electronics

MDPI

Article

Secure Device-to-Device Communication in IoT: Fuzzy Identity from Wireless Channel State Information for Identity-Based Encryption

Bo Zhang [1,2,*], Tao Zhang [1], Zesheng Xi [1], Ping Chen [3], Jin Wei [3,4] and Yu Liu [3,4]

[1] State Grid Smart Grid Research Institute Co., Ltd., Beijing 102201, China; zhangtao@geiri.sgcc.com.cn (T.Z.); xizesheng@geiri.sgcc.com.cn (Z.X.)
[2] School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[3] Institute of BigData, Fudan University, Shanghai 200437, China; pchen@fudan.edu.cn (P.C.); jwei17@fudan.edu.cn (J.W.); yuliu21@m.fudan.edu.cn (Y.L.)
[4] School of Computer Science, Fudan University, Shanghai 200433, China
* Correspondence: zhangbo@geiri.sgcc.com.cn

Abstract: With the rapid development of the Internet of Things (IoT), ensuring secure communication between devices has become a crucial challenge. This paper proposes a novel secure communication solution by extracting wireless channel state information (CSI) features from IoT devices to generate a device identity. Due to the instability of the wireless channel, the CSI features are fuzzy and time-varying; thus, we a employ locally sensitive hashing (LSH) algorithm to ensure the stability of the generated identity in a dynamically changing wireless channel environment. Furthermore, zero-knowledge proofs are utilized to guarantee the authenticity and effectiveness of the generated identity. Finally, the identity generated using the aforementioned approach is integrated into an IBE communication scheme, which involves the fuzzy extraction of channel state information from IoT devices, stable identity extraction for fuzzy IoT devices using LSH, and the use of zero-knowledge proofs to ensure the authenticity of the generated identity. This identity is then employed as the identity information in identity-based encryption (IBE), constructing the device's public key for achieving confidential communication between devices.

Keywords: Internet of Things; wireless channel state information; fuzzy identity; locally sensitive hashing; identity-based encryption; secure communication

Citation: Zhang, B.; Zhang, T.; Xi, Z.; Chen, P.; Wei, J.; Liu, Y. Secure Device-to-Device Communication in IoT: Fuzzy Identity from Wireless Channel State Information for Identity-Based Encryption. *Electronics* **2024**, *13*, 984. https://doi.org/10.3390/electronics13050984

Academic Editors: Lanting Fang and Yubo Song

Received: 4 January 2024
Revised: 30 January 2024
Accepted: 1 February 2024
Published: 5 March 2024

1. Introduction

With rapid technological advancements, the Internet of Things (IoT) is undergoing a remarkable and swift expansion, emerging as a key driving force in today's digital era. The rapid proliferation and application of the IoT span diverse domains, ranging from smart homes and industrial automation to healthcare and urban infrastructure. This technology's dynamic growth plays a pivotal role in enhancing the quality of life, fostering industrial innovation, and building intelligent societies. The convergence of sensors, embedded devices, and interconnected networks enables various physical objects to exchange real-time information, creating an extensive ecosystem of data. The widespread collection and analysis of these data provide businesses, governments, and individuals with deeper insights and the foundation for intelligent decision making. From the development of smart cities to advancements in precision agriculture, the IoT is profoundly transforming the ways we live and work.

Despite the rapid technological advancements that bring unprecedented connectivity and intelligence, security communication remains a significant challenge in the IoT. The use of traditional Public Key Infrastructure (PKI) systems, especially in large-scale deployments of IoT devices, inevitably faces the daunting task of establishing massive Public Key

Certification Centers. Managing the identity information of a vast number of devices not only introduces complexity and cost challenges but also carries the potential risks of performance bottlenecks and single points of failure.

Adopting an identity-based encryption (IBE) solution for establishing communication systems presents unique challenges, even though IBE can avoid the implementation of massive Public Key Certification Centers. Since the physical information of IoT devices often exhibits fuzziness, traditional IBE struggles to directly construct a stable device identity based on these ambiguous physical features. This challenge involves maintaining the consistency of the device identity in a dynamically changing environment to ensure the feasibility of secure communication. In such a scenario, the solution requires an innovative combination of physical information extraction and secure communication technologies to address the complexity and security requirements of IoT device identity management.

In addressing the aforementioned challenges, we introduced a novel device-to-device communication solution for the Internet of Things (IoT). Firstly, by extracting the wireless channel state information (CSI) from IoT devices, we successfully generated stable physical features for these devices. Leveraging the wireless communication characteristics among devices, we established a device identity. Subsequently, addressing the issue of the fuzzy nature of physical features potentially causing disruptions and affecting the stability of device identity, we introduced a locally sensitive hashing (LSH) algorithm. Through LSH, we were able to generate stable identity representations for the fuzzy physical features. This process not only ensures the consistency of the device identity in dynamic environments but also provides a reliable foundation for establishing secure communication. Finally, we applied this stable identity representation to the identity-based encryption (IBE) framework, constructing device public keys. Through this innovative integration, we achieved confidential communication between devices, presenting a new paradigm to address challenges faced by existing Public Key Infrastructure (PKI) and traditional IBE solutions.

Our contributions are as follows:

1. We propose a novel device-to-device communication solution for the Internet of Things (IoT). By extracting the wireless channel state information from IoT devices, we successfully generated stable physical features for these devices and combined them with IBE to implement secret communication based on CSI.
2. We propose a method to generate a fixed identity identifier from the fuzzy device CSI. This method allows the generation of a unique identity identifier from a stable identity feature in a deterministic manner, which can be used as the device public key information in identity-based encryption (IBE).
3. We introduce an identity verification scheme based on zero-knowledge proofs. This scheme can prove that the identity information generated by the device is derived from its CSI without revealing the CSI itself, thus safeguarding the privacy of the device.

2. Related Work

2.1. Identity-Based Encryption

In the year 2001, Boneh and Franklin [1]; Sakai, Ohgishi, and Kasahara [2]; and Cocks [3], independently proposed three identity-based encryption (IBE) schemes, marking the initiation of a new era in identity-based cryptography research. Both the Boneh–Franklin and Sakai–Ohgishi–Kasahara schemes utilize pairings satisfying bilinear mappings. Boneh and Franklin, employing the random oracle model, demonstrated that their scheme achieves indistinguishability against an adaptive chosen identity and chosen ciphertext attacks (IND-ID-CCA). The widespread application of bilinear pairings in cryptographic scheme design was a result of their work, guiding the development of identity-based cryptography. Subsequent works have often adopted methods similar to the Boneh–Franklin method, utilizing pairings with bilinearity to construct IBE schemes. In 2005, Waters [4] designed an efficient IBE scheme that does not rely on a random oracle.

In recent years, many scholars have proposed various improvements to IBE. One major challenge in the real-world deployment of IBE is the key escrow problem. If the private key

generator (PKG) is malicious, it can decrypt all messages. Khaleda Afroaz [5] proposed an IBE scheme based on anonymous identities, utilizing ring signatures to address this issue. In this process, identities are signed with ring signatures generated by a ring authority (RA) to mask the actual identity. Private keys are obtained from a key generation authority (KGA), preventing the KGA from knowing which key corresponds to which recipient. In the application of IBE schemes, V. Veeresh et al. [6] comprehensively overviewed the applications of IBE in cloud computing. Van-Quang-Huy Nguyen et al. [7] proposed a private identity-based encryption scheme for key management. Y. Liu et al. [8] introduced a hierarchical identity-based encryption scheme for key distribution in wireless sensor networks, reducing the computational time and saving storage space compared to IBE-based key distribution for wireless sensor nodes.

2.2. Channel State Information

In recent years, scholars both domestically and internationally have proposed numerous wireless device authentication schemes based on physical layer fingerprints. X. Wan et al. [9] employed the Received Signal Strength Indicator (RSSI) technique for device identification and monitoring. To enhance the spatial resolution and improve the accuracy of attack detection, multiple antennas were deployed at various landmarks to collect richer RSSI data. A. Mahmood et al. [10] introduced a distributed wireless device authentication method based on a Channel Impulse Response (CIR), enhancing the system detection accuracy through multipoint sensing technology. Q. Wu et al. [11] utilized Radio Frequency (RF) fingerprints for wireless device authentication, employing a Recursive Neural Network (RNN) to autonomously learn RF fingerprint features without manual intervention.

