ChirpKey: A Chirp-level Information-based Key Generation Scheme for LoRa Networks via Perturbed Compressed Sensing

Huanqi Yang[1,2], Zehua Sun[1,2], Hongbo Liu[3], Xianjin Xia[4], Yu Zhang[5], Tao Gu[5],
Gerhard Hancke[1,2] and Weitao Xu[1,2,*]
[1]City University of Hong Kong Shenzhen Research Institute
[2]Department of Computer Science, City University of Hong Kong
[3]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[4]Department of Computing, The Hong Kong Polytechnic University
[5]School of Computing, Macquarie University

***Abstract*—Physical-layer key generation is promising in establishing a pair of cryptographic keys for emerging LoRa networks. However, existing key generation systems may perform poorly since the channel reciprocity is critically impaired due to low data rate and long range in LoRa networks. To bridge this gap, this paper proposes a novel key generation system for LoRa networks, named *ChirpKey*. We reveal that the underlying limitations are coarse-grained channel measurement and inefficient quantization process. To enable fine-grained channel information, we propose a novel LoRa-specific channel measurement method that essentially analyzes the chirp-level changes in LoRa packets. Additionally, we propose a LoRa channel state estimation algorithm to eliminate the effect of asynchronous channel sampling. Instead of using quantization process, we propose a novel perturbed compressed sensing based key delivery method to achieve a high level of robustness and security. Evaluation in different real-world environments shows that *ChirpKey* improves the key matching rate by 11.03–26.58% and key generation rate by 27–49× compared with the state-of-the-arts. Security analysis demonstrates that *ChirpKey* is secure against several common attacks. Moreover, we implement a *ChirpKey* prototype and demonstrate that it can be executed in $0.2\,\mathrm{s}$.**

***Index Terms*—Physical-layer key generation, LoRa, Compressed sensing**

I. Introduction

A. Background and Limitations

LoRa is one of the Low-Power Wide Area Network (LPWAN) technologies and it has received increasingly attention in recent years due to its open protocol standard and Chirp Spread Spectrum (CSS) modulation. Similar to legacy wireless technologies such as Wi-Fi and ZigBee, LoRa networks are vulnerable to malicious attacks due to the broadcasting nature of wireless communication. To secure communication, physical-layer key generation has emerged as a promising solution, which complements conventional security schemes (i.e., public key cryptography [1], pre-shared key [2]) due to its high efficiency and no prior requirements. Although secret key generation has been extensively studied over the past decades, existing studies mainly focus on ZigBee [3], Wi-Fi [4]–[6], and 5G [7]. Physical-layer key generation in LoRa networks poses new challenges due to its low data rate and long range.

To address these new challenges, a number of efforts have been made recently. The received signal strength indicator (RSSI) (i.e., channel characteristic) based method with quantization [8] has been proposed to present the feasibility of physical-layer key generation in LoRa networks, but the number of keys extracted by RSSI is limited due to course-grained channel characteristic. To improve the key generation rate, a register RSSI based method [9] has been proposed, but the achieved key generation rate is still relatively low (e.g., $13.8\,\mathrm{bits/s}$ only). Hence, we reveal that the following two fundamental challenges remain unsolved.

(1) Coarse-grained and noisy channel measurement

Physical-layer key generation is based on channel reciprocity, which indicates the two communication parties will have highly correlated channel measurements if the channel can be probed within the channel coherence time. Therefore, accurate and fine-grained channel measurement is the basis for efficient key generation. Unfortunately, the commonly used physical-layer characteristics for LoRa networks is RSSI [8], [10], which can only provide coarse-grained channel information. Some recent works [9], [11] utilize register RSSI to improve the granularity of wireless channel, but the register RSSI measurements are still noisy [9]. Therefore, existing channel indicators cannot provide sufficient and accurate information for physical-layer key generation in LoRa networks.

(2) Inefficient quantization process

In most physical-layer key generation systems, after obtaining channel measurements, the next step is quantization which converts channel measurements into binary bits (i.e., 1 or 0). Although a variety of quantization mechanisms have been proposed [8]–[12], the quantization process is inherently lossy and error-prone [6]. Due to noisy channel measurement and the low data rate of LoRa networks [13], the quantization-based key generation system may need to exchange more packets to obtain more channel information and run an additional information reconciliation process to partially fix errors. Hence, it may further lead to system inefficiency and lack of robustness.

[*] Weitao Xu is the corresponding author.

B. Contributions

In this paper, we design and implement a novel physical-layer key generation system for LoRa networks named *ChirpKey*, which essentially addresses the above challenges.

To enable fine-grained channel sampling, we take a closer look at the CSS modulation of LoRa's physical-layer. In our preliminary study, we reveal that every chirp modulated in sender has a constant amplitude, which can be utilized to indicate channel state. Our insight is to divide a LoRa packet into multiple chirps and calculate the fine-grained changes of chirps. Following this idea, we propose a LoRa-specific channel measurement named Chirp-Level Signal Strength Indicator (CLSSI) by analyzing the changes of the chirp units in LoRa packets. Additionally, due to the half-duplex mode (i.e., asynchronous channel sampling) of LoRa transceivers, channel information may be missed which impairs the channel reciprocity. Fortunately, with sufficient channel information provided by CLSSI, we observe that the missing channel information can be practically estimated. Therefore, we design a channel state estimation algorithm by adopting a lightweight spline fitting method, and the estimated CLSSI provides an accurate and comprehensive measurement of wireless channel.

To address the second challenge, we propose a novel Perturbed Compressed Sensing (PCS) based key delivery method that can efficiently deliver a secret key generated by one LoRa device to another. Inspired by the success of compressed sensing theory, our initial idea is to deliver the compressed key and reconstruct it on the receiver side directly using compressed sensing with fair robustness. However, since standard compressed sensing requires the same measurement matrix to be pre-shared by Alice and Bob, using channel measurements may not be able to construct the matrix, hence it may fail to apply in our work. Alternatively, we leverage PCS for the design of key delivery. Since PCS accepts the tiny difference between the matrices of sender and receiver, we intuitively use the channel measurements (i.e., similar but not the same channel information) to construct the matrices, where one matrix is used for compressing the secret key and another matrix is used for decompressing the key. In particular, due to decoupling the need for quantization and information reconciliation process, the proposed method is able to effectively achieve system efficiency and robustness for secret bit agreement. Moreover, different from the quantization-based methods, since the secret key is obtained from a random key generator, *ChirpKey* can build a key with strong randomness.

We conduct extensive evaluations in both indoor and outdoor environments. Results show that *ChirpKey* achieves a high matching rate and outperforms the state-of-the-arts. We also demonstrate the security of *ChirpKey* against common attacks via rigorous proof and evaluation. In summary, this paper makes the following contributions.

