A Brief Review of Classic IoT security Protocols

Tingxuan Tang
Chengdu University of Technology, Chengdu, 610059, China
1159856320@qq.com

Abstract. Recent years have witnessed that Internet of Things (IoT) became more and more popular in many technical fields. Due to its high utility, IoT is closely linked with a number of domains in people’s daily life; hence, it is really crucial and necessary to reinforce and maintain the security of IoT. In fact, there are numerous protocols proposed to protect it, and these researches aim to find an effective way to resist attacks from hackers. In order to improve safety index of the existing models and build a nearly brilliant model. This paper analyzes and compares three published studies which are relevant to this topic through several factors. This paper describes these protocols from an understandable and explicit perspective which makes readers easy to catch the main idea of each protocol. After detailed explanation of the algorithm and analysis of several safety indicators, according to several comparison tables and bar charts, we finally figure out the best protocol among these three studies, and specific results are showed in the paper. Furthermore, we also give some available and useful suggestions about the future work to optimize the protocol in this industry.

Keywords: Internet of things, Security, Encryption, Decryption.

1. Introduction

In this increasingly developed world, Internet of Things (IoT) becomes to a hot topic in some specific fields. Throughout the course of human history, there are a large number of skills and technologies innovated, advanced and used to facilitate humans’ daily life. These inventions had a significant influence on most of crucial fields for human survival, like agriculture, transportation, communication and so on. The Internet of Things, according to its name, is a skill used to gather data from different devices to any virtual platform on existing Internet infrastructure [1], which is such a developed technology that revolutionizes almost every aspect of our lives [2]. In common situations, “Internet of Things” usually has three concepts. The first one is that the resulting global network interconnect smart devices and objects through different methods of extended Internet technologies. And it also refers to all of the supporting technologies which can help to achieve this scenario, for example, Radio Frequency Identification System (RFIS), sensor and so on. Thirdly, it represents the series of applications and services which apply these technologies to explore new businesses, markets and industries [3]. Owing to its powerful features and efficiency, IoT becomes to a popular target to be attacked. The domain of IoT is sharply developed due to the rapid growth in the quantity of devices connected to the Internet. In the train of the situation, most parts of the IoT are usually under attacks, which results in serious privacy threats and data leakages. In recent decades, there were lots of unauthorized network and system compromising cameras, routers and printers, and modifying their code to turn these devices into bots participating in a larger scale of attack. For instance, in 2016, the Dyn attack made some popular websites and applications like Twitter and Netflix cannot be accessed. Furthermore, the kinds of security and privacy issues in IoT are in a huge range. To be intuitive, IoT is actually built on top of the Internet; hence, it inherits portions of security problems of the Internet. Additionally, the system faces cross-layer heterogeneous integration issues, and each layer of the IoT architecture also has its own risks [4].

As a result, there are massive scientists designing models and schemes to protect the security of IoT. For example, in some specific systems, one of the most serious attacks which leads to significantly negative effects to IoT is the eavesdropper attack. [5] proposed that there is a specific method called stochastic cooperative jamming approach which is helpful to frustrate eves activities at random positions of a network. Then, in [6] and other relevant studies, these scientists concentrated on the problem about maximization of privacy rate with one to multiple eavesdroppers existing in
line with different receivers, transmitters and eavesdropper modes and diverse forms of transmissions. Channel State Information (CSI). Whereas, some studies were completed under the situation that both of transmitter and receivers did not learn the CSI of Eve. In 2022, [7] proposed that, they created a secured multiple user secrecy model for wireless IoT by using stochastic privacy optimization. In their scenario, under the situation that network users do not know the CSI of Eve, the system can establish a safe connection inter the multiple-antenna transmission. The presented technique makes sure that Eve can achieve basic or optimal matched filtering while smart jamming model is used in transferring IoT node to balance the Eve actions. Although compared with [5], the stochastic optimization technique in this study avoids the deterioration of data rate to the initial devices of IoT, the protocol requires long computation time and expensive implementation cost, and it is also difficult to encrypt and deal with multi-servers.