The RSSI, RSS, and CIR can be used to describe the characteristics of the wireless channel, but they provide limited channel information at a single frequency point. In contrast, CSI contains both amplitude and phase information for each OFDM subcarrier, offering more detailed wireless channel features and achieving a superior identification performance. To address this, R. Liao et al. [12] proposed a CSI-based device authentication scheme, employing a CNN as the recognition algorithm for physical layer fingerprints. Meanwhile, C. Shi et al. [13] utilized an SVM to identify user channel fingerprints. Ribouh et al. [14] designed a CSI-based key generation method for use in vehicular environments, treating the CSI values of each subcarrier as random sources and using a new QAM demodulation quantizer (QAM-Dem-Quan) to extract bit values for key generation. Ji et al. [15] proposed a key generation scheme combining adaptive link selection and a two-step decorrelation algorithm, leveraging CSI information. Wang et al. [16] combined CSI features with the static position of devices, constructing CSI data into CSI images. Through a deep learning-based recognition and authentication model, they learned the mapping relationship between CSI and the device identity, achieving device authentication. Wang et al. [17] introduced a physical layer authentication scheme based on Gaussian Process channel prediction. By establishing a mapping between historical CSI information and the sender's location information, they predicted the next legitimate CSI information for identity recognition. Additionally, they proposed a one-class authentication (OCA) scheme that does not require any channel information from attackers.

2.3. Zero-Knowledge Proof

Many researchers are concerned about privacy issues in data sharing and have proposed various data sharing schemes with privacy protection features. Lu et al. [18] introduced a novel approach based on federated learning and deep reinforcement learning to alleviate the transmission overhead and address the privacy concerns of data providers. The asynchronous federated learning mechanism, learning models from edge data, minimizes the total cost by selecting participating nodes and further enhances the efficiency of federated learning. To validate the authenticity of collected data and protect user privacy, a study [19] employs homomorphic encryption technology to construct a ciphertext space, ensuring the normal operation of data services under data encryption. Based on this

ciphertext space, they propose an identity-based signature scheme and a two-tier batch signature verification scheme.

A zero-knowledge proof (ZKP) refers to the ability of a prover to convince a verifier of the correctness of a statement without revealing any useful information. This concept was introduced by S. Goldwasser and others [20]. The security of zero-knowledge proofs relies primarily on cryptographic assumptions related to challenging problems, such as discrete logarithms and elliptic curve problems. The zero-knowledge proof system is widely used for authentication because it has the following properties:

- Completeness: Referring to the assurance that anything valid generated by an honest prover can be successfully verified by an honest verifier.
- Soundness: Ensuring that for any prover without access to the secret, they cannot forge something that would pass verification.
- Honest verifier with zero knowledge: Ensuring that, for an honest verifier, apart from knowing the outcome of the proof (i.e., the known parameters mentioned above), no other information is revealed.

In recent years, ZKPs have been preliminarily applied and practiced in the context of IoT access security. Walshe et al. [21] proposed an IoT and sensor device authentication scheme based on non-interactive zero-knowledge proofs. Unlike traditional ZKP, the authors replaced the ZKP's NP-hard problem and used Merkle trees to create authentication challenges. The authors conducted simulations to evaluate the performance of non-interactive zero-knowledge proofs relative to traditional zero-knowledge proofs. Salleras et al. applied it to scenarios such as 5G services for user identity authentication [22]. Service providers can verify proofs and grant services without knowing the user's identity. Users can remain anonymous when using the service by proving the ownership of their signature without revealing any other information.

Zero-knowledge proofs have been applied as a privacy protection technology in various fields, such as traffic management, the crowdsourced IoT, identity management schemes in blockchain, and vehicular ad hoc networks [23]. However, with the exponential growth of IoT devices in various applications, ensuring the efficient, secure, and bidirectional authentication of relevant devices without leaking any information remains crucial.

3. Wireless Channel State Feature Extraction

3.1. Multipath Effects

Wireless signals typically propagate in environments characterized by complex electromagnetic wave scattering. In such environments, various objects along the signal propagation path from the transmitting device to the receiving device can induce electromagnetic wave scattering. These objects cause the signal to reflect along different paths, eventually reaching the receiving device. Due to the presence of these reflected paths, the signal received at the receiving device is the result of the superposition of multiple reflections, a phenomenon commonly referred to as multipath effects.The origins of multipath effects are diverse, including outdoor structures, terrain such as mountains, and densely packed objects in indoor environments, all contributing to varied electromagnetic wave reflections and refractions.

As the transmitted signal traverses different electromagnetic wave reflection or refraction paths, differences in the amplitude, phase, and arrival time occur during the transmission process. At the antenna position of the receiving device, multiple reflections or refractions with different phases may simultaneously superimpose. Therefore, under the influence of multipath effects, the signal response amplitude received at the antenna of the receiving device exhibits significant variations. This phenomenon is known as multipath fading.

3.2. Orthogonal Frequency Division Multiplexing

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier modulation technique widely employed in the field of wireless communication, particularly in high-speed data transmission environments. The core concept of OFDM involves dividing a high-speed data stream into multiple lower-speed substreams and concurrently transmitting these substreams on multiple carriers to enhance the overall data transmission efficiency.

OFDM subdivides the high-speed data stream into several lower-speed substreams and allocates these substreams to different orthogonal subcarriers. This allocation ensures mutual orthogonality among the subcarriers, reducing interference between spectral bands. Due to the orthogonality of the subcarriers, symbols on each subcarrier can be transmitted at different phases, achieving higher spectral efficiency within the same frequency band. This adaptability makes OFDM suitable for the demands of high-speed data transmission.

Assuming a carrier bandwidth of B, it can be divided into N orthogonal subcarriers with a bandwidth of $\Delta f = B/N$. If the center frequency of the first carrier is f_0, then the frequency of the nth carrier is given by the following:

$$f_n = f_0 + (n-1)\Delta f \tag{1}$$

Modulating the OFDM symbol P_n onto the subcarrier n results in the transmission symbol $P_n^{e^{j2\pi f_n t}}$. Summing up the signals across all N subcarriers yields the final transmitted signal:

$$f(t) = \sum_{n=1}^{N} P_n e^{j2\pi f_n t} = e^{j2\pi f_0 t} \sum_{n=1}^{N} P_n e^{j2\pi(n-1)\Delta f t} \tag{2}$$

Once the receiver obtains the signal, it can determine the OFDM symbol P_n transmitted over the subcarrier using the following formula:

$$\Delta f \int_0^{\frac{1}{\Delta f}} f(t) e^{-j2\pi f_n t} dt = P_n + \sum_{k \neq n} \Delta f \int_0^{\frac{1}{\Delta f}} P_k e^{j2\pi f_k t} e^{-j2\pi f_n t} dt = P_n \tag{3}$$

3.3. Channel State Information

In the same physical environment, different subcarriers experience varying fading and multipath effects in the wireless channel. The channel responses at different frequency positions of the subcarriers exhibit differences. To describe the subchannel response at each subcarrier frequency position, a set of data representing the features of the channel frequency response is required. These data enable the communication system at the receiving end to adapt to the actual conditions of the wireless channel, allowing the reconstruction of the effective original wireless signal received by the receiving antenna. This series of data describing the channel frequency response is commonly referred to as channel state information (CSI). CSI is a crucial data metric in wireless communication, encompassing various aspects of the radio waves between the terminals and enabling the real-time dataization and visualization of the electromagnetic wave status between the communicating parties.

The receiving device can recover the original signal based on the channel conditions described by CSI, facilitating signal transmission under the adaptation to the current channel conditions. This is vital for achieving robust communication in OFDM systems. The CSI is obtained through channel estimation in OFDM wireless communication systems. In OFDM communication transmission systems, the primary purpose of channel estimation is the selection of pilot information. As wireless channels may experience fading during transmission, continuous tracking of the channel is necessary, requiring the ongoing transmission of pilot information. Through the estimation of pilot information, the receiving terminal can acquire the CSI of the wireless communication channel from the other party.