- We propose *ChirpKey*, a novel physical-layer key generation scheme for LoRa networks. *ChirpKey* addresses two key limitations in existing work and enables fast and secure physical-layer key generation.
- We propose a novel fine-grained channel state indicator, named CLSSI. Compared to existing channel indicators, CLSSI can provide fine-grained and accurate channel state information. To improve the integrity of channel information, we propose a lightweight channel state estimation method to comprehensively recover the channel information.
- We propose a novel PCS-based key delivery scheme to deliver secret keys. Compared to existing quantization-based solutions, the proposed method improves the robustness significantly. We demonstrate the security of the proposed scheme via rigorous proof and extensive evaluation.
- We conduct extensive experiments to evaluate *ChirpKey* in different real environments. Results show that *ChirpKey* achieves an average key matching rate of 99.58% and a key generation rate of 13 bits per measurement. Compared to the state-of-the-arts, *ChirpKey* improves key matching rate and key generation rate by 11.03–26.58% and 27–49×, respectively. Results also show that it takes less than 0.2 s to generate a 128-bit key and incurs low power consumption.

II. Related Work

Wireless Key Generation. Wireless key generation has received considerable attention over the past decades. In the literature, a large volume of systems have been proposed for different wireless technologies, such as Wi-Fi [6], [14], Zigbee [3], and Bluetooth [15]. In these studies, researchers have used a variety of physical-layer features, including Channel State Information (CSI) [6], [14], RSSI [9], [11], and phase [16]. For example, TDS [6] exploited Wi-Fi CSI as channel characteristics to generate keys for mobile devices. To enhance the channel reciprocity of CSI, Liu *et al.* [14] leveraged channel response in multiple Orthogonal Frequency-Division Multiplexing (OFDM) subcarriers, coupled with a Channel Gain Complement (CGC) scheme for key generation.

LoRa is an emerging wireless communication technology designed specifically for long-range and low-power communications. The low data rate and long airtime feature of LoRa bring new research challenges. To address these challenges, several key generation systems for LoRa networks have been proposed in recent years. For example, LoRa-Key [8] is the first RSSI-based key generation method for LoRa. In their follow-up research [9], a variant RSSI feature, register RSSI, is exploited, which can provide finer granularity of channel sampling for key generation. Recently, Yang *et al.* [11] utilized the mean value of adjacent rRSSI (arRSSI) of LoRa signals for key establishment on Internet of Vehicles (IoV) scenarios. However, the performance of these systems is limited because of the use of coarse-grained channel characteristics and inefficient quantization mechanism. Our work differs from existing studies in two aspects. First, we propose a novel LoRa-specific channel characteristics that can provide fine-grained channel state information. Second, we propose a novel PCS-based key delivery method instead of using quantization-based key generation methods.

Compressed Sensing. Compressed sensing is a signal processing technique used to reconstruct signals efficiently by finding solutions for underdetermined linear systems. In addition to data compression, it can also be applied in IoT security schemes [17], [18]. For example, H2B [19] used a compressed sensing-based reconciliation method to correct key mismatches due to the low SNR of the heartbeat interval signals. Additionally, Kryptein [18] proposed a compressed sensing-based encryption method to enable secure data queries for cloud-enabled IoT systems. Dautov *et al.* [17] explored the feasibility of constructing secure compressed sensing matrices based on wireless physical-layer security. These existing works apply compressed sensing to different IoT security scenarios, which inspires this paper to pioneer the use of a compressed sensing framework for wireless key generation. Unlike existing studies that used standard compressed sensing techniques for IoT security, our work uses the perturbed version of compressed sensing for the physical-layer key generation, which is more practical in real-world applications.

LoRa Security. With massive deployments of LoRa networks, the security issues have attracted significant efforts [20]–[23], with an emphasis on key management, authentication, and physical-layer key generation. For key management, existing works mainly focus on the derivation, distribution, and destruction process of application layer keys, including DevNonce [24] and NwkSKey [25]. Additionally, researchers have exploited different features of LoRa signals resulted from hardware imperfections for device authentication purpose, such as carrier frequency offset [26], amplitude-phase [27], and signal spectrogram [28]. However, such methods generally suffer from heavy computation cost and poor scalability. Thus, key generation has emerged as a promising solution for secure wireless communication due to its high energy efficiency and prior-free requirements compared with the aforementioned schemes. In this paper, we propose a novel secret key generation scheme to provide robust and lightweight key establishment for LoRa networks.

III. System Model

A. User Model

We assume there are two devices in a LoRa network, namely Alice and Bob, that intend to agree on the same key to safeguard their communication. They are both embedded with LoRa communication modules, with no prior sharing of secrets. They follow the work-flow in Fig. 1 to generate keys step by step. In the first phase, they measure the channel by exchanging a number of probe and response packets. Then both Alice and Bob start key generation phase, which includes channel probing, channel state estimation, perturbed measurement matrix generation, key compression and reconstruction, and privacy amplification. Finally, the key is used to encrypt/decrypt data to ensure secure communication.

B. Attack Model

We assume the presence of an attacker Eve, who tries to intercept the communication with the aim of generating the same key. Theoretically, due to the spatial de-correlation nature of the wireless channel, Eve will obtain completely different channel measurements if she is more than $\lambda/2$ away from Alice or Bob, where λ is the wavelength of the LoRa signal [5]. In practice, this condition can be easily satisfied because if Eve is less than $16.4\,\mathrm{cm}$ away from either Alice or Bob, she can be easily spotted (λ = $32.79\,\mathrm{cm}$ for $915\,\mathrm{MHz}$ LoRa). Same as prior works [6], [29], [30], we assume that Eve has complete knowledge of the key agreement process and that she has the ability to eavesdrop, inject, and replay messages in the public channel. Additionally, we assume the objective of Eve is to intercept the secret key rather than jamming their key establishment process (i.e., a Denial-Of-Service attack). In this paper, we consider three types of attacks that are widely considered in previous works [29], [31].

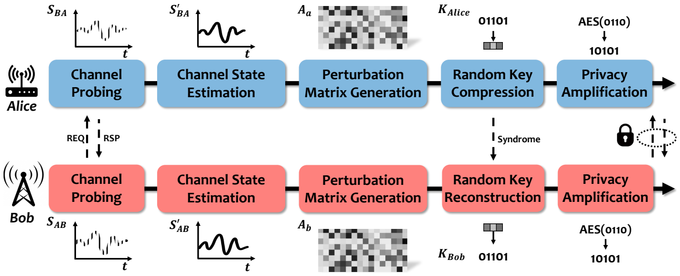


Fig. 1: System workflow.

