrow D_{SKB}(C_1)$. If $\delta_{a,1} \neq A$, then it rejects the message and terminates the authentication. Otherwise, it randomly chooses a nonce $N_{b,1} \in \{0,\ldots,\lfloor mmax/2 \rfloor\}$ and computes the response C_2 as $C_2=E_{PKA}(N_{b,1},B \times N_{a,1})$. Device B sends C2 to device A.
- A→B: Upon receiving the message C2, device A decrypts the message and obtains $(Nb,1,\delta b,1) \leftarrow DSKA(C2)$. A checks $\delta_{b,1} =? B \times N_{a,1}$.

If the equation does not hold, it rejects the message and terminates the authentication. Otherwise, it accepts the message and computes the response C3 as $C3=E_{PKB}(A\times N_{b,1},N_{a,2})$, where $N_{a,2}\in\{0,\ldots,\lfloor m_{max}/2\rfloor\}$. Device A sends C3 to device B.

- A⇌B: A and B exchange challenge-response messages till the (n−1)th message.
- A→B: Upon receiving the message Cn−1, device A decrypts the message and obtains (Nb,z,δb,z)←DSKA(Cn−1). Device B is authenticated if

$$\delta_{b,z}=B\times N_{a,z}.$$

Then, A calculates the response Cn as

$$Cn=E_{PKB}(A\times N_{b,z},0).$$

Device A sends Cn to device B. Upon receiving the message Cn, B decrypts the message and obtains (δa,z+1,T)←DSKB(Cn). It checks

$$\delta_{a,z+1}=?A\times N_{b,z}, \text{ and}$$

$$T=?\ 0.$$

If the equations hold, then A is authenticated, otherwise it is rejected.

3.3.3.2 Discussion

The proposed protocol is an n-pass lightweight mutual authentication protocol. The value of n is related to the desired security level of the protocol and the system parameters of the encryption scheme. The following concrete example shows the relationship among the system parameters. Suppose that the protocol needs to provide 64-bit security for mutual authentication. A configuration could be that device A and device B select random numbers (N_a,N_b) such that $\log N_a=\log N_b=64$. They run a 3-pass (n=3) mutual authentication as below.

- (A→B): A randomly choose an integer $N_{a,1}$, such that $\log N_{a,1}=64$. It generates a ciphertext $C1=E_{PKB}(A, N_{a,1})$ and sends C_1 to device B.
- (B→A): Upon receiving the message C1, B checks if $\delta_{a,1}=A$, where $(\delta_{a,1}, N_{a,1})\leftarrow D_{SKB}(C_1)$. If $\delta_{a,1}$ is valid, B randomly picks an integer $\delta_{a,1}=64$, such that $N_{b,1}$. It computes a ciphertext $C2=E_{PKA}(N_{b,1},B \times N_{a,1})$ and sends C_2 to A.
- (A→B): Upon receiving the message C_2, A checks if $\delta_{b,1}=B\times N_{a,1}$, where $(N_{b,1},\delta_{b,1})\leftarrow D_{SKA}(C_2)$. If $\delta_{b,1}$ is valid, then B is authenticated and A responds $C_3=E_{PKB}(A\times N_{b,1},0)$. Upon receiving the message C_3 from A, B checks if $\delta_{a,2}=A\times N_{b,1}$ and T=0, where $(\delta_{a,2},T)\leftarrow D_{SKB}(C_3)$. Device A is authenticated if $\delta_{a,2}$ and T are valid.

Let the length of a device identity ID be l bits. In the above 3-pass mutual authentication, the maximum length m_{max} of plaintext (δ,N) is at least 64+l bits. The parameter p of the encryption scheme is larger than m_{max} due to condition 3) (in Section 3.3.1). To reduce the computational cost in a round, the plaintext can be cut into z pieces. As an example, a device chooses 8 nonces $(N_1,\ldots,N_8)$, such that N_i is an 8-bit nonce. Two devices A and B need 17 messages transmission to run the protocol, that is a 17-pass lightweight mutual authentication protocol. Therefore, the relationship between n and z is n=2z+1.

Lightweight device-to-device communication is usually based on UDP connection which may cause transmission errors and packet loss. In a 3-pass mutual authentication protocol, a device attempts to decrypt the whole message even if it is from an unauthorized user. On the other hand, the proposed

protocol can detect an invalid message to save energy. Note that this is specifically for low power networks based on UDP connection.

3.4 Security Analysis

In this section, we analyze the security of the proposed encryption scheme and our lightweight mutual authentication protocol.