CSI reflects the communication channel conditions between the transmitter and the receiver, manifesting on OFDM symbols as variations in the amplitude and phase of signals

on each subcarrier. According to the signal system model, let H represent the channel frequency response, X be the transmitted signal, and N denote noise. The received signal Y can be expressed as follows:

$$Y = XH + N \tag{4}$$

In WiFi, OFDM technology is commonly employed, and CSI reflects the representation of the channel frequency response (CFR) on subcarriers. The expression of the CFR can be easily obtained through the pilot portion of each OFDM symbol:

$$H = \sum_{k \in K} \|h_k\| \times e^{-j\varphi_k} \tag{5}$$

In this context, $\|h_k\|$ and φ_k represent the amplitude and phase values of the signal on each subcarrier, respectively. The CFR can describe small-scale multipath effects and is widely utilized as a channel indicator. By adjusting the firmware, it is possible to obtain the CFR of multipath channel samples, i.e., the measured values of CSI, on existing wireless terminal devices, such as commercial IEEE 802.11a/g/n equipment.

CSI represents the sampling of the wireless channel frequency response at various subcarrier frequencies in OFDM. On each wireless channel, each antenna at the receiving end can generate a set of CSI data. Due to the presence of multipath effects and fading phenomena in the wireless channel environment, different subcarrier frequencies are affected to varying degrees. As a result, each subchannel possesses a unique gain. This implies that each subchannel has corresponding CSI values, reflecting its distinctive frequency response features. The CSI features of the wireless channel exhibit robustness, uniqueness, and irreproducibility.

3.4. Wireless Channel Features

The wireless channel possesses four features, reciprocity, time variability, short-term stationarity, and spatial sensitivity, which are descrived below:

1. Channel reciprocity: Channel reciprocity in a wireless channel refers to the symmetric propagation features between the transmitter and the receiver. In other words, if the propagation channel from the transmitter to the receiver is reciprocal, then the propagation channel from the receiver to the transmitter is also reciprocal. Reciprocity simplifies the analysis and modeling of wireless communication systems. However, in practical communication environments, various factors such as noise, channel variations, and environmental conditions may lead to incomplete consistency in the collected channel state data due to the asynchrony in signal reception times between communication parties.
2. Time variability: The propagation features of a wireless channel undergo changes over time. This time variability can be caused by various factors, including the movement of objects, changes in obstacles along the signal path, and variations in atmospheric conditions. Time variability is particularly crucial for mobile communication systems as it impacts the signal transmission quality and system performance.
3. Short-term stationarity: Despite the time variability of the wireless channel, it is often possible to approximate the channel as approximately stationary over short periods. This implies that within very short time intervals, the propagation features of the channel can be considered constant, simplifying the complexity of system design and signal processing.
4. Spatial sensitivity: The propagation features of a wireless channel are highly sensitive to spatial changes. Even within relatively small spatial ranges, significant variations in the channel's features may occur. This sensitivity is due to the signal's propagation through multiple paths, influenced by reflection, refraction, and scattering. Spatial sensitivity is particularly important in places with complex structures, such as indoor and urban environments, necessitating the adoption of appropriate antenna configurations and signal processing techniques to address it.

3.5. CSI Acquisition Module Based on Multipath Effects

3.5.1. CSI Acquisition

In general, CSI is often represented by the channel frequency response (CFR) in a wireless multipath channel environment. When the transmission frequency is denoted as f, the formula for CSI is given by the following:

$$H(f) = \sum_{n}^{N} a_n e^{-j2\pi f \tau_n} \tag{6}$$

In OFDM wireless networks, such as IEEE 802.11a/g/n, each pair of antenna channel propagations results in a set of CSI data. The length of each set of CSI data corresponds to the number of subcarriers, and the individual CSI values within the set are represented by the following formula:

$$CSI_{m,n} = [csi_{-i}, csi_{-i+1}, \ldots, csi_0, csi_1, \ldots csi_i] \tag{7}$$

In the above formula, m represents the index of the transmitting antenna, n denotes the index of the receiving antenna, and i represents the subcarrier index. When a device transmits data with a bandwidth of 20 MHz in the 2.4 GHz frequency band, the utilized channel will consist of 64 subcarriers. The measured CSI at the receiver will include the frequency response for each subcarrier, forming a 64-dimensional complex CSI vector. In a Multiple-Input Multiple-Output Orthogonal Frequency Division Multiplexing (MIMO-OFDM) wireless communication systems (assuming the system has V transmit antennas and G receive antennas), the CSI will be expanded into a three-dimensional complex matrix of size $V \times G \times 64$. This provides an alternative representation for the CSI:

$$\mathrm{CSI}_k = |csi_k| e^{-j\angle csi_k}, k = -i, \ldots, -1, 1, \ldots, i^{\circ} \tag{8}$$

where $|csi_k|$ represents the amplitude and $\angle csi_k$ represents the phase.

The majority of CSI amplitudes tend to stabilize within a relatively concentrated range, with only a portion of the data exhibiting discrete deviations. However, CSI phase data are characterized by an unstable pattern, displaying a distribution with random rotations. Additionally, significant discrepancies are observed in the CSI data between the receiving and transmitting ends at the same position.

3.5.2. Preprocessing of CSI

CSI data may have certain errors due to environmental noise. To eliminate these errors, preprocessing is applied. First, due to hardware errors, environmental factors, etc., there may be outliers or anomalous points in the data sequence. To reduce the impact of errors and enhance the robustness, an outlier detection method is employed. Secondly, normalization is used to reduce data dispersion. Finally, the samples undergo smoothing to eliminate the effects caused by environmental noise.

1. Outlier Handling

After obtaining CSI', the n samples are denoted as $^{-}\mathrm{CSI}'_n = c_n^1, c_n^2, \cdots, c_n^M (n = 1, 2, \cdots, N)$, where N represents the number of collected samples and M represents the number of available subcarriers collected (52 in this case). Let $L_m = \left(c_1^m, c_2^m, \cdots c_N^m\right)$ represent the average value of each subcarrier across all samples, where $m = 1, 2, \cdots, M$. The median window chosen in this paper is $C_{D,n}^I$ and the sample values within a range of $2l$ around it. The standard deviation for all samples is derived from the absolute deviation, as shown in the following formula:

$$\sigma_I^m = \frac{1}{\gamma} \mathrm{median}(|c_i^m - c_I^m|) \tag{9}$$

where c_I^m represents the median of the window, c_i^m represents the samples within the window, and $\gamma = \sqrt{2}\,\mathrm{erfinv}(0.5)$. The removal of outliers in the samples is processed using the following formula:

$$c_i^m = \begin{cases} c_i^m & \left|c_i^m - c_I^m\right| \le \eta\sigma_I^m \\ c_I^m & \left|c_i^m - c_I^m\right| > \eta\sigma_I^m \end{cases} \tag{10}$$

2. Normalization

Due to environmental interference, the amplitude of CSI may exhibit certain variations over time, but, overall, it will have similar fluctuation trends. To reduce the adverse effects of amplitude fluctuations, the amplitude sequence of CSI is normalized using the following formula:

$$c_i = \frac{a_i - a_{\min}}{a_{\max} - a_{\min}} \tag{11}$$

After normalization, the amplitude sequence of CSI exhibits a smaller degree of dispersion while preserving the original shape information to a certain extent.

3. Smoothing

The smoothing process involves working solely on the amplitude values of each subcarrier, aiming to reduce temporal influences. The smoothing method is expressed as follows:

$$\tilde{c}_n^m = \frac{1}{w} \sum_{\max\left(0, n - \left\lfloor \frac{w}{2} \right\rfloor\right)}^{\min\left(N, n + \left\lfloor \frac{w-1}{2} \right\rfloor\right)} c_n^{m_0} \tag{12}$$

where w represents the length of the smoothing window.

4. Non-Interactive Channel Feature Similarity Comparison

4.1. Zero-Knowledge Proof

A zero-knowledge proof (ZKP) refers to the ability of a prover to convince a verifier of the correctness of a statement without revealing any useful information. This concept was introduced in the early 1980s by S. Goldwasser and others. The security of zero-knowledge proofs relies primarily on cryptographic assumptions related to challenging problems, such as discrete logarithms and elliptic curve problems. This paper focuses on the enhancement of Sigma Protocols in zero-knowledge proofs, grounded in the assumption of the difficulty of the discrete logarithm problem. The objective is to tailor Sigma Protocols to meet the specific requirements of certain industry applications while maintaining the security principles inherent in zero-knowledge proofs.