- Eavesdropping attack. Since Alice and Bob need to exchange some information via a public channel during key generation, Eve can eavesdrop their conversation. Then, with the eavesdropped information, Eve tries to run *ChirpKey* to establish the same key as Alice and Bob.
- Imitating attack. Imitating attack is a common attack in mobile scenarios, where Eve observes the trajectory of Alice or Bob, and tries to imitate its moving trajectory to obtain similar channel measurements. However, as mentioned above, Eve cannot be too close to Alice/Bob, otherwise it increases the risks of being detected.
- Predictable channel attack. Predictable channel attack is a common attack in static scenario, where Eve tries to intentionally cause an expected change in channel measurements between Alice and Bob. For example, when Alice and Bob are static devices, Eve can travel regularly between them to create predictable channel changes.

IV. Chirp-level Channel Information Extraction.

This section presents the proposed fine-grained channel indicator and a channel state estimation method to improve channel reciprocity.

A. Fine-grained Channel Information for LoRa

As mentioned previously in Section I, existing physical-layer characteristics for LoRa networks (e.g., RSSI) cannot provide fine-grained channel information, impairing channel reciprocity. As proof shown in Fig. 2, the correlation between Alice and Bob using RSSI and register RSSI drops significantly as data rate is lower than $250\,\mathrm{bps}$, leading to poor channel reciprocity in physical-layer key generation. One of our key observations reveals that changes of LoRa chirp can indicate the channel states, which is promising to achieve fine-grained channel measurement. Hence, we propose a novel

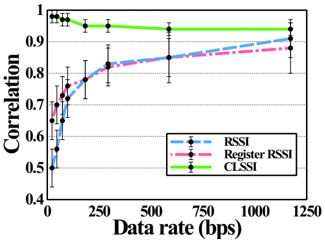


Fig. 2: Correlation analysis.

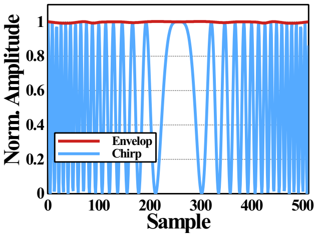


Fig. 3: Envelop of a chirp.

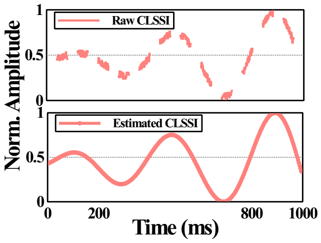


Fig. 4: Channel state estimation.

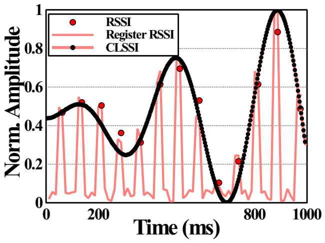


Fig. 5: CLSSI and RSSI.

CLSSI in the first step of *ChirpKey* (i.e., channel probing), as shown in Fig. 1. In principle, the proposed CLSSI can enhance channel reciprocity to obtain sufficient and accurate channel information for key generation, compared with traditional packet-level RSSI. The calculation details are as follows.

Recall each LoRa packet is composed of multiple chirps. According to the definition of chirp $x(t) = \sin(\phi(t))$, where $\phi(t)$ is a linear function for LoRa signal, the frequency of chirp varies with time while the amplitude is constant; therefore, the upper envelope of the received chirps can reflect the attenuation of LoRa signal after propagating in the air, as illustrated in Fig. 3. Inspired by this observation, we first divide a LoRa packet into multiple chirp units. Then we detect the local maxima of each chirp. Finally, we can obtain the CLSSI by calculating the mean amplitude of the envelope of the local maxima for each chirp. Compared to traditional RSSI and the recently proposed register RSSI [9], [11], CLSSI offers two advantages. First, CLSSI is LoRa-oriented and its calculation process is lightweight, making it suitable for LoRa network. Second, CLSSI is based on the chirp-level information, which can provide fine-grained channel state information. As will be demonstrated in Sec. VI-C, the performance of *ChirpKey* is significantly improved after using CLSSI.

B. Channel State Estimation

The proposed CLSSI can provide fine-grained channel information, but the non-simultaneous channel sampling in time-division duplex communication systems significantly impairs the channel reciprocity. As shown in the upper figure of Fig. 4, the CLSSI sequences acquired by Alice or Bob after channel probing can only reflect half of the channel variance due to the non-simultaneous channel sampling. In other words, Alice can only extract channel information when it is receiving a packet but not when it is sending a packet to Bob, and *vice versa*. Besides, the measurement noise caused by hardware imperfections also deteriorates the performance of key generation.

To address these problems, we propose to use the channel state when Alice and Bob are receiving packets to estimate the channel state when they are sending packets, thus both Alice and Bob can obtain the complete channel state information during channel probing. Suppose the extracted CLSSIs for Alice and Bob are $S_A = \{S_a^1, S_a^2, \ldots, S_a^n\}$ and $S_B = \{S_b^1, S_b^2, \ldots, S_b^n\}$, respectively, where S_a^i and S_b^i are the CLSSI sequences obtained by Alice and Bob after they receive the i-th packet. Note that because Bob first sends a probe packet to Alice, the chronological order of these CLSSI sequences is $\begin{pmatrix} S_a^1 & \times & S_a^2 & \ldots & S_a^n & \times \\ \times & S_b^1 & \times, & \ldots & \times & S_b^n \end{pmatrix}$, where × means missed channel measurement at that time. We can see S_a^1 and S_b^1 are not calculated simultaneously due to the half-duplex mode of LoRa transceivers. To solve this problem, we propose to interpolate to halfway between Alice and Bob's probing times. Take Alice as an example, we apply an interpolation on S_a^1 and S_a^2 to estimate $S_a^{1'}$, which is the channel characteristic measured at the same time as S_b^1. By repeating the same procedure on the remaining CLSSI sequences, Alice can obtain $S_A' = \{S_a^1, S_a^{1'}, S_a^2, S_a^{2'}, \ldots, S_a^n, S_a^{n'}\}$. Similarly, Bob can obtain the estimated channel characteristics $S_B' = \{S_b^{1'}, S_b^1, S_b^{2'}, S_b^2, \ldots, S_b^{n'}, S_b^n\}$. In this way, both Alice and Bob can obtain the complete channel characteristics and each CLSSI is measured at the same time because the chronological order of the estimated CLSSI sequences is $\begin{pmatrix} S_a^1 & S_a^{1'} & S_a^2 & \ldots & S_a^n & S_a^{n'} \\ S_b^{1'} & S_b^1 & S_b^{2'} & \ldots & S_b^{n'} & S_b^n \end{pmatrix}$.