3.4.1 Key Security

The private key is usually assumed to be securely stored. A public key encryption system will be totally broken if the private key is compromised. The adversary A_I in attack Type I aims to recover the private key of the system. We show that there is no PPT adversary A_I who has non-negligible advantage to reveal the private key. The security of a private key is based on the LWE problem.

According to the construction of public keys, there are q d-dimensional vectors $\{pk_1,\ldots,pk_q\}$. A vector pk_i, where i=2,…,q, contains a noise rit. The small noise is to hide the private key because of the hardness of LWE problem. Let the private key $SK=(k_1,\ldots,k_d)$. Considering a subset S, such that $|S|=d$, of a public key, it contains at least 2d−1 unknown values. That is, there are d private key elements and d−1 noises from elements of S. Note that each element of S is related to an equation. The system of equations cannot be uniquely solved to recover the d private key elements. Hence, the private key of our encryption scheme is secure and the security level is $\log r^{d-1}$ bits.

3.4.2 Indistinguishability

Another aspect is the indistinguishability of ciphertext. AII attempts to launch Type II attack and to succeed in the experiment $Exp^{IND-CPA}_{A,C}[\lambda,q,b]$. In our encryption scheme, there are randomly selected w public key vectors with repetitions allowed in Encrypt algorithm. These vectors are used to compute a random value to encrypt a message. Again, the encryption algorithm does not need random number generator for large numbers. It is particularly suitable to lightweight devices. Based on the scheme, a ciphertext is in form of

$$\begin{aligned} C &= m\mathbf{pk}_1 + \sum_{2\le i\le q}^{w} (\mathbf{pk}_i - pk_{i-1,d}\mathbf{pk}_1) \\ &= m\mathbf{pk}_1 + R, \end{aligned}$$

where R is a random number generated by the sum of selected public key vectors. An adversary AII runs an experiment with a challenger. The experiment outputs success iff A_{II} can distinguish random numbers R^*_0 and R^*_1. If w vectors are correctly guessed by A_{II}, he can compute R and recover the message m. Let the number of combinations of selected vectors be ρ. The advantage of A_{II} to win the experiment is

$$\mathbf{A}dv(\mathcal{A}_{II}) \ge \frac{1}{\rho}, \quad \rho = \binom{q-1}{w},$$

where 1/ρ is negligible. So that the security level of indistinguishability is $\log_\rho$ bits.

Note that the proposed scheme does not provide Indistinguishability under Chosen Ciphertext Attack (IND-CCA, or IND-CCA2 for adaptive adversary), which is a stronger security notion than IND-CPA. However, achieving IND-CCA security (generally) needs more computational cost than providing IND-CPA security. Lightweight IoT devices desire to use lightweight authentication protocols with certain security level. Also, the encrypted nonce is meaningless and it cannot be revealed based on the security analysis. An IND-CPA secure encryption scheme is sufficient to the proposed lightweight mutual authentication protocol. Therefore, we take a trade-off between the strong (IND-CCA/CCA2) security and efficiency.

6.2.1 Potential Attacks

To attack the proposed encryption scheme, an adversary could aim to find the private key (key security) or reveal encrypted message (indistinguishability). We show the resistance of such attacks by analyzing the concrete construction of the scheme.

First, let us consider the public key as the following d×q matrix.

$$\begin{bmatrix} pk_{11} & pk_{21} & \dots & pk_{q1} \\ pk_{12} & pk_{22} & \dots & pk_{q2} \\ \vdots & \vdots & \ddots & \vdots \\ pk_{1d} & pk_{2d} & \dots & pk_{qd} \end{bmatrix}$$

If an adversary is to find the private key from the matrix, it has to solve the similar system of equations analyzed in Section 3.4.1 which contains d independent equations and 2d−1 unknown variables, so that the key is not solvable from the matrix.

Second, an adversary could attempt to reveal the encrypted message directly or by educational guess. We show that the adversary cannot find the message from a given ciphertext C. Consider that the ciphertext can be rewritten as

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_d \end{bmatrix} = \begin{bmatrix} pk_{11} & pk_{21} & \dots & pk_{q1} \\ pk_{12} & pk_{22} & \dots & pk_{q2} \\ \vdots & \vdots & \ddots & \vdots \\ pk_{1d} & pk_{2d} & \dots & pk_{qd} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_q \end{bmatrix},$$

that is

$$C = \sum_{i \in \mathbb{Z}, 1 < j \leq q} a_i \mathbf{pk}_j.$$

If an adversary is able to find the solution to ai, then the message can be recovered. However, Section 3.4.2 shows that the large number of combinations result in a negligible probability of solving the system. Moreover, an adversary may consider a lattice based on the vectors $\{pk_1,\dots,pk_q\}$. According to the lattice definition, the dimension (d) should satisfy the condition d≥q (full-rank lattice if d=q). Therefore, the given vectors cannot form a lattice and $\{a_i\}$ is not solvable by using lattice operations.