The Sigma Protocol is an interactive proof system consisting of a prover (P) and a verifier (V) in a three-step process. This system is based on the discrete logarithm problem and a commitment key (ck) generated with a security parameter λ. The commitment key ck is derived from the discrete logarithm problem and a random number $r \in \mathbb{Z}_q^*$, where represents a non-negative integer ring of order q. The prover can generate a commitment $Com_{ck}(c; r)$ for a secret c, representing a declaration of the secret c. Based on prior research results, it is known that, with negligible probability, this commitment uniquely binds to the secret c and does not leak information about c.

The Sigma Protocol process is described as follows:

1. The prover initiates the process by sending the commitment regarding the secret c to the verifier.
2. Upon receiving the commitment, the verifier randomly selects a challenge value $x \in \mathbb{Z}_q^*$ and sends it to the prover.
3. Based on the received challenge value x, the prover generates the corresponding response value z and sends it to the verifier.
4. These three steps constitute the proof by the prover for the secret c. With the known parameters, including the commitment key ck, commitment value $Com_{ck}(c; r)$, chal-

lenge value x, and response z, the verifier is capable of validating the legitimacy of the proof.

The intersection proof proposed in this paper is a modification of the aforementioned Sigma Protocol, based on the Pedersen commitment scheme. The properties of this commitment scheme are defined as follows:

Let $\mathbb{G}$ be a q-order cyclic group generated by the commitment key ck, with generators g and h. Given a random number $r \in \mathbb{Z}_q^*$, a Pedersen commitment based on the message c can be generated as follows:

$$Com_{ck}(c; r) = g^c \cdot h^r \in \mathbb{G} \tag{13}$$

The Pedersen commitment is a homomorphic commitment scheme that, under the discrete logarithm assumption, possesses two essential properties, hiding and binding:

- Concealment (hiding): For any adversary with a probabilistic polynomial time computational capability, if the adversary is unable to effectively distinguish between the commitments $Com_{ck}(c_0; r_0)$ and $Com_{ck}(c_1; r_1)$ corresponding to two distinct messages c_0 and c_1, then the commitment scheme is considered to possess concealment.
- Binding: For any adversary with a probabilistic polynomial time computational capability, given a known commitment $Com_{ck}(c_0; r_0)$, the task is to find another secret value $c_!$ such that their commitment values are equal, i.e., $Com_{ck}(c_0; r_0) = Com_{ck}(c_1; r_1)$. If the probability of successfully achieving this task is negligible, then the commitment scheme is deemed to exhibit binding characteristics.

4.2. Interactive Similarity Matching

Similar to fingerprint recognition, when collecting fingerprints from the same individual multiple times, various factors such as sweat, pressure variations, and other elements on the fingers can result in subtle differences in the collected fingerprints. Ensuring fuzzy identity recognition while maintaining both accuracy and fault tolerance in the presence of these subtle variations is a crucial consideration.

In the process of identity authentication, despite the preprocessing of channel state information (CSI), deviations from standard values may still occur during the quantification process. Therefore, it is necessary to enhance the tolerance of identity recognition without compromising accuracy. In this context, we introduce the concept of fuzziness to assess the similarity of fuzzy equations. The extracted CSI exhibits fuzzy characteristics, with the CSI collected by both parties not being entirely consistent but demonstrating a high degree of similarity. This paper, by comparing the similarity of CSI from both parties and obtaining the intersection, negotiates a shared channel key for communication. Simultaneously, the successful comparison of CSI similarity also signifies that the remote end possesses CSI highly similar to the local end, enabling dual-end authentication.

This paper employs a zero-knowledge proof based on the Sigma Protocol for conducting a Comparative Similarity Index (CSI) matching. Initially, the collected CSI information is quantified into a binary CSI vector. This binary vector serves as a representation of the CSI, and the similarity matching involves comparing the binary vectors held by both parties, leading to the determination of their binary vector intersection. Subsequently, by constructing an n-th order polynomial to express the binary vector, the similarity matching of binary vectors is transformed into the intersection operation of an n-th order polynomial. The intersection calculation is achieved by verifying the polynomial's zero points. This process substantiates the high degree of similarity between the binary vectors held by both parties, thereby facilitating the mutual authentication of the communicating entities.

For a binary string s owned by the communication party Alice, it is divided into n segments of a fixed length, represented as $(s_1, \ldots, s_n)$. The prover wishes to convince the verifier of possessing the secret vector $s = (s_1, \ldots, s_n)$ without disclosing any information. Similarly, after preprocessing, Bob obtains the set $s = (s'_1, \ldots, s'_n)$.

In the first place, we construct an n-degree polynomial $p(x) = (x - 1 \cdot s_1), \ldots, (x - n \cdot s_n) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$. Clearly, the polynomial evaluates to 0 only when the input $x = i \cdot s_i (i \in [1, n])$. This property can be utilized for a prover to verify their possession of secrets without disclosing these secret values (including their corresponding values in the group field) to others. Based on this idea, the problem of similarity comparison can be transformed into a zero-knowledge similarity comparison problem: the prover needs to prove that they possess a polynomial $p(x)$, and the zeros of this polynomial can be verified by the verifier.

The similarity comparison relation is defined as follows:

$$R = \left\{ \begin{array}{cc} ck \leftarrow \mathrm{Setup}(1^\lambda); P_i \in \mathbb{G}; a_i, r_i, x \in \mathbb{Z}_q^* & \sum_{i=1}^{n} P_i^{(s^i)} = \mathrm{Com}_{ck}(0; r') \\ P_i = \mathrm{Com}_{ck}(a_i; r_i)(i = 0, \ldots, n) & \text{where } s \in 1 \cdot s_1, \ldots, n \cdot s_n \end{array} \right\} \quad (14)$$

Firstly, we define the variables required for the algorithmic flow description, based on the discrete logarithm problem. Let $\mathbb{G}$ be a cyclic group of order q, where q is a prime number. The generators of this group are denoted as $\mathbb{g}$ and $\mathbb{h}$, both belonging to $\mathbb{G}$. For an element x in the multiplicative group of q-order positive integers $\mathbb{Z}_q^*$, it holds that $\mathbb{g}^x \in \mathbb{G}$. Boldface notation such as $\boldsymbol{a}$ is used to represent vectors, for instance, $\boldsymbol{a} = (a_0, \ldots, a_n)$ denotes a vector composed of n numerical values. The prover can generate a corresponding Pederson commitment for their secret $\boldsymbol{a}$, denoted as $Com(\boldsymbol{a}; r)$, which can be succinctly expressed as $Com(\boldsymbol{a}; r) = \mathbb{g}^{\boldsymbol{a}} \cdot \mathbb{h}^r$.

The prover initially generates, for each secret value a_i, corresponding random numbers r_i, u_i, and v_i. The commitments $A_i = Com_{ck}(a_i; u_i)$ and $R_i = Com_{ck}(r_i; v_i)$ are computed and disclosed to the verifier. Subsequently, the verifier generates a random number x as the challenge value for the aforementioned commitments. The prover, in response to the verifier's challenge x, computes $f_i = a_i \cdot x + r_i$ and $z_i = u_i \cdot x + v_i$. Finally, the verifier performs validation on the proof, checking if the prover possesses the secret vector $\boldsymbol{a} = (a_0, \ldots, a_n)$ and confirming whether the subchannel characteristics are mutual. If the result corresponds to a commitment of 0 in the form $Com_{ck}(0; u')$, then the currently computed value s_j' is considered a shared secret between the communicating parties and is utilized in generating the final channel key.

Channel Key Computation

After the application of the feature extraction algorithm to the CSI information, a set of binary sequences is obtained. It is ensured that a significant portion of the binary sequences held by both Alice and Bob exhibit similarity, as illustrated in Figure 1.

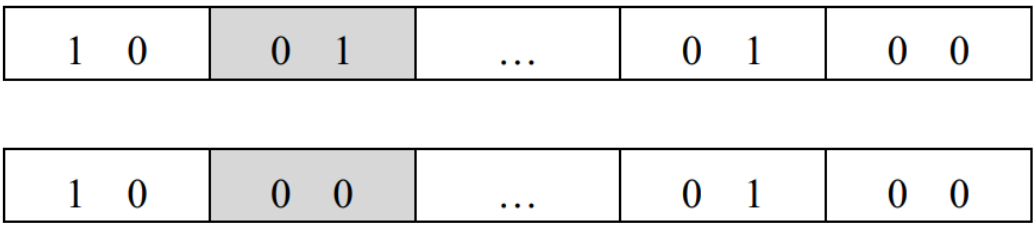


Figure 1. Quantized and segmented binary feature sequences.