In *ChirpKey*, we choose uni-variate spline fitting [32] method for the following reasons. First, compared with polynomial fitting, the spline function uses low-order polynomial for fitting, which is simple in calculation. Second, the spline function is a piecewise function of low-order polynomial, which guarantees smooth transitions between points and the ability to converge. Therefore, spline fitting can provide fast and stable channel estimation for *ChirpKey*.

To demonstrate the effectiveness of the above methods, Fig. 5 plots the RSSI, register RSSI, and our CLSSI from the same LoRa signal. We can observe that RSSI has fewer sampling points and more outliers. Although register RSSI provides more sampling points, most of the sampling points are hardware noise. Moreover, only a small number of sampling points are similar to RSSI but they are unstable. In comparison, the proposed CLSSI can provide accurate and fine-grained channel measurement for LoRa signals.

V. PCS-based Key Delivery

As aforementioned, the traditional quantization process is inherently lossy and may cause mismatches between legitimate nodes [6], e.g., the fixed threshold may lead to mismatched quantized bits for Alice and Bob [4], [29], [33]. While simply using compressed sensing method may fail due to different matrics caused by the channel measurements. To address, this section presents the proposed PCS-based key delivery method. Since the proposed method is based on perturbed compressed sensing, we first briefly describe the technical background of PCS, then present the measurement matrix generation, key compression and reconstruction. Finally, we provide a rigorous proof to demonstrate the security of the proposed method.

A. Principle of Perturbed Compressed Sensing

Compressed sensing is a technique in the signal processing field which allows acquiring signals while taking few samples. Assume there is a linear compression system $y = Ax$, where $x \in \mathbb{R}^N$ is the original signal, $A \in \mathbb{R}^{M\times N}(M < N)$ is the measurement or sensing matrix, and $y \in \mathbb{R}^N$ is the compressed signal. Compressed sensing states that if x is sparse, it can be reconstructed from far fewer samples than required by the Nyquist–Shannon sampling theorem.

In standard compressed sensing, the measurement matrix A is assumed to be exactly known and identical by the compressor and receiver, so that the receiver can recover x by ℓ_1 minimization:

$$\hat{x} = \arg\min_{x} \|x\|_1 \quad \text{subject to } \|y - Ax\|_2 < \epsilon, \tag{1}$$

where ϵ is used to account for noise. Unfortunately, such an ideal assumption is not always the case in practice. For example, when the compressed signal y and measurement matrix A is transmitted from a compressor to a recoverer via a wireless channel, the received y and A are often perturbed versions due to noise:

$$\hat{A} = A + E \text{ and } \hat{y} = y + e, \tag{2}$$

where $E \in \mathbb{R}^{M\times N}$ and $e \in \mathbb{R}^{M\times 1}$ denote unknown perturbations from different sources, such as ambient noise, measurement error, and coding error. In this case, the standard compressed sensing recovery process becomes how to recover the original signal x from the perturbed compressed signal y and measurement matrix $\hat{A}$. According to perturbed compressed sensing theory [34], this problem can be solved by the following ℓ_1 minimization:

$$\arg\min_{x,e,E} \|e\|_2^2 + \|E\|_F + \lambda\|A\|_1 \quad \text{subject to } \hat{y} = \hat{A}x, \tag{3}$$

where $\lambda > 0$ is the regularization parameter, $\|\cdot\|_2, \|\cdot\|_F$, and $\|\cdot\|_1$ stand for ℓ_2, Frobenius, and ℓ_1 norms, respectively. To solve the above ℓ_1 minimization problem, we can use the fast reconstruction algorithm based on total least-squares and proximal splitting [34].

B. Perturbed Measurement Matrix Generation

Based on the above PCS theory, we propose a novel PCS-based key delivery method. The main idea of our method is *since Alice and Bob have similar but not the same channel measurements (i.e., S'_A and S'_B), if we can use S'_A and S'_B to construct two measurement matrices that are similar to A and $\hat{A}$ above, then Alice can compress the key which can be recovered by Bob. While for attacker Eve, since her channel measurements are totally different, she cannot generate a similar perturbed measurement matrix to successfully reconstruct the key.* In the following, we present the details of the proposed method which includes perturbation matrix generation, compression and reconstruction.

Perturbation matrix generation. Structured matrix perturbation [35] is a commonly used perturbation with a fixed structure in PCS. Each of its columns is an unknown constant, which is imposed with a known operation that defines the direction of the perturbation. In this paper, we use the circulant matrix as the structured perturbation matrix because it is easy to be implemented in low-power LoRa end nodes. The construction method of a typical circulant matrix $f(C)$ is as follows. First, a random vector C is generated, that is, $C = (c_0, c_1, \cdots, c_{N-1}) \in R^N$. Then the generated random vector C performs cyclic displacement for M times to construct the remaining $M - 1$ row vectors. Finally, all the row vectors are used to generate the entire matrix $f(C)$ through cyclic displacement as

$$f(C) = \begin{pmatrix} c_0 & c_{N-1} & \cdots & c_1 \\ c_1 & c_0 & \cdots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_{M-1} & c_{M-2} & \cdots & c_M \end{pmatrix}. \tag{4}$$

Perturbed measurement matrix generation. We assume there is a default sensing matrix A_0 (we use random Gaussian matrix in *ChirpKey*), and the goal of Alice and Bob is to use their channel measurements to generate a perturbation matrix which perturbs the default matrix. We find that using the generated perturbation matrix to directly perturb the default matrix will decrease the reconstruction capability when the average value of the perturbation matrix is large. Thus, to control the magnitude of the generated perturbation matrix, we apply a scale factor η to the estimated CLSSI values of Alice S'_A and Bob S'_B to control the magnitude of the perturbation matrix of Alice: $\hat{S}_A = \frac{1}{\eta}S'_A$ and $\hat{S}_B = \frac{1}{\eta}S'_B$. The analysis of selecting a suitable η is presented in Sec.V-D. Afterwards, the generated perturbed measurement matrices of Alice and Bob can be obtained by adding generated perturbation matrix to the default matrix A_0 as

$$A_a = A_0(I + E_a) \text{ and } A_b = A_0(I + E_b), \tag{5}$$

where I is the identity matrix, $E_a = f\left(\hat{S}_A\right)$ and $E_b = f\left(\hat{S}_B\right)$ are the generated circulant matrices of Alice and Bob.

C. Compression and Reconstruction

After obtaining the perturbation measurement matrices A_a and A_b, Alice and Bob are able to perform information compression and reconstruction.