On the other hand, for the multiantenna concept, both slow and fast fading channels took AN aided transmit strategies into consideration by [8]. Furthermore, secrecy rate and secrecy throughput that hampered by secrecy outage probability (SOP) are usually considered as the performance metrics for slow fading channels, and ergodic secrecy rate is usually assumed as the fast-fading channels privacy metric according to [9]. To be specific, for the goal of minimizing SOP, [10] refined the power allocation between two different signals when there is only one eavesdropper, and [11] did it when there are several eavesdroppers.

Based on these tentative learning about the IoT domain, this paper initially introduces some related works and relevant background information. And then, it gives some basic but necessary knowledge and mathematical foundation and formulas to help readers have a better understanding of the following paragraphs. After that, it explains three chosen studies further with detailed tables and mathematical derivation processes, and the main steps are all display in clear graphs. What is more, this study also compares these models through different factor in several tables and bar chart and it provides corresponding explanation about them. At the end of the essay, it proposes some potential problems, hidden hazards and tries to give some available and feasible solutions in the future.

2. Preliminaries

Response Time: it refers to the time between the moment a client sends out a request and the moment the server sends back its first response, and it is usually measured in milliseconds.

Time-To-Live (TTL): it refers to the longest alive time of a certificate [12].

Serial number (S/N): it refers to the exclusive manufacturer number of the device [12].

Digital signature (DS): it refers to the given common digital signature imprint of the certificate [12].

Hyper Elliptic Curve Cryptography: a high-security group which is available to fast genus-2-hyperelliptic-curve formulas for variable-base-point single-scalar multiplication and fast elliptic-curve formulas for fixed-base-point scalar multiplication and multi-scalar multiplication [13].

Chosen-Ciphertext Attack (CCA): it refers to the kind of attack where adversaries try to get private information through decryption oracle [14].

Approximately smooth projection hash function: assume that $hk$ is a tuple whose elements follow a discrete Gaussian distribution, for $hk \in K, x = (y, w) \in X$, compute $H_{hk}(x) = H_{hk}(y, w)$ and $hp = a(hk)$, for a random number $s$, $H_{hk}(y, w) \approx \text{Hash}(hp, (y, w), s)$. And the statistical distance between $[hp, H_{hp}(x)]$ and $(hp, p)$ is negligible on the safety parameter [15].

Right Circular Shift function: it is one of the shift micro-operations which are applied to the serial transfer of information. In common situation, they are also used with arithmetic micro-operation and other data-processing operations. In this function, every bit in the register will be shifted to their right, and after the changing, the MSB will be vacant, as a result, the LSB’s value replaces. Fig. 1 displays this process.
Key consensus: it refers to: \( KC = (\text{params, Con, Rec}) \), where \( \text{params} = (q, m, g, d, aux) \) represents system parameters, and \( \text{Con} = (k_1, v) \) is probability polynomial time conciliation algorithm, \( k_1 \) is the key which is shared by others, \( v \) donates the hint signal which will be sent to another communication stage publicly in the subsequent process to make these two parties reaching a consensus. \( \text{Rec} = k_2 \) is the algorithm which is used to determine the polynomial time reconciliation [15].

Learning With Errors (LWE): it is a mathematical system which is applied in many cryptographic schemes. It usually helps to recover some messages by solving a sequence of ‘approximate’ randomly generated linear equations. Initially, the system was capable of taking up to build a Public Key Encapsulation model [16].

3. **Analysis of Three Classic Schemes**

   **Study 1. Communication-aware Adaptive Key Management (CAKM)**

   The aim of this scheme is improving active IoT devices’ end-to-end security, and it also pays attention to the communication session between devices. In order to implement the distribution of security keys, the scheme mainly uses hyper elliptic curve cryptography. And we will explain the CAKM model from two stages [12].

   **3.1 Devices anonymity and request time classification**

   In this phase, authenticating the physical device is achieved by an external certificate authority (CA). In order to make it anonymity, the devices in communication use gateways or cloud to store or retrieve data.

   Step 1. Fix the response time \( t_s \) of the cloud service provider by initial authentication

   Step 2. Extend the TTL of the authentication based on persisting communication. The device submits its unique ID and S/N to the CA, and if the certificate is verified mutually, CA will response it. Furthermore, the validity of the certificate generally depends on its \( t_s \).