In the proposed scheme, the sequence is divided into segments, and each segment is treated as a secret value s_i for verification purposes. The verifier only needs to sequentially compute the verification for each segmented secret value s_j'. If the equality holds, it can be inferred that the corresponding segments in the binary sequences of the communicating parties are identical, thereby establishing a common element. Ultimately, the verifier obtains the intersection of the secret value vectors $\boldsymbol{s}$ and $\boldsymbol{s}'$, denoted as $\boldsymbol{s} \cap \boldsymbol{s}'$. This intersection sequence serves as the shared channel key for the mutual authentication in the key negotiation process.

4.3. Non-Interactive Key Similarity Comparison

The aforementioned similarity comparison is observed to be interactive. In this context, apart from the commitment value initially transmitted, a challenge–response process must be completed between the prover and the verifier to execute the entire procedure. The interactive nature of this process may result in an increased execution time and potential resource waste or even incompatibility when deployed in certain protocols. This paper leverages the Fiat–Shamir transformation to convert the aforementioned three-step interactive similarity comparison into a non-interactive process requiring only one step. The principle of this transformation involves utilizing the hash value of the interactive data as the challenge for the second step of the Sigma Protocol and using this challenge value to compute the response in the third step. Finally, the commitment and response are sent together to the verifier.

Upon applying the Fiat–Shamir transformation, the non-interactive similarity comparison process is outlined as follows:

1. In the execution of the aforementioned Protocol 1, the prover randomly generates three sets of random vectors $(r_0, \ldots, r_n)$, $(u_0, \ldots, u_n)$, and $(v_0, \ldots, v_n)$, and produces the corresponding commitment values $(A_0, \ldots, A_n, R_0, \ldots, R_n)$.
2. The prover generates a challenge value for the data to be sent, where $x = H(ck, A_0, \ldots, A_n, R_0, \ldots, R_n)$ and $H(\cdot)$ denotes a hash function.
3. In accordance with the challenge value x, the secret vector $\boldsymbol{a} = (a_0, \ldots, a_n)$, random value vector $\boldsymbol{r} = (r_0, \ldots, r_n)$, $\boldsymbol{u} = (u_0, \ldots, u_n)$, and $\boldsymbol{v} = (v_0, \ldots, v_n)$. The prover computes the corresponding response vectors $\boldsymbol{f} = (f_0, \ldots, f_{n-1})$ and $\boldsymbol{z} = (z_0, \ldots, z_{n-1})$.
4. The final prover will transmit the proof, denoted as $Proof = (x, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{f}, \boldsymbol{z})$, to the verifier in a single instance.

5. Implementation

5.1. Locality-Sensitive Hashing

We employ the locality-sensitive hashing (LSH) algorithm to process bit sequences. LSH is an algorithm designed for similarity search in massive datasets. Essentially, LSH is a technique for data dimensionality reduction, aiming to map data points from the original high-dimensional space to a lower-dimensional space while attempting to preserve their similarity.

The LSH algorithm consists of two main parts. In the first part, we utilize the SimHash algorithm for data dimensionality reduction. It involves merging long bit sequences through weighted combination, merging, and dimensionality reduction steps, resulting in shorter bit strings. In the second part, we apply LSH to locally search for matching pairs among the short bit strings. The bit strings are hashed into hash buckets after being split, ultimately generating a unique identity for terminal devices.

The SimHash algorithm is primarily employed for text similarity detection, using Hamming distance to calculate similarity, making it suitable for the dimensionality reduction of long bit sequences. In the classical SimHash algorithm, text needs to be hashed and transformed into a binary string of 0s and 1s that can be measured using Hamming distance. The m = 6 SimHash algorithm is illustrated in the accompanying Figure 2.

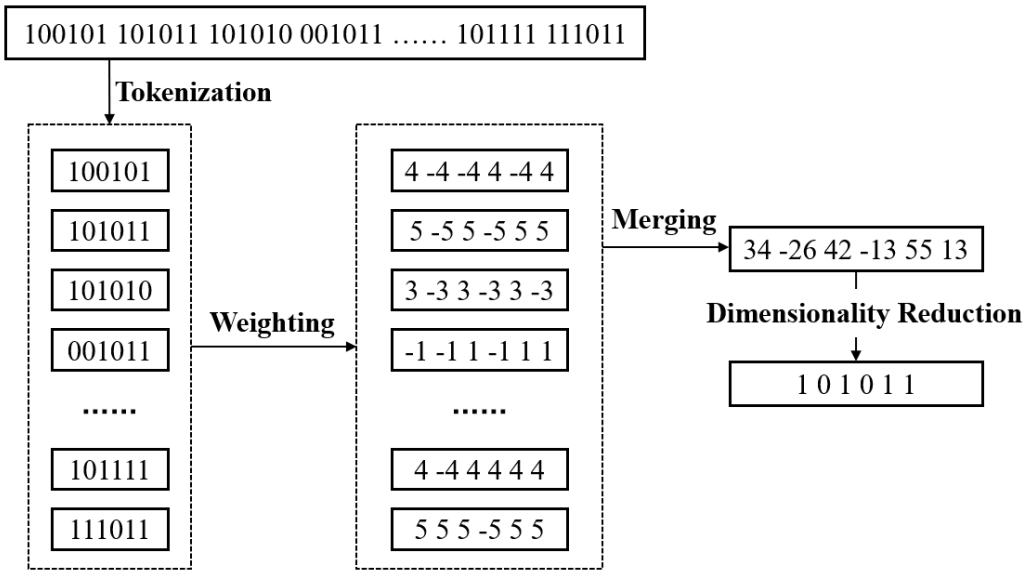


Figure 2. m = 6 SimHash algorithm.

The SimHash algorithm employed in this experiment is primarily divided into the following components:

1. Tokenization: The long bit sequence is segmented into multiple equidistant feature strings, each with a length of m bits.
2. Weighting: Each equidistant feature string is assigned a weight, based on its importance. Additionally, each bit in the feature string, whether 1 or 0, is transformed. For instance, for a feature string of length 6 bits, the assigned weight is 4, and the weighted representation becomes $4\ -4\ -4\ 4\ -4\ 4$.
3. Merging: The weighted feature strings are cumulatively added to obtain a final sequence string.
4. Dimensionality reduction: The merged sequence string undergoes dimensionality reduction. If a particular position in the sequence string is greater than 0, it is set to 1; otherwise, it is set to 0. The output is the signature corresponding to the long bit sequence. Multiple signatures are combined to produce the SimHash signature matrix.

After obtaining the SimHash signature, the locality-sensitive hashing (LSH) algorithm is employed to hash the signature matrix, aggregating the results into hash buckets. This facilitates local searching for matching pairs, with the hash bucket signature serving as the final identifier. During the hashing process, the algorithm further reduces the dimensionality of the input data, enabling unique and stable identity outputs for inputs with minor variations. The locality-sensitive hashing algorithm is depicted in Figure 3.

The specific steps of the locality-sensitive hashing (LSH) algorithm are as follows:

1. Initially, the signatures in the previously obtained signature matrix are divided into different bands, with each band containing a fixed number of rows.
2. Each band is hashed into a distinct hash bucket.
3. Identification is applied to each hash bucket, combining the identifiers of the buckets where each segment of the signature resides. The final result represents the output of the LSH algorithm.

These steps facilitate the segmentation of signatures into bands, followed by the hashing of each band into separate hash buckets. Subsequently, the identification of each hash bucket is performed, combining the identifiers of the buckets associated with each segment of the signature. The ultimate outcome constitutes the output of the LSH algorithm.