3.4.3 Protocol Security

Our lightweight mutual authentication protocol applies the proposed encryption scheme as a building block. The security of our protocol relies on the encryption scheme and the Needham-Schroeder protocol. As the security analysis in Sections 3.4.1 and 3.4.2, the underlying public key encryption scheme is secure. So that the security of the proposed n-pass mutual authentication is guaranteed by the security of the Needham-Schroeder protocol [35]. It resists attacks such as man-in-the-middle attack and impersonation attack. Note that identities and the corresponding public keys are securely configured prior to an authentication. Let the length of a nonce N be ℓ bits. The security level of our protocol is (n−1)ℓ/2 bits.

4. Implementation

In this Chapter, we evaluate the performance of our protocol in software and hardware environments. On the same security level, our protocol performance is significantly better than existing RSA and ECC based protocols. We will first give some optimizations to our lightweight public key encryption scheme for enhanced security and performance. Then we present the experimental environment, implementation methods and performance comparison.

4.1 Scheme Optimizations

The following optimizations improves the efficiency and reduces the size of ciphertext and public keys.

Setup. This algorithm is the same as the Setup algorithm in Section 3.3.1 and it outputs system parameters *params*.

KeyGen. This algorithm is close to the KeyGen algorithm in Section 3.3.1. The difference is that pk_1 is represented by $pk_{1,d}$, that is $pk_1=pk_{1,d}$. It outputs a key pair $(PK,SK)=(\{pk_1,\ldots,pk_q\},<k_1,\ldots,k_d>)$.

Encrypt. Let $PK=\{pk_1,\ldots,pk_q\}$ be a public key. To encrypt message m, Step1 and Step2 of this optimized encryption algorithm are the same as these of the Encrypt algorithm in Section 3.3.1. Let the output of Step2 be (C,α), where $C=(c_1,\ldots,c_d)$. It computes

$$c_{d+1}=m-\alpha,$$

and outputs a cipher text $C=(c_1,\ldots,c_{d+1})$.

Decrypt. Taking as input a ciphertext $C=(c_1,\ldots,c_{d+1})$ and a private key $SK=(k_1,\ldots,k_d)$, it computes

$$x=k_1c_1+\cdots+k_dc_d+c_{d+1} \bmod p$$
$$m = x \bmod t.$$

4.1.1 Optimized Ciphertext and Key Size

We now show that both the ciphertext and key size can be reduced. Consider that a public key vector $pk_i=(pk_{i,1},\ldots,pk_{i,d})$, each element $pk_{i,j}$ is randomly sampled. It allows other public key vectors to reuse these elements. If a the elements $(pk_{i,1},\ldots,pk_{i,d-1})$ can be derived from other public key vectors, then the vector pki only needs to store the last element pki,d. We give two instances as follows.

1. Let $pk_i=(pk_{i,1},\ldots,pk_{i,d})$ and $pkj=(pk_{j,1},\ldots,pk_{j,d})$ be two public key vectors, such that elements of pk_i are randomly sampled from $\{1,\ldots,p-1\}$. To generate the vector pkj, we shift the first $d-1$ elements of pki by one position and set pkj as

   $$pk_j=(pk_{i,2},\ldots,pk_{i,d-1},pk_{i,1},pk_{j,d}).$$

   $pk_{j,d}$ is computed as in the KeyGen algorithm by given $(pk_{j,1},\ldots,pk_{j,d-1})$ and SK. Note that a vector can be reused at most $d-1$ times to avoid the case in which two public vectors have the same first $d-1$ elements.

2. Given a vector pki, the vector pkj is defined as

   $$(pk_{i,1}pk_{i,2},pk_{i,2}pk_{i,3},\ldots,pk_{i,d-1}pk_{i,1},pk_{j,d}).$$

   This method can be used with the above method that it can largely reduce the size of public keys.

The public key size has been reduced further in our implementation. Taking Table 3 as an example, an element of a public key vector should be 4 bytes. For each of the first $d-1$ elements, the size can be

reduced to 1 byte, while the last element $pk_{i,d}$ should remain in 4 bytes. In this case, the size of an optimized public key vector is 70 percent smaller than that of the original key vector.