   Step 3. Thus, scientist proposed these two equations:

   \[
   t_d + t_s = t_r + t_w + t_t \quad (1)
   \]

   When \( r_d = r_r \),

   \[
   r_d \times t_r = p_r \times t_r \quad (2)
   \]

   In mathematical equation (1), \( t_d \) represents the delay of responding to a request; \( t_r \) represents the time of request initialization; \( t_w \) represents the time of waiting and \( t_t \) represents the time of transmission. And this formula is used to estimate \( t_d \). In mathematical equation (2), \( r_d \) represents the frequency of accepting device requests of processing; \( r_r \) represents number of serviced requests and \( p_r \) represents the rate of request processing. And this formula is used to check whether \( p_r \) meets the standard. In other words, \( p_r \) at time \( t_r \) should cover \( r_r \) at that time, and the TTL of the certificate should be updated based on the value of \( t_d \).

   Step 4. Compare TTL and \( t_d \)
Situation 1. $t_d < \text{TTL}$
Under this situation, mathematical equation (2) holds, and if there is no extra $r$, the certificate will face revocation, and the next part of communication will be kept by the device who does not have TTL field. Consequently, the device will ask for new TTL through reporting its ID, S/N and previous TTL. The $t_d$ is divided into N parts to allocated requests.

Situation 2. $t_d = \text{TTL}$

Under this situation, interval of the certificate should update to $TTL - 1$, and the communication time should extend to $t_d - 1$. For N requests, the average interval of servicing a request is $\frac{t_d}{N}$, thus, the accurate time is:

$$[t_r \times (n - 1)], \forall t_r < t_d$$

Set $X (X<N)$ as the number of requests processed in the time $t_d$, then the next TTL is $t_d \times (N - X)$. So, the IoT device can accept and handle requests in time $t_d \times (N - X)$.

Situation 3. $t_d > \text{TTL}$
Under this situation, the TTL should be prolonged. And the IoT device can precomputes the time to $(N-1) t_d$. If the TTL is extended successfully, the required time to service changes to $\sum_{i=1}^{N} t_{d_i}$.

3.2 Communication authentication
In this phase, communication authentication achieved by HECC through generating keys, and the mode of authentication is preceded as Digital Signature (DS). Then, the hyper elliptic range for generating random points is decided by CA [12].

Set $C$ as the hyper elliptic curve spread over a space $S$, according to genus function $G$,

$$C = y^2 + h(x)y$$ (3)

In mathematical equation (3), $y$ represents a point on the hyper elliptic curve with $h(x)$ degree. Set $p_{k-s}$ as the public key of sender and $p_{k-r}$ the public key of receiver, $pv_{k-s}$ and $pv_{k-r}$ as their private keys. If the sender takes action, it will ask for $p_{k-r}$ from the receiver. Set $[\tau, \Delta]$ as the encoded pair of message components. In order to ensure the security of message $M$, $\tau$ and $\Delta$ can be represented as:

$$\tau = l \ast p$$  
$$\Delta = M + l \ast p_{k-r}$$ (4)

In mathematical equation (4), $p$ and $l$ should both prime numbers, and senders creates a secret key: $s_{k-s}$ during the interval $[t_r, t_d]$. 

Fig. 2 describes processes of three different situation after comparing TTL and $t_d$. Among them, (a) shows how the scheme works when $t_d < \text{TTL}$, and similarly, (b) donates the situation that $t_d = \text{TTL}$, and (c) represents the process when $t_d > \text{TTL}$. 

![Fig. 2 Process of the first stage](image-url)
thus, $s_{k-s}$ can be:

$$s_{k-s} = \left\{ \begin{array}{ll}
\sum_{i=1}^{n} i \prod_{i=1,j\neq i}^{n} \frac{j}{j-1} |p|, & t_d = TTL \\
\sum_{i=1}^{n-x} i \prod_{i=1,j\neq i}^{n-x} \frac{j}{j-1} |p|, & t_d > TTL
\end{array} \right.$$ 

During this period, the receiver will decode it to a secret key $S_{k-r} = \sum_{i=1}^{n} p v_{k-r_i} \bigcap C_p$.