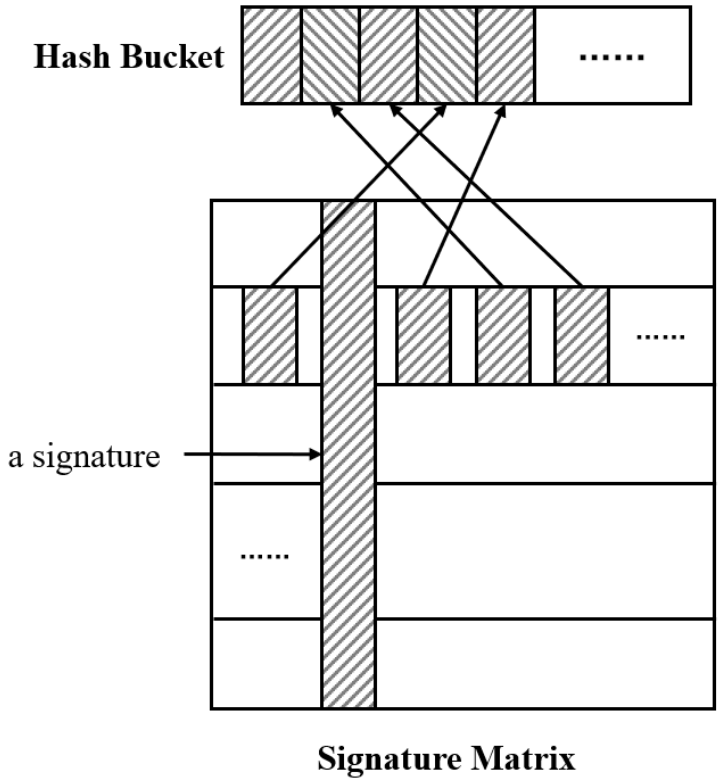


Figure 3. The locality-sensitive hashing algorithm.

5.2. CSI-Based IBE for IoT

We let k be the security parameter given to the setup algorithm.

Setup: (1) Choose a large prime p for some prime $q > 3$. Let E be the elliptic curve defined by $\cdot y^2 = x^3 + 1\cdot$ over $\cdot \mathbb{F}_p$. Choose an arbitrary $P \in E/\mathbb{F}_p\cdot$ of order q. (2) Pick a random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. (3) Choose a cryptographic hash function $H : \mathbb{F}_{p^2} \rightarrow \{0,1\}^n$ for some n. Choose a cryptographic hash function $G : \{0,1\}* \rightarrow \mathbb{F}_p$.

The message space is $M = 0,1^n$. The ciphertext space is $C = E/\mathbb{F}_p \times \{0,1\}^n$. The system parameters are $< p, n, P, P_{pub}, G, H >$. The master key is $s \in \mathbb{Z}_q^*$.

Extract: For a given string $ID \in \{0,1\}^*$, the algorithm builds a private key d as follows: (1) Map the ID to a point $Q_ID \in E/\mathbb{F}_p$ of order q. (2) Set the private key d_{ID} to be $d_{ID} = sQ_{ID}$ where s is the master key.

In the first step, we need to initially use the CSI Acquisition module in Section 3 to obtain the physical state information features of the device. Since in the actual environment these physical state information features are always fluctuating, it is necessary to use LSH to transform the steady-state features of the device into the same hash. Only in this way can they be used as identity identifiers in IBE, serving as public keys externally. In this process, it is also necessary to utilize the non-interactive channel feature similarity comparison in Section 4 to prove the effectiveness of mapping feature values to hash values through LSH and to maintain the privacy of the device's physical state features.

Encrypt: To encrypt $m \in M$ under the public key ID, take the following steps: map the ID into a point $Q_{ID} \in E/\mathbb{F}_p$ of order q. Then, choose a random $r \in \mathbb{Z}_q$, and set the ciphertext to be

$$C =< rP, m \oplus H(g_{ID}^r) > \tag{15}$$

where $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in \mathbb{F}_{p^2}$.

Decrypt: Let $C = (U, V)$ be a ciphertext encrypted using the public key ID. If $U \in E/\mathbb{F}_p$ is not a point of order q, reject the ciphertext. Otherwise, to decrypt C using the private key d_{ID}, compute the following:

$$V \oplus H(\hat{e}(d_{ID}.U)) = m \tag{16}$$

6. Implementation and Experiment

6.1. Construction of a Key Exchange Module Testing System

6.1.1. Testing Environment

The testing environment for channel key negotiation based on channel fingerprints was established with distinct communication characteristics, namely, line-of-sight (LOS) and non-line-of-sight (NLOS) characteristics, as illustrated in Figures 4 and 5, respectively.

1. In the NLOS scenario, as illustrated in Figure 4, the communication devices of the two parties are positioned at distances of 3 m, 4 m, and 5 m, with the attacker at the closest proximity of 0.1 m.
2. In the LOS scenario illustrated in Figure 5, the devices of the communicating parties are situated at distances of 25 cm, 50 cm, and 75 cm, with the attacker positioned at a minimum proximity of 0.1 m.

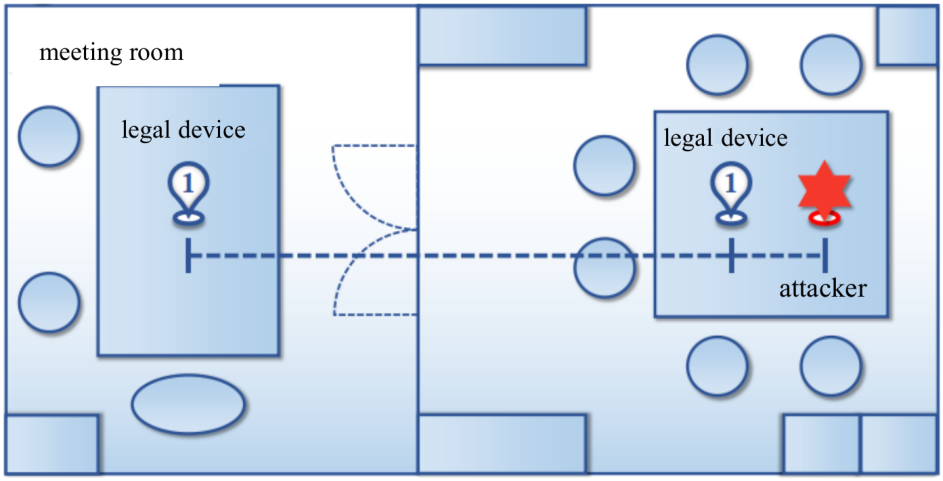
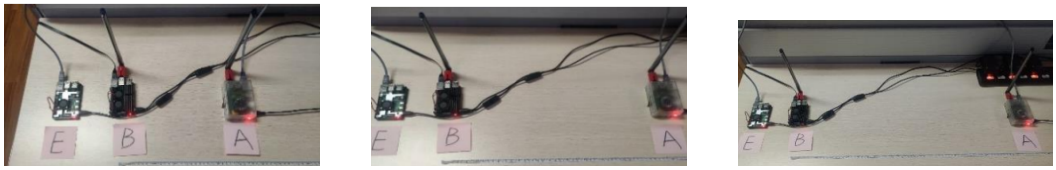


Figure 4. Illustration of key negotiation module testing scenario (non-line of sight).



(**a**) A and B are 25 cm apart (**b**) A and B are 50 cm apart (**c**) A and B are 75 cm apart

Figure 5. Illustrative diagram of key exchange module testing scenario (within line of sight).

6.1.2. Testing Apparatus

In the channel key negotiation module testing, three Raspberry Pi devices (all Raspberry Pi 3 Model B+) were employed as the testing equipment, with the device information outlined in Table 1. The three Raspberry Pi devices assumed the roles of Alice (legitimate communication device (1)), Bob (legitimate communication device (2)), and Eve (illegitimate communication device, i.e., the attacker) during the testing. Specifically, Alice and Bob functioned as legitimate communicating parties, executing normal access requests, channel feature collection, identity authentication, key negotiation, and communication processes. Meanwhile, Eve played the role of an attacker, engaging in man-in-the-middle attacks and replay attacks during the identity authentication and key negotiation stages conducted by Alice and Bob.

Table 1. Device information for channel key negotiation module testing.

No.	Model of Device	Name of Device	Identity of Device
1	Raspberry Pi 3 Model B+	Alice	Legal equipment
2	Raspberry Pi 3 Model B+	Bob	Legal equipment
3	Raspberry Pi 3 Model B+	Eve	Attacker

6.1.3. Software and Hardware Parameters

All Raspberry Pi models utilized in this paper are Raspberry Pi 3 Model B+. The Wi-Fi chipset integrated into all Raspberry Pi devices is identified as BCM43455C0. The antennas employed for both transmission and reception are 2.4 GHz single-frequency antennas with a gain of 6 dBi. All experimental tests are conducted based on the wireless communication protocol IEEE 802.11n, utilizing channels 1–14 with a center frequency range of 2.412–2.472 GHz and a bandwidth of 20 MHz. The firmware version installed on the Wi-Fi chipset is 7_45_189, and the operating system deployed on the Raspberry Pi is the Raspberry Pi OS, with a kernel version of 4.19.