Compression by Alice. First, Alice generates a random binary sequence $K_A \in R^N$. Since compressed sensing requires sparse input, we sparse the random sequence by interpolating four zero bits (sparsity level > 4) between each bit to transform K_A to be a spare vector K_A^s. Next, Alice calculates a syndrome which is defined by $Syn = [Syn_1, Syn_2] = [A_a K_A^s, A_a K_A]$. The first part of the syndrome (i.e., $A_a K_A^s$) is a compression of the secret key K_A^s. The second part of the syndrome (i.e., $A_a K_A$) is used for error checking and correction. Afterwards, Alice transmits the syndrome Syn to Bob through the public channel.

Algorithm 1: PCS-based key delivery.

```
Input: $\hat{S}_A$ and $\hat{S}_B$: estimated CLSSI sequences measured by
       Alice and Bob. $A_0$: the original measurement matrix.
Output: $K_A$ and $K_B$: secret keys for Alice and Bob.
Alice:
$A_a = A_0 + f\left(\hat{S}_A\right)$
$K_A = PRNG(seed)$                                   ▷ generate random key
$K_A^s = sparse(K_A)$                                    ▷ sparse the key
$Syn = [Syn_1, Syn_2]$, $Syn_1 = A_a K_A^s$, $Syn_2 = A_a K_A$
Send $Syn$ to Bob via public channel
Bob:
Receive noised syndrome $Syn' = Syn + e$
$A_b = A_0 + f\left(\hat{S}_{AB}\right)$
$K_B^s = solve\_\ell_1(Syn_1, A_b)$
$K_B' = de\text{-}sparse(K_B^s)$                          ▷ reconstruct the key
$\Delta = A_b K_B' - Syn_2$
$\Delta_{AB} = solve\_\ell_1(\Delta, A_b)$
$K_B = K_B' \oplus \Delta_{AB}$                        ▷ solve the mismatched bits
return $K_A, K_B$
```

Reconstruction by Bob. Let Syn' be the received syndrome with some noise $Syn' = Syn+e = [Syn_1+e, Syn_2+e]$. After receiving the syndrome, Bob uses the first part of syndrome $(Syn_1 + e)$ to reconstruct a sparse key K_B^s by solving the following ℓ_1-regularized total least-squares problem:

$$\begin{aligned} &\underset{e, E_p, K_B^s}{\arg\min} \quad \|e\|_2^2 + \|E_p\|_F + \lambda \|K_B^s\|_1 \\ &\text{subject to} \quad (A_b + E_p) K_B^s = Syn_1 + e, \end{aligned} \tag{6}$$

where E_p is the perturbation difference between A_b and A_a, namely $E_p = A_b - A_a$. After obtaining the sparse key K_B^s, Bob de-sparses it to obtain K_B'. After this step, Bob obtains an estimated key K_B' from the first part of the syndrome.

In practice, K_B' is not exactly the same as Alice's original key K_A because of noise. Therefore, Bob uses the second part of the syndrome to correct the errors. To this end, Bob first calculates a mismatched vector Δ as follows

$$\begin{aligned} \Delta &= A_b K_B' - Syn_2 = (A_b K_B' - A_a K_A) + e \\ &= A_b K_B' - (A_b + E_p) K_A = A_b (K_B' - K_A) + e \\ &= A_b \Delta_{AB} + e, \end{aligned} \tag{7}$$

where Δ_{AB} means the mismatches between Alice's original key K_A and Bob's estimation K_B'. Since Alice and Bob are legitimate devices, there are only few mismatches in their keys, i.e, Δ_{AB} is sparse. Therefore, Bob can reconstruct the mismatches by solving the following ℓ_1-regularized total least-squares problem:

$$\underset{\Delta, e}{\arg\min} \|e\|_2^2 + \lambda \|\Delta\|_1 \quad \text{subject to } A_b \Delta_{AB} = \Delta + e, \tag{8}$$

With Δ_{AB}, Bob can deduce K_A by simply calculating $K_B = K_B' \oplus \Delta_{AB}$, where $\oplus$ is XOR operation. Finally, both Alice and Bob agree on the same key $K_A = K_B$. The above key delivery process is summarized in Algorithm 1.

D. Security of the Proposed Method

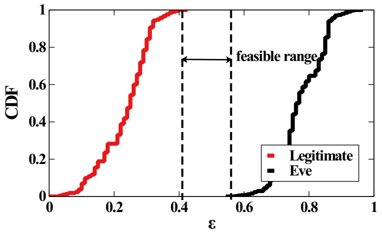


Fig. 6: Security of the proposed method.

Since the syndrome Syn is transmitted via an unauthenticated channel, the attacker Eve can also eavesdrop the message syn'. With the knowledge of syn', she can then perform the three types of attacks discussed in Sec. III to derive her own key K_{Eve} using the same method as Bob. We now quantitatively demonstrate that the above vulnerability can be addressed by properly choosing the perturbation matrices.

Corollary 1: Suppose the perturbation matrix of Alice, Bob and Eve are A_a, A_b, and A_e, respectively. The PCS-based key delivery method is perfectly effective (i.e., Bob can reconstruct K_A successfully) and secure (i.e., Eve is unable to reconstruct K_A successfully) if there exists a parameter ε satisfying

$$\frac{\|A_b - A_a\|_2}{\|A_a\|_2} \leq \varepsilon < \frac{\|A_e - A_a\|_2}{\|A_a\|_2}. \tag{9}$$

Proof 1: If Bob cannot recover Δ_{AB} and K_A with a ε that satisfies condition $\frac{\|A_b - A_a\|_2}{\|A_a\|_2} \leq \varepsilon$, it is contradictory to the *sufficient* condition for successful recovery according to PCS theory [35]. If Eve can recover K_A with $\frac{\|A_e - A_a\|_2}{\|A_a\|_2} \leq \varepsilon$, it is contradictory to the *necessary* condition for successful compressed sensing decoding [35].

Therefore, the problem becomes finding a feasible range of ε that satisfies the *Corollary 1* (successful reconstruction of the legitimate node and the unsuccessful reconstruction of the attacker). Fig. 6 plots the distribution of the lower bound and upper bound of ε. We can see that there is a feasible range $[0.41, 0.56]$ to use. In other words, if ε lies in the feasible range, Equation. 9 holds, meaning that Eve cannot use her observed message syn' to obtain the same key. Therefore, we control ε within the feasible range in the following experiments.

Prior studies also show that when the same measurement matrix A_0 is used repeatedly, the syndrome could be conditionally accessed [8], [36]. This issue can be easily solved by updating the original measurement matrix A_0 periodically. For example, after Alice and Bob agree on the same key, they can use it as a seed of Gaussian random matrix generator to update A_0. Note that although we need to pre-store A_0 in *ChirpKey*, it is a public knowledge rather than a pre-shared secret.