When $t_{r+1} = s_{k-s} = s_{k-r}$, then the message can be decoded. And the encrypted message is $M = \Delta - \text{tr}$. The scheme provides advanced authentication which can resist security attacks better, At the same time, without influencing the security level, IoT base authentication modes is much more efficient.

Study 2. Password authentication key exchange (PAKE)

The aim of this scheme is allowing the device to exchange their keys when there are less authentication information and time for the exchange. In order to generate an asymmetric key agreement structure, the scheme mainly uses the approximate smooth projection hash function and key consensus. And we will explain the PAKE model through two rounds [15].

Step 1. get the key pair through the CCA security encryption scheme $(A_{i,b}, S_{i,b})$, and the public key $(pk)$ and private key $(sk)$ are:

$$pk = \{A_{i,0}, A_{i,1}\}_{i\in[n]}$$
$$sk = \{S_{i,0}, S_{i,1}\}$$

Step 2. Choose and compute parameters of the key consensus:

$$params = (q, m, g, d, aux)$$
$$aux = \{q' = lcm(q, m), \alpha = \frac{q'}{q}, \beta = \frac{q'}{m}\} \quad (5)$$

In mathematical equation (5), $lcm(q, m)$ is the least common multiple of $q$ and $m$. In addition, the password $w$ should be shared by the prover and the verifier.

3.3 Round1

Step 3. Prover calculates the ciphertext: $y = A(s, 1, w)' + x$, where $s$ and $x$ are both random vectors; $A$ is a public key. Then it will be sent to verifier.

3.4 Round2

Step 4. Verifier selects hash key $(hk)$ and calculate the projection key $(hp)$: $hk = (h_1, ..., h_k)$ and $hp = B^T(h_1, ..., h_k) = (u_1, ..., u_k)$

Step 5. Calculate the range of the radius of discrete Gaussian sampling $(r)$:

$$\sqrt{q} \times \omega(\sqrt{\log n}) \leq r \leq \frac{\varepsilon}{8mn^2\beta}$$

Step 6. Verifier calculates $\sigma_1$: $\sigma_{1,i} = h_i^T[y - U \times (\frac{1}{w})]$.

Step 7. Calculate $\sigma_a$ by a random number $(e)$, $\sigma_a = (a\sigma_1 + e) mod q'$

Step 8. Calculate $v'$, $v$ and session key $(sk)$:

$$v' = \sigma_a mod \beta$$
$$v = \frac{v'g}{\beta}$$
$$sk = \frac{\sigma_a}{\beta}$$

Step 9. Send message $(hp, v)$ to prover

Step 10. Prover calculates $sk$: 

Thus, $s_{k-s}$ can be: 

$$s_{k-s} = \left\{ \begin{array}{ll}
\sum_{i=1}^{n} i \prod_{i=1,j\neq i}^{n} \frac{j}{j-1} |p|, & t_d = TTL \\
\sum_{i=1}^{n-x} i \prod_{i=1,j\neq i}^{n-x} \frac{j}{j-1} |p|, & t_d > TTL
\end{array} \right.$$
\[ sk = \left( \frac{\sigma \alpha}{\beta} - \frac{(v + 1/2)}{g} \right) \mod m \]

Especially, based on the key consensus’ correctness, both the prover and the verifier need to receive the same session key.

And you can review the whole process through the Fig. 3.

Fig. 3 The process of PAKE [15]

Study 3. Post Quantum Public and Private Key Cryptography Optimized for IoT Security

The aim of this scheme is making the cryptography algorithms have a better adaptation to 5G IoT environments. In order to encrypt data streams in an IoT environment which supports 5G, the scheme mainly uses Variable Length Encoding (VLE) to design two lightweight LWE-based algorithms. And we will explain the model from two aspects: symmetric key cryptography and asymmetric key cryptography [17].