The ciphertext size is also reduced because of the smaller public key size. Taking our configurations (Table 3) as an example, the original ciphertext size is 56 bytes and the optimized ciphertext is 34 bytes.

Table 3: Parameters of Our Protocol

Parameter	Value
p	$4,294,967,296$
t	$65,537$
r	$2,048$
w	32
d	14
q	90
$m_{\max}$	$65,535$
n	21
Key Security	143 bits
Indistinguishability	80 bits
Protocol Security	80 bits

4.1.2 Security Level

The security of the optimized encryption scheme is improved. Because the first public key vector is hidden, for any d equations from the definition of public key contain 2d unknown variables, instead of 2d−1 unknowns in the original scheme. The key security level of the optimized scheme is $\log r^d$ bits. The message security level $log\binom{q-1}{w}$ remains bits.

4.2 Experimental Environment

4.2.1 Contiki OS

Contiki is an open source operating system for the development of resource constrained devices. It can emulate and verify the validity of IoT networks. To conduct an evaluation, a tool called Cooja simulator is used in our experiment. Contiki supports various network protocols, standards and hardware platforms. For example, it implements the wireless standard IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) which is commonly used in IoT applications. Some popular protocols such as Constrained Application Protocol (CoAP) and HTTP are also supported. A software implementation can be emulated on the target hardware platform prior to testing it on real devices. An emulation precisely shows the behavior of specific device. Contiki OS also provides an even-driven kernel with mutli-threading features [38].

4.2.2 Software Emulation Environment

Software emulation was performed in the Cooja simulator. We used the instant Contiki on virtual machine to run an emulation. Tmote Sky is a popular mote and it is well supported by Contiki. So that we choose Sky mote as the hardware platform and implemented our schemes on it. Note that we used default configurations of Contiki for sky mote. The frequency of sky mote in the emulation was 3.9 MHz rather than the standard 8 MHz. Table 2 shows some configurations of our emulation environment.

The implementation of our protocol was done by using C programming language without assembly code. The performance could be further improved if assembly code is applied. Note that all implementations were performed and emulated on sky mote with the same configuration in Table 2.

4.2.3 Hardware Environment

We used CC2538 evaluation modules (Fig. 4) to test the performance of our protocols. The microcontroller of CC2538 is more powerful than that of Sky mote. For 80 bits protocol security level, the optimized mutual authentication protocol took around 125 ms. The speed of the original protocol and the optimized protocol are close because the difference of computational cost can be ignored on this platform.

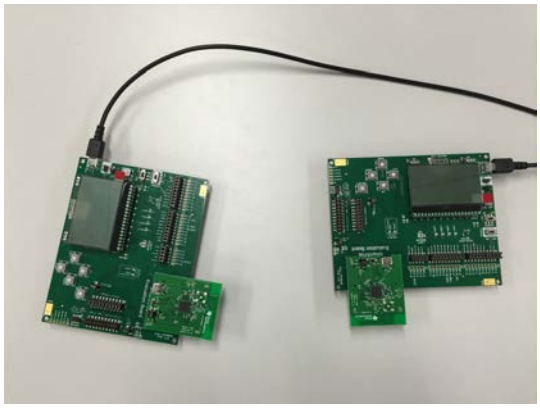

Figure 4: Device-to-device mutual authentication using CC2538 evaluation modules

4.3 Implementation and Performances

The implementation of our mutual authentication protocol considered the restrictions of lightweight devices. To avoid large number computation, the parameters of schemes were set to small numbers. Table 3 shows the values of parameters of the proposed protocol. The parameter settings guarantee the encryption scheme achieves 80 bits message security with at least 143 bits key security and the security of protocol is 80 bits. The implemented encryption scheme took 2 bytes as the maximum message length. According to our protocol, a ciphertext can transmit a nonce of 8 bits. Hence, the protocol needs 21 passes to achieve 80 bits security level.

We evaluated the performance of our implementation with Sky mote in Cooja simulator. In the first experiment, two Sky motes communicated directly. Table 4 shows the performance of our protocols in the security level of 80 bits. An encryption took about 34 ms in the original encryption scheme, while it is reduced to 29 ms in the optimized scheme. The ciphertext size is also reduced from 448 bits to 272 bits. Therefore, the total mutual authentication time was 134 ms less than the original protocol.