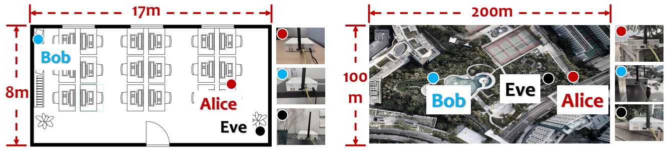
E. Privacy Amplification

Although the PCS-based key delivery method achieves high reliability as evaluated in Sec. VI, it also reveals some information to attackers because Alice and Bob have to transmit the syndrome in the public channel. This problem can be solved by privacy amplification techniques, such as hash functions [8], [29]. In *ChirpKey*, We employ the widely used hash function SHA-128 as a privacy amplification technique to increase the randomness of the final keys. After generating the same key, Alice and Bob can use AES-128 or other symmetric key encryption methods to encrypt/decrypt their communications.

VI. Evaluation

A. Experimental Setup

Data collection. As shown in Fig. 7, we use three USRP N210 SDR with WBX Daughterboard as Alice, Bob, and Eve, respectively. We use the following LoRa parameters: $F = 915\,\text{MHz}$, $BW = 125\,\text{kHz}$, $SF = 12$, and $CR = 4/8$, where F denotes frequency, SF denotes spread factor, BW denotes bandwidth and CR denotes code rate. Alice is configured to be an end-device, and it is connected to a Raspberry Pi module with 1.5 GHz Quad core Cortex-A72 and 4 GB RAM via Ethernet cable. Bob is configured to be a LoRa Gateway, and it is connected to a server with AMD EPYC 7522 64-core processor via Ethernet cable. The LoRa radio signal processing algorithm is implemented in C++ with GNU Radio in SDR. The key generation algorithm is implemented in the server and Raspberry Pi for Alice and Bob, respectively. As shown in Fig 7, we conduct experiments in both indoor and outdoor environments. In both environments, we consider two different scenarios: static scenario where both Alice and Bob are static, and mobile scenario where Bob is static but Alice is moving. This setting is realistic because LoRa can be used in both static wireless sensor networks and mobile networks. The attacker Eve is placed $1\,\text{m}$ away from Alice. The whole experiment lasted for two weeks and we collected more than 2,000,000 channel measurements at different times of different days.



(a) Indoor experiment. (b) Outdoor experiment.

Fig. 7: Experimental setup.

Metrics. For a key generation system, we follow two widely used metrics: 1) *Key matching rate (KMR)* is the percentage of the matching bits over all bits; 2) *Key generation rate (KGR)* is the number of bits generated in one second.

B. Overall Performance

Baselines. We select below state-of-the-art quantization-based key generation systems in LoRa as our baselines.

- **LoRa-key** [8] uses RSSI channel measurement and compressed sensing based reconciliation.
- **LoRa-liSK** [10] uses RSSI channel measurement and error-correction code based reconciliation.
- **Gao *et al.*** [9] uses register RSSI channel measurement and compressed sensing based reconciliation.
- **Vehicle-key** [11] uses register RSSI channel measurement and autoencoder based reconciliation.

To be fair, we adjust their settings to get the best results. Specifically, for LoRa-liSK [10] and LoRa-Key [8], we set α (the guard band ratio in quantization) to 0.8. For Gao *et al.* [9], the interval is set to 20 and the round number is set to 50. For CS-based reconciliation method in LoRa-Key [8] and Gao *et al.* [9], the measurement matrix size is set to 50×128. For Vehicle-key [11], the compressed size of autoencoder is set to 64. For these methods, we use Arduino Uno with Dragino LoRa Shield[1] to collect RSSI and register RSSI values.

The results of different methods in various environments are presented in Fig. 8, Fig. 9, and Fig. 10, respectively. From Fig. 8, we observe that *ChirpKey* achieves the highest KMR in all the scenarios. The average key matching rate of *ChirpKey* is 25.61% higher than LoRa-liSK [10], 26.58% higher than LoRa-Key [8], 16.31% higher than Vehicle-key [11], and 11.03% higher than Gao *et al.* [9]. The stability (i.e., standard deviation) of the KMR obtained by *ChirpKey* also outperforms other methods. From Fig. 9, we can see that the KGR of *ChirpKey* is 27.5× higher than LoRa-key [8], 26.9× higher than Vehicle-key [11], 49× higher than Gao et al. [9], and 27.5× higher than LoRa-liSK [10], respectively. Fig. 10 shows the entropy of the keys generated by different methods. We observe that *ChirpKey* improves entropy by 6-8% compared with baselines. Additionally, as shown in Fig. 8 and Fig. 9, the KMR in outdoor environment is lightly lower than that of indoor areas because there are less multi-path effect in outdoor areas. The KGR in mobile scenario is slightly higher than that of static scenario because there are more channel variations when Alice is moving. To sum up, the results show that *ChirpKey* improves KMR, KGR, and entropy significantly compared to the state-of-the-arts in different scenarios.

C. Evaluation of Chirp-level Channel Information

Impact of CLSSI. We now demonstrate the advantage of the proposed CLSSI over two widely used channel measurements in LoRa networks, namely RSSI and register RSSI. As shown in Fig. 11, the proposed CLSSI achieves higher KMR and KGR compared with both RSSI and register RSSI. Specifically, the KGR is increased by 2.72% and 2.94%, while the KGR is increased by 55.32× and 26.73× compared with using RSSI and register RSSI, respectively. Therefore, our CLSSI can provide fine-grained and accurate channel state information for LoRa networks.

Impact of channel state estimation. In this experiment, we evaluate the performance of the proposed channel state estimation method. To this end, we compare the KGR with and without channel state prediction. As shown in Fig. 12, the proposed method consistently achieves higher matching rate in all scenarios, indicating our method is effective in improving the channel reciprocity for legitimate nodes and hence the KGR. Specifically, the KGR is increased by 2×, 2.1×, 1.9×, and 1.95× in indoor-static, indoor-mobile, outdoor-static, and outdoor-mobile scenarios, respectively. Besides, the channel state estimation method can also decrease the deviation of the KGR, indicating it improves the stability of *ChirpKey*.