3.5 Symmetric Key Cryptography (SymLoki-VLE)

Step 1. Setup

The setup parameters are illustrated in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>an integer which represents the required number of bits used to store ( l )</td>
</tr>
<tr>
<td>( l )</td>
<td>a positive number, and ( l = 2^k )</td>
</tr>
<tr>
<td>( q )</td>
<td>a positive prime number, and ( \frac{3^3}{4} \leq q \leq l )</td>
</tr>
<tr>
<td>( N )</td>
<td>a positive number</td>
</tr>
<tr>
<td>( M )</td>
<td>another positive number, and ( M &lt; N )</td>
</tr>
<tr>
<td>( G_\sigma )</td>
<td>a discrete Gaussian Distribution with parameter ( \sigma )</td>
</tr>
</tbody>
</table>

Step 2. KeyGen

Generate a secret key \( S \) with \( N \) elements, and its size is \((N \times k)\) bits

Step 3. Encryption

Send two ciphertext: \( C_1 \) whose size is \( N \), \( C_2 \) whose size is \( L \) (\( 2 \leq L \leq M \))

Signify message \( M \) (like \( M = 1 \)):

Encrypt bit “0” \((L - 1)\) times, and encrypt “1” at \( L^{th} \) position.
Then we need to encrypt those individual bits, and each bit “b” can be embedded in the \( C_2 \) array whose size is \( L \). In this case, elements in \( C_2 \) can be represented as \( C_{2i} = (\langle RCS(C_1, i - 1), S > + E_i + b \cdot \frac{q}{2}\rangle \mod q \). In it, \( 1 \leq i \leq L \), \( C_{2i} \) is the \( i^{th} \) element in \( C_2 \), and \( RCS \) is mentioned in preliminaries.

Step 4. Decryption

Get the individual encrypted bit “b” by checking the condition: \( b_i = bool((\langle C_{2i} - \langle RCS(C_1, i - 1), S >\rangle \mod q \geq \frac{q}{2}) \). In this equation, when we get the same decrypted bit “1” for all \( M \) positions, the encrypted bit can be represented as “1”; similarly, when we get the bit “0”, we should quit checking and the encrypted bit can be represented as “0”. You can understand it better by Fig. 4.

![Fig. 4 Symmetric Key algorithm working process [17]](image)

### 3.6 Asymmetric Key Cryptography (Loki-VLE)

Kaushik et al. [17] created it on the basis of Learning With Errors skills, and they combined it with the variable length encoding in Symmetric Key Cryptography.

Step 1. Setup

Choose \( k, l, q, N, M, G_\sigma \) and \( \sigma \), all their standards are same as those in Symmetric Key Cryptography.

Step 2. KeyGen

Generate a random public key polynomial \( A \) who has \( N \) different integers and whose size is \( (k \cdot N) \) bits. Then, we need to calculate key \( B_i = (\langle RCS(A, i - 1), S > + E_i \rangle \mod q \). After that, we can announce the public key as \( (A, B) \), and the secret key vector \( S \) will not change, at the same time, we should protect it as possible as we can.

Step 3. Encryption

Set the message bit as \( S \) and send it in \( M \) different times. Then, select a random integer \( v \) and two other vectors \( E1 \) whose size is \( N \), and \( E2 \) whose sizes is \( M \). After that, generate two ciphertexts \( C1 \) and \( C2 \) who have \( M \) elements and whose maximum size are \( (k \cdot M) \) bits. Then, signify message \( M = 1 \), encrypt bit “0” \( (L - 1) \) times or bit “1” \( M \) times and message \( M = 0 \), encrypt bit “0” \( M \) times or bit “1” \( (L - 1) \) times, in both situations, \( 2 \leq L \leq M \).

Then we need to encrypt those individual bits as:

\[
C_{1i} = (A_i \times v + E_{1i}) \mod q
\]

\[
C_{2i} = (B_i \times v + b \cdot \frac{q}{2} + E_{2i}) \mod q
\]
In mathematical equation (6) and (7), \(1 \leq i \leq L\); \(C_1\) and \(C_2\) are the \(i^{th}\) element in vector \(C_1\) and \(C_2\); \(E_1\) and \(E_2\) are the \(i^{th}\) elements in vectors \(E_1\) and \(E_2\); \(b\) is the bit which needs to be encrypted.