The performance results show that the proposed schemes are efficient due to the lightweight computational requirements. The proposed schemes perform modulo addition, subtraction and multiplication, while it avoids modulo exponentiation computations which are time-consuming. Note that the specific parameters used in the experiment depend on the security requirement. For example, the parameters in Table 3 is sufficient to achieve the 80-bit (protocol) security. Table 5 shows the computational cost of proposed encryption schemes which will be repeated n times for a mutual authentication. It is clear that n depends on the specific implementations. Assume that the desired security is 80-bit, n=2 if each nonce is 80 bits, but n=21 if a nonce is 8 bits. For lightweight IoT devices, the settings could be vary to suit individual applications.

Table 4: Device-to-Device Performance

Protocol	Encrypt [ms]	Decrypt [ms]	Ciphertext [bits]	Mutual Auth. [ms]
Orig. protocol	34	2	448	1535
Opt. protocol	29	2	272	1401

Table 5: Performance of Proposed Encryption Schemes

	Encrypt	Decrypt
Orig. scheme	$(wd + w - 1)A + 1S + dM$	$(d - 1)A + dM$
Opt. scheme	$(wd + w - d - 1)A + 1S$	$dA + dM$

Table 6: Performance Comparison with ECC and RSA

Security Level [bits]	RSA [ms]	ECC [ms]	Our Opt. Protocol [ms]
80	23,500	6,160	1,401
112	175,560	16,600	1,960

Table 7: Device-to-Server Performance

Number of Motes	First Auth. [ms]	Total Auth. [ms]	Message Retrans. Rate
15	13,628	97,549	1.83%
25	16,182	132,865	4.8%
35	23,816	180,639	8.05%
45	41,812	344,118	11.76%

Table 6 shows a performance comparison with ECC and RSA baesd on [39]. The ECC protocol refers to a handshake using ECDSA. Without loss of generality, we convert the result of [39] to the Sky mote at 3.9 MHz. On the security level of 112 bits, the speed of our optimized protocol is around 88 and 7 times faster than RSA and ECC, respectively.

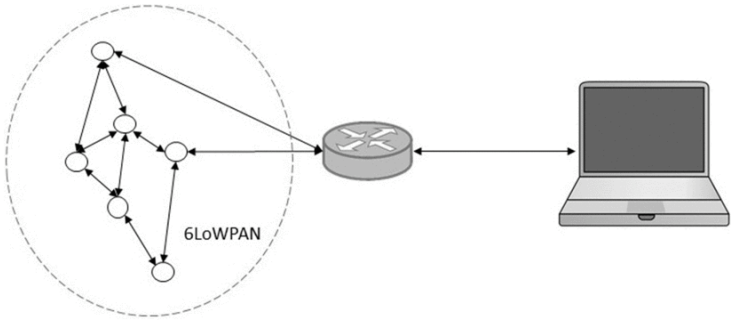

Figure 5: Device-to-Device mutual authentication architecture

In Smart City applications, a system usually obtains environmental information from a set of sensors. It requires a mote (with sensors) to interact with a server such as a Linux host. Fig. 5 illustrates an

architecture of device-to-server environment. It consists of three components: motes, edge router and the server. The edge router communicates with motes through 6LoWPAN network and the router also connects to the server via IPv4/IPv6

To evaluate our solution in this scenario, we tested the performance under different number of Sky motes. Table 7 shows the performance of the optimized protocol. The time obtained in this table is counted from the system boot so it is different from the result of previous tables. Taking 15 motes as an example, the first mutual authentication can be done in approximately 13,628 ms after the system started. The total mutual authentication time for 15 motes is 97,549 ms. The Message Retransmission Rate (MRR) represents the probability in which a protocol message needs to be resent. With the increasing number of motes, the MRR value will be larger. A reason is the weak communication environment. In the experiment, all Sky motes were booted simultaneously and they sent messages to the communication channel. Some messages or server responses were lost because a large number of messages block the network. Specifically, a retransmission operation is most likely occurred in the beginning phase. If more motes are authenticated, the retransmission rate will be decreased. Certainly, MRR can be minimized if we design a mechanism to control the system boot.

The performance of authentication system is also influenced by the network topology. An inappropriate topology can make a mote temporarily unreachable in 6LoWPAN network and the latency can be raised. In either case, a retransmission operation will be carried out every one minute (in our configuration). It will seriously delay the mutual authentication process. Therefore, we randomly placed the motes and Table 7 shows an average in different topologies.