D. Evaluation of Parameters in the PCS Method

Impact of measurement matrix size. We examine the impact of the size of measurement matrix on *ChirpKey* by changing the matrix size from 40×128 to 58×128 (i.e., increase M from 40 to 58 gradually). As shown in Fig. 13, the

[1] https://www.dragino.com/products/lora/item/102-lora-shield.html

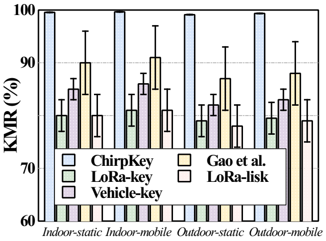


Fig. 8: Comparison of matching rate.

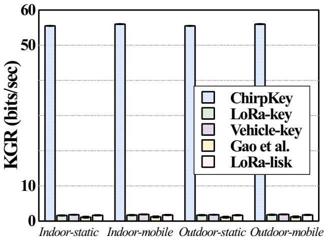


Fig. 9: Comparison of generation rate.

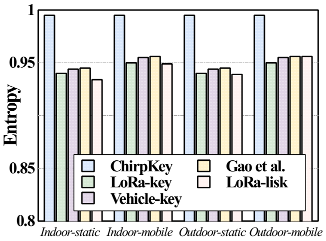


Fig. 10: Comparison of entropy.

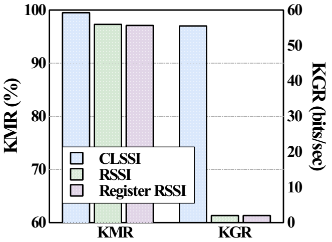


Fig. 11: Impact of CLSSI.

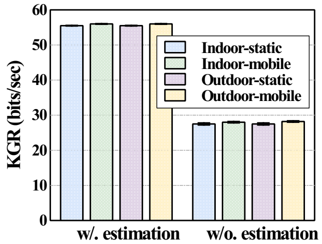


Fig. 12: Impact of Channel Estimation.

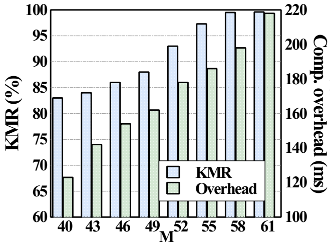


Fig. 13: Impact of measurement matrix size.

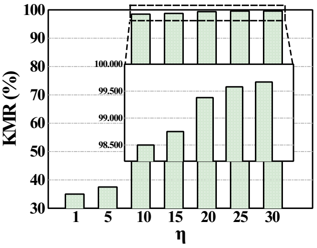


Fig. 14: Impact of scale factor.

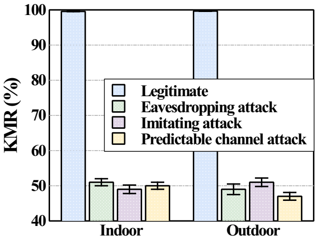


Fig. 15: Attack analysis.

average KMR increases as the matrix size increases, indicating that the reconstruction capability of *ChirpKey* is improved with larger size of compressed vector. However, the KMR levels off when M is larger than 58, while the computational overhead keeps increasing. Therefore, we set M to 58 in the following experiments. To transmit a compressed vector of size 58, $58 \times 16\,\text{bits} = 116\,\text{bytes}$ are required. Since LoRa offers a maximum payload size of 242 bytes, only one packet is required to transmit the compressed vector.

Impact of scale factor. Then, we analyse the impact of the size of scale factor on *ChirpKey* by changing the scale factor η from 1 to 30. As shown in Fig. 14, the KMR increases as the scale factor increases, suggesting that the reconstruction capability of *ChirpKey* is improved with larger scale factor η. We also notice that when $\eta \leq 25$, KMR tends to be stable. So we choose $\eta = 25$ for the follow-up experiments and analysis.

E. Evaluation of LoRa configuration

Impact of different SF. As shown in Tab. I, the KMR of *ChirpKey* increases with the growth of SF. Specifically, the KMR increases by 2.36% and 3.9% when SF increases from 7 to 9 and from 9 to 12, respectively. The slight improvement of KMR can be explained as follows. The bit rate of LoRa can be expressed as $R_b = SF \times \frac{BW}{2^{SF}} \times CR$. Therefore, a higher spread factor can provide a lower data rate, leading to a smaller slope of the chirp frequency in LoRa signal. In the meantime, the lower data rate makes the air time of transmitting a chirp longer, which increases the duration of channel state monitoring in the LoRa packet. On average, *ChirpKey* achieves 92.86%, 94.66%, and 98.89% KMR for $SF = 7$, $SF = 9$, and $SF = 12$, respectively.

TABLE I: KMR under different LoRa configurations.

SF BW	7	9	12	Mean
125 kHz	94.33%	96.33%	99.58%	96.75%
250 kHz	92.92%	94.42%	98.76%	95.37%
500 kHz	91.32%	93.22%	98.32%	94.29%
Mean	92.86%	94.66%	98.89%	**95.47%**

Impact of different BW. From Tab. I, we can observe that the KMR of *ChirpKey* decreases slightly with the increase of BW. Specifically, the KMR drops by 1.42% and 1.13% when BW increases from $125\,\text{kHz}$ to $250\,\text{kHz}$ and from $250\,\text{kHz}$ to $500\,\text{kHz}$, respectively. The reason is the same as above. The data rate increases as BW increases, providing more time to measure the channel variations. The results show that our system can still achieve 96.75%, 95.37%, and 94.29% KMR when the bandwidth is configured to $125\,\text{kHz}$, $250\,\text{kHz}$, and $500\,\text{kHz}$, respectively, demonstrating the robustness of *ChirpKey* in different bandwidth settings.

F. Security Analysis

Eavesdropping attack. In the eavesdropping attack, Eve eavesdrops all the messages between Alice and Bob with the aim of generating the same key by statically placing it close to Alice. As discussed in Sec. V-C, the only message transmitted between legitimate users is syn, which is the compressed vector of K_A^s and K_A. We now evaluate whether Eve can use syn to deduce the same secret key. We assume that Eve has full knowledge of *ChirpKey*, so she has the ability to conduct the ℓ_1- regularized solving process as Alice and Bob. As shown in Fig. 15, Eve can only reach 50.14% and 49.67% KMR in indoor and outdoor environments, respectively. Therefore, even Eve can eavesdrop the exchanged information between Alice and Bob, she still cannot use the information to deduce the secret key. This is due to the spatial de-correlation of wireless channel, the channel measurements of Eve is different from Alice and Bob. Therefore, the measurement matrix constructed by Eve cannot be used to reconstruct the compressed vector syn of K_A. Reconstructing the syn with dissimilar measurement matrix result in wrong predictions and Eve can only achieve about 50% KMR as shown in the experiment.