Step 4. Decryption

Now, we have ciphertexts \(C_1\) and \(C_2\) (both representing message \(M\), but with sequences of encrypted bits having varying lengths). After solving the LWE problems of all elements in the array, decrypt the ciphertexts \(C_1\) and \(C_2\) by checking the condition \(b_i = \text{bool}((C_2_i - <\text{RCS}(C_1, i - 1)_S>q \geq q)^2)\). In this mathematical equation, when we get the same decrypted bit “1” for all \(M\) positions, we should quit checking and the encrypted bit can be represented as “1”; when we have the bit “0”, the encrypted bit can be represented as “0”. You can review this process by Fig. 5.

![Fig. 5 Asymmetric Key algorithm working process](image)

However, based on the detailed analysis and explanation above, there is a common problem in these three schemes: the size of public key in these three schemes are quite a few short. As a result, in order to ensure the privacy and security level, size of corresponding private key is longer, and receivers may need more time to decrypt. From the author’s perspective, it is an effective way to modify the length of public key to balance the sizes of both keys.

Furthermore, compared with other two models, the first scheme is much better. The CAKM scheme is capable of protecting several sorts of security attacks, such as eavesdropping, identity theft, organized crime and so on. And the CAKM-IoT method also has a good performance on computational overhead, security level, latency, successful message rate, throughput and other factors.

### 4. Comparison

![Table 2 Comparison of Three Schemes](image)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Year</th>
<th>Ref</th>
<th>Evaluation Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAKM (Tamizhselvan)</td>
<td>2022</td>
<td>[12]</td>
<td>C1 C2 C3 C4 C5 C6 C7 C8 C9</td>
</tr>
<tr>
<td>PAKE (Zhao et al.)</td>
<td>2022</td>
<td>[15]</td>
<td>C1 C2 C3 C4 C5 C6 C7 C8 C9</td>
</tr>
</tbody>
</table>
Table 3 Factor Identifier

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Factor</th>
<th>Identifier</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Security Level</td>
<td>C6</td>
<td>Ciphertext Size</td>
</tr>
<tr>
<td>C2</td>
<td>Computing Time</td>
<td>C7</td>
<td>Computing Overhead</td>
</tr>
<tr>
<td>C3</td>
<td>Speed</td>
<td>C8</td>
<td>Communication Overhead</td>
</tr>
<tr>
<td>C4</td>
<td>Latency</td>
<td>C9</td>
<td>Transmission Overhead</td>
</tr>
<tr>
<td>C5</td>
<td>Key Size</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Table 2, it is clear that all these three algorithms mention about their own security evaluation and give corresponding evidences, and Table 3 gives the explicit factors of each identifier in Table 2. For efficiency, scientists explain it from different angles, such as speed, time, latency and so on. Except CAKM scheme, both of other two schemes provide proof of improvement about memory. At the same time, some of them explain the relevant overhead they need in details. Following tables describes comparisons from single factor.

4.1 Security Level Analysis

Table 4 Security Comparison

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Security Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAKM (Tamizhselvan)</td>
<td>60%-110%</td>
</tr>
<tr>
<td>PAKE (Zhao et al.)</td>
<td>$Adv_{PAKE}(k) \leq \frac{Q}{</td>
</tr>
<tr>
<td>SymLoki-VLE (Kaushik et al.)</td>
<td>——</td>
</tr>
<tr>
<td>Loki-VLE (Kaushik et al.)</td>
<td>——</td>
</tr>
</tbody>
</table>

In the CAKM scheme, the security level depends on IoT device quantity. To be specific, when there are only 200 machines, the security level is only nearly 60%. However, if we add the number of these device to 6 times (1200), the security value can increase by almost 110%. As a consequence, CAKM has a high security when the number of devices is big enough.

In the second scheme, we have the mathematical inequality: $Adv_{PAKE}(k) \leq \frac{Q}{|D|} + negl(k)$, in which $Adv_{PAKE}$ represents advantage of the adversary, and D is the specific area for password dictionary, and $Q(k)$ is the boundary number of the passwords tested by attackers online, and $k$ is a constant parameter which is not related to the safety. When the inequality holds, the IoT system will not be damaged by adversaries anymore.