4.4 Online/Offline

In digital signatures, online/offline signatures [40] can be used for quick signing on lightweight devices. In the offline phase, a device pre-computes a part of signature before the message is given. It then securely stores the intermediate result. In the online phase, the device computes the signature based on a stored value and the given message. Online/offline signature schemes perform heavy computations in offline phase so that it can generate a signature quickly during an authentication. Problems to lightweight devices are two-folds: the memory storage space and the tamper-resistant capabilities.

Assume that the Sky mote can securely store some intermediate value. The efficiency of our protocols can be significantly improved. As we can see, most computations are performed in encryption scheme. The idea is to carry out Step1 and Step2 of an encryption offline. When a message is given during an authentication, the device performs Step3 with a stored value of Step3. For example, the optimized encryption scheme only needs to compute one subtraction for a ciphertext and the generation time is negligible. The time of mutual authentication using the optimized protocol can be reduced to 792 ms from 1,401 ms.

5. OAuth Protocol for SDN based security Architecture

As part of Milestone 3, we are working on the development of a security architecture for IoT networks by leveraging the underlying features supported by Software Defined Networks (SDN). Our security architecture restricts network access to authenticated IoT devices. We will use fine granular policies to secure the flows in the IoT network infrastructure and provide a lightweight protocol to authenticate IoT devices. In particular, we aim to use Lightweight Elliptic Curve Cryptography (ECC) to achieve authentication and OAuth Protocol for authorisation in our security architecture. In this chapter, we present brief overview of our approach. A more detailed report on the security architecture will be provided as part of Milestone 3 deliverable.

5.1 Lightweight ECC based Authentication for IoT Devices

We are working on the development of Lightweight ECC based Authentication for IoT Devices The ECC based public key system uses the algebraic structure of elliptic curves as their finite points. It is computationally faster than other public key cryptosystems such as RSA. Hence, it is more suitable for computationally constrained IoT devices. It can be used to achieve key agreement as well as encryption and digital signature.

In our SDN security architecture, the IoT devices will be authenticated using a lightweight ECC based authentication protocol proposed in [42]. Then we will use the key established using this protocol to encrypt the data from the authenticated IoT devices. This lightweight security protocol will work in conjunction with the OAuth protocol. The ECC based authentication protocol used in our architecture will have 3 stages: Setup, Installation and Key Agreement. We will show that the ECC authentication scheme is secure against active attackers who are capable of eavesdropping, modifying and injecting messages in the protocol.

5.2 Authorization for Network Services using OAuth Protocol

The authorization service using the OAuth protocol [43, 44] works as follows: It consists of four actors: i) Client, ii)Resource Owner, iii) Resource Server and iv) Authorization Server. The client contacts the Resource Owner of the resource. The Resource Owner grants the access to the client by sending an authorization code. The client delivers the received authorization code to the Authorization Server. The Authorization Server verifies the authorization code and releases a token containing the details of the consent provided to the client (time limit, scope, and so on). The client forwards the token to the Resource Server. The Resource Server checks the validity of the received token, and in the affirmative case provides access to the protected resource.

6. Conclusion

In this report we presented a summary of work done as part of second milestone “Lightweight Authentication Mechanism and OAuth Protocol for IoT Devices” which consists of the following tasks: i) Design of the Lightweight Authentication Protocol for IoT and ii) Security Analysis of the Authentication Protocol for IoT.

As part of this milestone, we developed a lightweight mutual authentication protocol based on a novel public key encryption scheme for smart city applications. The proposed protocol takes a balance between the efficiency and communication cost without sacrificing the security. We evaluate the performance of our protocol in software and hardware environments. On the same security level, our protocol performance is significantly better than existing RSA and ECC based protocols. We also provide security analysis of the proposed encryption scheme and the mutual authentication protocol. The proposed protocol is an n-pass lightweight mutual authentication protocol. The value of n is related to the desired security level of the protocol and the system parameters of the encryption scheme. Our lightweight mutual authentication protocol applies the proposed encryption scheme as a building block. The security of the proposed n-pass mutual authentication is guaranteed by the security of the Needham-Schroeder protocol. We will show that our protocol can resists attacks such as man-in-the-middle attack and impersonation attack. We evaluated the protocol on Contiki OS and CC2538 evaluation modules. The experimental evaluations show that our protocol is respectively 88 and 7 times faster than RSA and ECC on the security level of 112 bits. The mutual authentication time can be further reduced if online/offline technique is enabled.

We are working on the development of a security architecture for IoT networks by leveraging the underlying features supported by Software Defined Networks (SDN). We aim to use Lightweight Elliptic Curve Cryptography (ECC) to achieve authentication and OAuth Protocol for authorisation in our security architecture.