The software parameters for the LOS and NLOS test environments of the channel key negotiation module are configured as presented in Tables 2 and 3, respectively.

Table 2. Parameter configuration for non-line-of-sight indoor testing software.

No.	Communication Distance (m)	Minimum Distance to the Supplier (cm)	Packet Transmission Rate (pkts/s)	Number of Transmitted Packets (pkts)	Number of Channel Key Negotiation Iterations
1	3	10	500	50,000	5000
2	4	10	500	50,000	5000
3	5	10	500	50,000	5000

Table 3. Parameter configuration for line-of-sight testing software.

No.	Communication Distance (m)	Minimum Distance to the Supplier (cm)	Packet Transmission Rate (pkts/s)	Number of Transmitted Packets (pkts)	Number of Channel Key Negotiation Iterations
1	25	10	500	20,000	2000
2	50	10	500	20,000	2000
3	75	10	500	20,000	2000

After experimental verification, it has been demonstrated that there is a significant disparity in the modulus results between Alice and Bob compared to those of Eve. This observation elucidates the efficacy of the modulus operation in enhancing the distinguishability between attackers and legitimate devices. Following the modulus operation, the proportion of data variations across the three sets remains within the anticipated and reasonable range. Subsequent quantization processing further amplifies the observed distinctions.

6.2. OFDM and CSI

6.2.1. Testing Parameters

This paper selected a wireless network with a bandwidth of 20 MHz and 64 OFDM subcarriers, utilizing a wireless terminal with a single transmitter and receiver antenna as a sample. Upon receiving data packets, the receiver calculated the CSI using a channel estimation method, resulting in a matrix of size $1 \times 1 \times 64$. In the long training sequence, the amplitudes of 12 subcarriers are zero, numbered -32, -31, -30, -29, -28, -27, 0, 27, 28, 29, 30, and 31. This implies that the CSI data on these subcarriers is not obtained through channel estimation and is not associated with the wireless channel. After removing these data points, a matrix of size $1 \times 1 \times 52$, denoted as the complex sequence, was obtained, where each value corresponds to the amplitude and phase of a subcarrier.

In this scenario, if the above sequence is directly used for training and recognition, considering factors such as hardware errors and environmental interference, there may be

outliers in the sample sequence, leading to the poor recognition performance of the trained model. Therefore, it is necessary to perform outlier handling, normalization, and smoothing on the obtained amplitude and phase sequences. This process is further explained in the following section, which discusses the preprocessing of CSI data.

6.2.2. Functional System Testing

This paper employed various wireless IoT devices of different types and brands, including Raspberry Pi, wireless cameras, smart lights, and smart plugs. These devices were connected to a router serving as an access point (AP) to create a simulated IoT network. All these IoT devices operated in the 2.4 GHz frequency band with a bandwidth of 20 MHz, and their communication data packets were captured. Nexmon was chosen as the CSI extraction tool because it offers support for larger bandwidths and a higher number of subcarriers compared to other tools.

This paper continuously recorded the changes in CSI amplitude curves for different IoT devices under four interference conditions, including scenarios with no human movement and individuals walking back and forth from distances of 1 m, 2 m, and 3 m from the terminal. A selection of typical IoT devices was chosen, and the CSI information varied among different devices. Even under interference conditions, the measured CSI amplitude curves maintained a similar shape, with an overall shift in amplitude occurring across all subcarriers. Further experiments indicated that, after processing, the differences in extracted device identities under various interference conditions were within the tolerance range of our algorithm.

6.3. CSI and LSH

Due to the inherent noise in the channel, signals inevitably degrade during transmission. The user identity used for encryption and decryption should be accurate. Therefore, channel state information (CSI) cannot be directly used as a user identity. Only through processing with locality-sensitive hashing (LSH), where the same segment of signal falls into the same hash bucket after stabilization, can CSI be used as a means of user identification. As illustrated in Figure 6, we transmitted 100 identical signal segments repeatedly. To simplify the problem, we calculate the average of the signal sequence in both amplitude and frequency dimensions. However, due to the inherent noise in the channel, the signals received by the actual receiver exhibit subtle differences in amplitude and frequency. These differences, amplified by the avalanche effect of cryptographic hash functions, result in completely different identity representations. Therefore, it is necessary to employ LSH to ensure that these 100 signal segments stably fall into the same hash bucket before they can be used for identity generation.

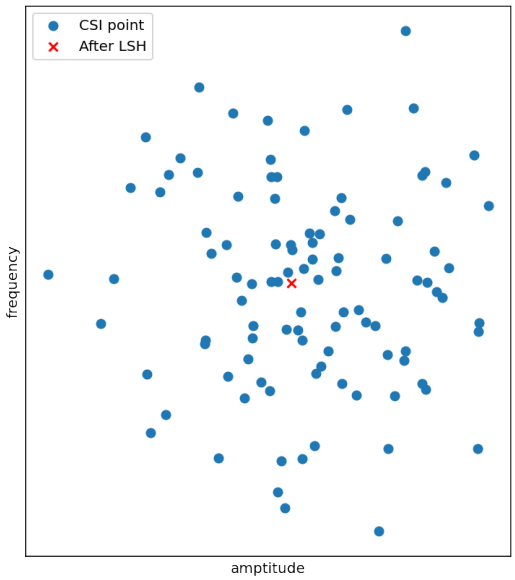


Figure 6. Stablity of LSH.

7. Discussion

7.1. Potential Attack and Security

In this section, we will discuss potential attacks that the methods described in this paper may face, as well as the security of the proposed solution.

7.1.1. Man-in-the-Middle Attack

The man-in-the-middle attack refers to the attacker inserting themselves between communicating parties to intercept or manipulate the information exchanged. Attackers can impersonate user identities through man-in-the-middle attacks. However, in the proposed solution in this paper, user identities rely on the physical characteristics of the user's device, specifically constructed through channel state information. Unlike traditional approaches that depend on specific message information submitted by users, the attacker cannot mimic the user's channel environment. Therefore, they are unable to impersonate users through man-in-the-middle attacks.

7.1.2. Replay Attack

The replay attack refers to the act of an attacker retransmitting previously intercepted communication data without modification, with the aim of deceiving the system to achieve unauthorized access or execute certain unauthorized operations. In this paper, on one hand, similar to a man-in-the-middle attack, attackers cannot mimic the channel state in which the user is located. On the other hand, in zero-knowledge proofs, a random freshness number is used as secret information. Therefore, attempting to impersonate a user through replaying messages cannot pass the zero-knowledge verification. Thus, in the discussed solution in this paper, it is not possible to impersonate a user through a replay attack.

7.1.3. Brute-Force Attack

The brute-force attack is a method of password cracking where attackers attempt all possible password combinations until the correct one is found. This method does not rely on pre-obtained information but systematically tries every possible password and is often automated using computer programs. It is a relatively straightforward but time-consuming attack, and its success typically depends on the strength and complexity of the password. In this paper, due to the difficulty of the discrete logarithm problem on elliptic curves, brute-force attacks are computationally infeasible. Meanwhile, attackers not only need to brute-force crack passwords but also attempt to brute-force crack the user's channel state. The attacker can only be successful if they find a channel that is equivalent to the one in

which the user's device is located. However, the state of the channel itself is uncertain, making it impossible for attackers to brute-force crack the user's channel state.

7.2. Challenges

7.2.1. Channel Time Variability

Channel time variability refers to the phenomenon where the characteristics of a communication channel change over time. In wireless communication environments, factors such as multipath propagation and the movement of obstacles can cause variations in the channel. This time variability may result in signal fading, changes in multipath effects, and even a distortion of the channel. Due to the existence of channel time variability, the user identity generated based on CSI in this paper is also subject to time variations. Therefore, to ensure the effectiveness of the system, periodic updates of the channel state and user identity are necessary. This introduces an additional communication overhead and may reduce the system's availability. Addressing the time variability of the channel for efficient real-time updates is a direction for future research.