In symmetric cryptography (SymLoki-VLE), the security is related to LWE hardness. And the detailed proof is provided in paper [17]. And the asymmetric Key Cryptography (Loki-VLE) is similar whose algorithm can be essentially divided into small problems on the basis of LWE. However, there is no known algorithm which is capable of deciphering it. Thus, we are allowed to consider that both of them are developed to evade attacks quantumly.

4.2 Computing Time Analysis

Table 5 Computational Time Comparison

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAKM (Tamizhselvan)</td>
<td>10ms-45ms</td>
</tr>
<tr>
<td>PAKE (Zhao et al.)</td>
<td>$O(kmn)$</td>
</tr>
</tbody>
</table>
SymLoki-VLE (Kaushik et al.) & 0.23s \\
Loki-VLE (Kaushik et al.) & 0.18s \\

In CAKM scheme, computational time is closely related to the number of IoT devices, and the more devices there are, the longer the time will be. Compared with some previous and well-known methods, like Bi-GISIS, HCCPK and ESKA, the time cost of CAKM is significantly reduced.

For PAKE scheme, actually, there is not a precise computing time in the paper, but [15] provide the time efficiency (time complexity) in the paper. According to their records, for verifiers, their computational time complexity is $O(kmn)$, with prover $O(mn)$. In this equation, $k$ is the length of message authentication code; $m$ is the number of messages, and $n$ is a random integer. It is obvious that the time efficiency of provers is lower than that in other algorithms.

According to Kaushik et al. [17], SymLoki-VLE finishes the key generation almost in a split second (about 0.00006 seconds), with encryption spending just 0.118 seconds and decryption just 0.108 seconds. In contrast, Loki-VLE spends more than a half of its time (approximately 0.128 seconds) on decryption, and it only take 0.043 seconds and 0.005 seconds to generate keys and encrypt respectively.

### 4.3 Key Size Analysis

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Key Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAKM (Tamizhselvan et al.)</td>
<td>80 bits</td>
</tr>
<tr>
<td>PAKE (Zhao et al.)</td>
<td>$k$ bits ($0 \leq k \leq m - 1$)</td>
</tr>
<tr>
<td>SymLoki-VLE (Kaushik et al.)</td>
<td>1050 bytes</td>
</tr>
<tr>
<td>Loki-VLE (Kaushik et al.)</td>
<td>700 bytes</td>
</tr>
</tbody>
</table>

In the first scheme, the author does not give a precise value or relevant range, but he sets the key size as 80 bits as a value in testing environment; thus, we can infer that “80 bits” is an appropriate and average value in CAKM scheme.

For PAKE scheme, the size of the key is bigger than 0 and smaller than $(m - 1)$, actually, it is a random number in this range where $m$ represents the quantity of messages.

Although it seems that SymLoki-VLE and Loki-VLE require much more memory to save the key, it should be noted that the four algorithms are in different conditions. And if we import other algorithms and compare them with the last two schemes, it is clear to see the advantages of SymLoki-VLE and Loki-VLE. And Fig. 6 describes the difference visually.

![Fig. 6 Key size comparison among four schemes](image)

5. Conclusion

In this paper, we firstly introduce the background and development of IoT domain, and its advantages and convenience are also mentioned. Besides, it also provides some fundamental and professional concepts to make it easy to understand the follow-up analysis. The keynote of the article
is reviewing three previous schemes which are mainly used to resist common attacks by displaying their algorithms and complete process. According to the detailed analysis of these three schemes, it gives us realization that ensuring the security of a model is the most crucial. Under this condition, we should try our best to reduce memory cost and computational cost, as well as improve the efficiency. To achieve this goal, it is a good way to combine different algorithms into one scheme. For instance, we have already known that the Asymmetric Key Cryptography (Loki-VLE) has lower computing time overhead, thus we can try to apply it with CAKM model which has high security level. In the future, it is our common obligation to propose more advanced and cost-effective schemes to protect the IoT security.

References


599