The authors would like to thank ISIF ASIA for their financial contribution to the Project.

REFERENCES

1. A. Zanella, N. Bui, A. P. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, Feb. 2014.
2. H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, “Smart cities and the future internet: Towards cooper- ation frameworks for open innovation,” in The Future Internet- Future Internet Assembly 2011: Achievements and Technological Promises. Berlin, Germany: Springer, 2011, pp. 431–446.
3. I. Anshel, M. Anshel, D. Goldfeld, and S. Lemieux, “Key agreement, the Algebraic Eraser™ , and lightweight crytography,” Con- temporary Math., vol. 416, no. 2006, pp. 1–34, 2006.
4. E. Artin, “Theory of braids,” Ann. Math., vol. 48, pp. 101–126, 1947.
5. Security in low resource environments, 2006. [Online]. Available: http://www.securerf.com/wp-content/uploads/2014/03/ White-Paper-Security-in-Low-Resource-Environments.pdf
6. J. Hoffstein, J. Pipher, and J. H. Silverman, “NTRU: A ring-based public key cryptosystem,” in Proc. 3rd Int. Symp. Algorithmic Number Theory, 1998, pp. 267–288.
7. Standards for efficient cryptography SEC 1: Elliptic curve cryptography, 2000. [Online]. Available: http://www.secg.org/SEC1-Ver-1.0.pdf
8. Y. K. Lee, L. Batina, and I. Verbauwhede, “EC-RAC (ECDLP based randomized access control): Provably secure RFID authenti- cation protocol,” in Proc. IEEE Int. Conf. RFID, 2008, pp. 97–104.
9. Y. K. Lee, L. Batina, D. Singel'ee, and I. Verbauwhede, “Low-cost untraceable authentication protocols for RFID,” in Proc. 3rd ACM Conf. Wireless Netw. Secur., 2010, pp. 55–64.
10. Y. K. Lee, L. Batina, and I. Verbauwhede, “Untraceable RFID authentication protocols: Revision of EC-RAC,” in Proc. IEEE Int. Conf. RFID, 2009, pp. 178–185.
11. J. Hermans, R. Peeters, and B. Preneel, “Proper RFID privacy: Model and protocols,” IEEE Trans. Mobile Comput., vol. 13, no. 12, pp. 2888–2902, Dec. 2014.
12. M. Brown, D. Cheung, D. Hankerson, J. L. Hernandez, M. Kirkup, and A. Menezes, “PGP in constrained wireless devices,” in Proc. 9th Conf. USENIX Secur. Symp., 2000, pp. 19–19.
13. K. H. Ko, S. Lee, J. H. Cheon, J. W. Han, J. Kang, and C. Park, “New public-key cryptosystem using braid groups,” in Proc. 20th Annu. Int. Cryptology Conf. Advances Cryptology, 2000, pp. 166–183.
14. E. Lee, S. Lee, and S. G. Hahn, “Pseudorandomness from braid groups,” in Proc. 21st Annu. Int. Cryptology Conf. Advances Cryptology, 2001, pp. 486–502.
15. I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, “New key agree- ment protocols in braid group cryptography,” in Proc. Conf. Topics Cryptology: Cryptographer’s Track RSA, 2001, pp. 13–27.
16. K. H. Ko, D. H. Choi, M. S. Cho, and J. Lee, “New signature scheme using conjugacy problem,” IACR Cryptology ePrint Archive, vol. 2002, 2002, Art. no. 168.
17. S. Lee and E. Lee, “Potential weaknesses of the commutator key agreement protocol based on braid groups,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn. Advances Cryptology, 2002, pp. 14–28.
18. D. Hofheinz and R. Steinwandt, “A practical attack on some braid group based cryptographic primitives,” in Proc. 6th Int. Workshop Theory Practice Public Key Cryptography, 2003, pp. 187–198.
19. A. Kalka, M. Teicher, and B. Tsaban, “Short expressions of permutations as products and pryptanalysis of the algebraic eraser,” Advances Appl. Math., vol. 49, no. 1, pp. 57–76, 2012.
20. A. Ben-Zvi, S. R. Blackburn, and B. Tsaban, “A practical cryptography of the algebraic eraser,” IACR Cryptology ePrint Archive, vol. 2015, p. 1102, 2015.