7.2.2. Location-Dependent Channel State

The location-dependent channel state refers to the characteristics of a communication channel that are influenced by the spatial position of communication devices, capturing how the channel changes based on the location and movement of the devices involved. The communication channel state describes the variations in the signal during the transmission process, encompassing changes such as fading, delay, and other effects caused by propagation paths, multipath effects, obstacles, and other influences. The position, movement, and environmental changes of devices can all have an impact on the channel state. Therefore, the solution discussed in this paper has limitations for general discussions about wireless mobile devices. The devices discussed in this paper are also fixed devices in specific scenarios. In Figure 7, we demonstrate the effect of user identity recognition when the device undergoes a position shift. We recorded the successful recovered number at the red cross, and it can be observed that after deviating a certain distance and angle, the device becomes unacceptable. The location dependency of the channel is also one of the directions for future research, which holds significant practical value for applications in wireless scenarios.

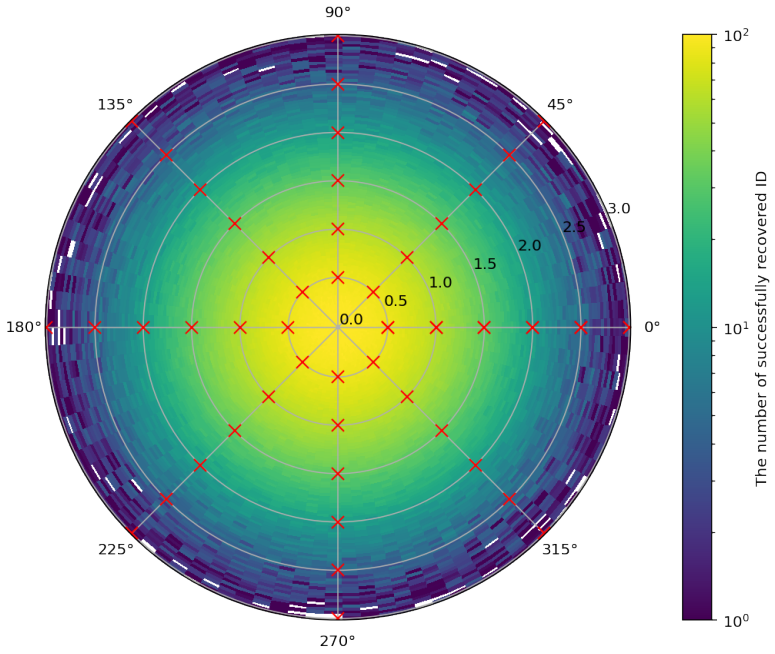


Figure 7. Location-dependent channel state and the number of successfully recovered IDs.

8. Conclusions

This paper introduces an innovative secure communication solution that leverages wireless channel state information (CSI) features from IoT devices for generating device identities. Given the inherent instability of wireless channels, the CSI features are inherently fuzzy and subject to time variations. To address this, we employ the locally sensitive hashing (LSH) algorithm, ensuring the stability of the generated identity within a dynamically changing wireless channel environment. Additionally, zero-knowledge proofs are incorporated to validate the authenticity and effectiveness of the generated identity. Subsequently, the identity produced through this approach is integrated into an identity-based encryption (IBE) communication scheme. This scheme encompasses the fuzzy extraction of channel state information from IoT devices, stable identity extraction for fuzzy IoT devices using LSH, and the application of zero-knowledge proofs to ensure the authenticity of the generated identity. The resultant identity serves as the basis for identity-based encryption, constructing the device's public key, and facilitating confidential communication among devices.

Author Contributions: Conceptualization, B.Z.; Methodology, B.Z. and Z.X.; Investigation, P.C.; Data curation, Y.L.; Writing—original draft, B.Z.; Writing—review & editing, T.Z. and Z.X.; Visualization, J.W.; Supervision, P.C.; Project administration, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China [2022YFB3104300].

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Bo Zhang, Tao Zhang and Zesheng Xi were employed by the company State Grid Smart Grid Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In *Annual International Cryptology Conference, Proceedings of the CRYPTO 2001, Santa Barbara, CA, USA, 19–23 August 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
2. Sakai, R. Cryptosystems based on pairing over elliptic curve. In Proceedings of the Symposium on Cryptography and Information Security-SCIS'01, Orlando, FL, USA, 22–25 July 2001.
3. Cocks, C. An identity based encryption scheme based on quadratic residues. In Proceedings of the Cryptography and Coding: 8th IMA International Conference, Cirencester, UK, 17–19 December 2001; Proceedings 8; Springer: Berlin/Heidelberg, Germany, 2001; pp. 360–363.
4. Waters, B. Efficient identity-based encryption without random oracles. In Proceedings of the Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Proceedings 24; Springer: Berlin/Heidelberg, Germany, 2005; pp. 114–127.
5. Afroaz, K.; Rao, Y.S.; Rukma, R.N. A Key Escrow Free Anonymous Identity Based Encryption Scheme Using Ring Signatures. In Proceedings of the 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), Indore, India, 23–24 April 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 596–600.
6. Veeresh, V.; Parvathy, L.R. Identity-based Encryption to Implement Anti-Collusion Information Sharing Schemes in Cloud Computing. In Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 4–6 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1177–1182.
7. Ngo, D.H. Private Identity-Based Encryption For Key Management. In Proceedings of the 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), Ho Chi Minh City, Vietnam, 26–27 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 416–420.
8. Liu, Y.; Wu, X.; Chen, X. A scheme for key distribution in wireless sensor network based on Hierarchical Identity-Based Encryption. In Proceedings of the 2015 IEEE 12th International Conference on Networking, Sensing and Control, Taipei, Taiwan, 9–11 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 539–543.
9. Wan, X.; Xiao, L.; Li, Q.; Han, Z. FHY-layer authentication with multiple landmarks with reduced communication overhead. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
10. Mahmood, A.; Aman, W.; Iqbal, M.O.; Rahman, M.M.U.; Abbasi, Q.H. Channel impulse response-based distributed physical layer authentication. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 4–7 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–5.

11. Wu, Q.; Feres, C.; Kuzmenko, D.; Zhi, D.; Yu, Z.; Liu, X.; Liu, X. Deep learning based RF fingerprinting for device identification and wireless security. *Electron. Lett.* **2018**, *54*, 1405–1407. [CrossRef]
12. Liao, R.; Wen, H.; Pan, F.; Song, H.; Xu, A.; Jiang, Y. A novel physical layer authentication method with convolutional neural network. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 29–31 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 231–235.
13. Shi, C.; Liu, J.; Liu, H.; Chen, Y. Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT. In Proceedings of the18th ACM International Symposium on Mobile ad Hoc Networking and Computing, Chennai, India, 10–14 July 2017; pp. 1–10.
14. Ribouh, S.; Phan, K.; Malawade, A.V.; Elhillali, Y.; Rivenq, A.; Al Faruque, M.A. Channel state information-based cryptographic key generation for intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7496–7507. [CrossRef]
15. Ji, Z.; Zhang, Y.; He, Z.; Yeoh, P.L.; Li, B.; Yin, H.; Li, Y.; Vucetic, B. Wireless secret key generation for distributed antenna systems: A joint space-time-frequency perspective. *IEEE Internet Things J.* **2021**, *9*, 633–647. [CrossRef]
16. Wang, S.; Huang, K.; Xu, X.; Zhong, Z.; Zhou, Y. Csi-based physical layer authentication via deep learning. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1748–1752. [CrossRef]
17. Wang, H.M.; Fu, Q.Y. Channel-prediction-based one-class mobile IoT device authentication. *IEEE Internet Things J.* **2021**, *9*, 7731–7745. [CrossRef]
18. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4298–4311. [CrossRef]
19. Niu, C.; Zheng, Z.; Wu, F.; Gao, X.; Chen, G. Trading data in good faith: Integrating truthfulness and privacy preservation in data markets. In Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, USA, 19–22 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 223–226.
20. Goldwasser, S.; Micali, S.; Rackoff, C. The knowledge complexity of interactive proof-systems. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 203–225.
21. Walshe, M.; Epiphaniou, G.; Al-Khateeb, H.; Hammoudeh, M.; Katos, V.; Dehghantanha, A. Non-interactive zero knowledge proofs for the authentication of IoT devices in reduced connectivity environments. *Ad Hoc Netw.* **2019**, *95*, 101988. [CrossRef]
22. Salleras, X.; Daza, V. SANS: Self-sovereign authentication for network slices. *Secur. Commun. Netw.* **2020**, *2020*, 8823573. [CrossRef]
23. Gabay, D.; Akkaya, K.; Cebe, M. A privacy framework for charging connected electric vehicles using blockchain and zero knowledge proofs. In Proceedings of the 2019 IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium), Osnabrueck, Germany, 14–17 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 66–73.