21. S. R. Blackburn and M. J. B. Robshaw, “On the security of the Algebraic Eraser tag authentication protocol,” Appl. Cryptography Netw. Security-14th Int. Conf., ACNS 2016, Proc., ser. LNCS,
22. M. Manulis, A. Sadeghi, and S. Schneider, Eds., Guildford, UK, Springer, vol. 9696, pp. 3–17, Jun. 2016.
23. I. Anshel, D. Atkins, D. Goldfeld, and P. E. Gunnells, “Defeating the Ben-Zvi, Blackburn, and Tsaban attack on the algebraic eraser,” CoRR, vol. abs/1601.04780, 2016. [Online]. Available: http://arxiv.org/abs/1601.04780
24. D. Goldfeld and P. E. Gunnells, “Defeating the kalka–teicher–tsaban linear algebra attack on the algebraic eraser,” CoRR, vol. abs/ 1202.0598, 2012. [Online]. Available: http://arxiv.org/abs/1202.0598
25. R. A. Perlner and D. A. Cooper, “Quantum resistant public key cryptography: A survey,” in Proc. 8th Symp. Identity Trust Internet, 2009, pp. 85–93.
26. IEEE P1363, (2008). [Online]. Available: http://grouper.ieee.org/ groups/1363/
27. J. Hermans, F. Vercauteren, and B. Preneel, “Speed records for NTRU,” in Proc. Int. Conf. Topics Cryptology, 2010, pp. 73–88.
28. D. Stehl'e and R. Steinfeld, “Making NTRU as secure as worst-case problems over ideal lattices,” in Proc. 30th Annu. Int. Conf. Theory Appl. Cryptographic Techn. Advances Cryptology, 2011, pp. 27–47.
29. R. Peeters, J. Hermans, and J. Fan, “IBIHOP: Proper privacy pre- serving mutual RFID authentication,” in Proc. Workshop RFID IoT Secur., 2013, pp. 3–16.
30. S. Vaudenay, “On privacy models for RFID,” in Proc. Advances Crypotology 13th Int. Conf. Theory Appl. Cryptology Inf. Secur., 2007, pp. 68–87.
31. C.-I. Lee and H.-Y. Chien, “An elliptic curve cryptography-based RFID authentication securing E-health system,” Int. J. Distrib. Sen- sor Netw., vol. 2015, no. 642425, pp. 109–126, 2015.
32. C. Hsu, S. Wang, D. Zhang, H. Chu, and N. Lu, “Efficient identity authentication and encryption technique for high throughput RFID system,” Secur. Commun. Netw., vol. 9, no. 15, pp. 2581–2591, 2016.
33. S. Wang, T. Lei, L. Zhang, C. Hsu, and F. Yang, “Offloading mobile data traffic for QoS-aware service provision in vehicular cyber-physical systems,” Future Generation Comput. Syst., vol. 61, pp. 118–127, 2016.
34. S. Wang, C. Fan, C. Hsu, Q. Sun, and F. Yang, “A vertical handoff method via self-selection decision tree for internet of vehicles,” IEEE Syst. J., vol. 10, no. 3, pp. 1183–1192, Sep. 2016.
35. R. M. Needham and M. D. Schroeder, “Using encryption for authentication in large networks of computers,” Commun. ACM, vol. 21, no. 12, pp. 993–999, 1978.
36. G. Lowe, “An attack on the needham-schroeder public-key authentication protocol,” Inf. Process. Lett., vol. 56, no. 3, pp. 131– 133, 1995.
37. O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in Proc. 37th Annu. ACM Symp. Theory Comput., 2005, pp. 84–93.
38. 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.
39. A. Dunkels, B. Gro€nvall, and T. Voigt, “Contiki-A lightweight and flexible operating system for tiny networked sensors,” in Proc. 29th Annu. IEEE Conf. Local Comput. Netw., 2004, pp. 455–462.
40. K. Piotrowski, P. Langendorfer, and S. Peter, “How public key cryptography influences wireless sensor node lifetime,” in Proc. 4th ACM Workshop Secur. Ad Hoc Sensor Netw., 2006, pp. 169–176.

41. S. Even, O. Goldreich, and S. Micali, “On-line/off-line digital schemes,” in Proc. 9th Annu. Int. Cryptology Conf. Advances Cryptology, 1989, pp. 263–275.
42. A. Mohammadali et al., “A novel identity-based key establishment method foradvanced metering infrastructure in smart grid,”IEEE Trans. on Smart Grid, 2016.
43. D. Hardt, “The oauth 2.0 authorization framework,” 2012
44. S. Sciancalepore et al., “Oauth-iot: An access control framework for the internetof things based on open standards,” inComputers and Communications (ISCC),2017 IEEE Symposium on. IEEE, 2017, pp. 676–681