Imitating attack. In the imitating attack, Eve is able to observe the mobile behavior of Alice. Then Eve tries to follow Alice's trajectory to obtain similar channel measurements with Alice with the aim of generating a similar perturbation matrix as Alice. Since path loss, shadow fading, and small-scale fading account for the majority of the channel measurements fluctuations, following Alice's trajectory will result in similar path loss and shadow fading, but not multi-path effect, which is the primary source of randomness [4], [37], Therefore, Eve cannot obtain similar channel measurements as Alice. Fig. 15 shows the KMR of Eve in indoor and outdoor environments. We can observe that the KMR of *ChirpKey* is 99.96% and 99.43% in indoor and outdoor environment, while the KMR of Eve is only 48.28% and 51.59% in indoor and outdoor environments, respectively.

Predictable channel attack. In the predictable channel attack, Eve tries to make predictable changes to the channel between Alice and Bob, with the goal of obtaining the same channel measurements as Alice and Bob. We evaluate the predictable channel attack by asking a volunteer to periodically moving with a fixed trajectory between Alice and Bob. In this way, the measured CLSSI between Alice and Bob becomes predictable to Eve. As shown in Fig 15, the KMR of Eve is only 50.18% and 47.59% in indoor and outdoor environments, respectively. The results show that due to the time-varying nature of wireless channel and multi-path effect, Eve cannot obtain similar CLSSI as Alice and Bob. Therefore, *ChirpKey* can successfully defend predictable channel attacks.

The above results show that through the three attacks Eve can generate keys with up to approximately 50% KMR, which means if we use 128-bit key for encryption, the probability of deducing the same key is extremely low, i.e., $0.5^{128} = 2.94e^{-39}$. Therefore, the fading nature of the wireless channel can guarantee the secure communication of Alice and Bob. Eve cannot obtain the same key as long as she is half wavelength away from Alice or Bob, which is a practical assumption in real-world scenarios.

G. Key Randomness

The randomness of the generated keys is verified using the NIST set of statistical tests [38]. P-values are produced by this suite to show how random the key sequence is. The randomness hypothesis is rejected if the p-value is less than 1%, indicating that the secret key is not random. We can observe from Tab. II that every p-value for all tests is greater than 1%, suggesting that the generated keys by *ChirpKey* pass the random test and have a high level of randomness.

TABLE II: NIST test.

Test	Static Indoor	Mobile Outdoor	Static Indoor	Mobile Outdoor
Freq.	0.502	0.941	0.725	0.775
Block Freq.	0.321	0.743	0.709	0.757
Cumsum (Fwd).	0.621	0.821	0.609	0.802
Cumsum (Rev).	0.475	0.743	0.744	0.687
Runs.	0.917	0.089	0.492	0.121
Longest Run of 1's.	0.155	0.349	0.669	0.811
Approx. Entropy.	0.998	1.000	0.999	1.000
FFT.	0.281	0.293	0.541	0.729
Serial.	0.766	0.329	0.124	0.623

H. Computational Overhead

We evaluate the computation time and energy consumption required for *ChirpKey* to generate a 128-bit key. We use power monitor [2] to calculate the energy consumption of end device. The computation time is calculated by using the built-in time calculation function in the debug tool. The computation time and energy consumption of different components are shown in Tab. III. Note that Alice and Bob perform different steps as end node and the gateway, respectively. In addition, the computation time for perturbation matrix generation and privacy amplification are in the order of microseconds and therefore not included in Tab. III. The results show that key generation can be completed within $0.2\,\mathrm{s}$ and require low energy consumption to generate a 128-bit key.

TABLE III: Computation overhead.

User Performance	Computation time (ms)		Energy consumption (mJ)	
Stage	Alice	Bob	Alice	Bob
Channel variance estimation	1.98	0.22	7.843	-
Compression/reconstruction	0.0108	198	0.0713	-
Total	1.9908	198.22	7.9143	-

VII. Conclusion

In this paper, we propose *ChirpKey*, a physical-layer key generation system for LoRa networks. In *ChirpKey*, we propose a number of novel methods to significantly improve the performance of key generation process. First, we design CLSSI, a LoRa-specific channel indicator to provide fine-grained and accurate channel information. Additionally, we propose a lightweight channel state estimation method to comprehensively recover the channel information. Then, we propose a novel PCS-based key delivery method to securely deliver the secret keys. Extensive evaluations show that *ChirpKey* achieves an average KMR of 99.58% and outperforms existing techniques by 11.03%–26.58%. In addition, our security analysis shows that *ChirpKey* is resistant to several common attacks. Results also show that *ChirpKey* takes less than $0.2\,\mathrm{s}$ to generate a 128-bit key, proving that *ChirpKey* can provide fast and secure wireless key generation for LoRa networks.

Acknowledgment

The work described in this paper was substantially sponsored by the project 62101471 supported by NSFC and was partially supported by the Shenzhen Research Institute, City University of Hong Kong. The work was also partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 21201420 and CityU 11201422), Shenzhen Science and Technology Funding Fundamental Research Program (Project No. 2021Szvup126), NSF of Shandong Province (Project No. ZR2021LZH010), CityU MFPRC grant 9680333, CityU SIRG grant 7020057, CityU APRC grant 9610485, CityU ARG grant 9667225 and CityU SRG-Fd grant 7005666.

[2] https://www.msoon.com/high-voltage-power-monitor

References

[1] K.-A. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, 2015.

[2] X. Ge, J. Yu, H. Zhang, C. Hu, Z. Li, Z. Qin, and R. Hao, "Towards achieving keyword search over dynamic encrypted cloud data with symmetric-key based verification," *IEEE TDSCM*, 2019.

[3] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE TAP*, 2005.

[4] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *ACM Mobicom*, 2009.

[5] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *ACM Mobicom*, 2008.

[6] W. Xi, C. Qian, J. Han, K. Zhao, S. Zhong, X.-Y. Li, and J. Zhao, "Instant and robust authentication and key agreement among mobile devices," in *ACM CCS*, 2016.

[7] L. Jiao, N. Wang, P. Wang, A. Alipour-Fanid, J. Tang, and K. Zeng, "Physical layer key generation in 5g wireless networks," *IEEE WC*, 2019.

[8] W. Xu, S. Jha, and W. Hu, "Lora-key: Secure key generation system for lora-based network," *IEEE IoT-J*, 2018.

[9] J. Gao, W. Xu, S. Kanhere, S. Jha, J. Y. Kim, W. Huang, and W. Hu, "A novel model-based security scheme for lora key generation," in *ACM/IEEE IPSN*, 2021.

[10] A. K. Junejo, F. Benkhelifa, B. Wong, and J. A. Mccann, "Lora-lisk: A lightweight shared secret key generation scheme for lora networks," *IEEE IoT-J*, 2021.

[11] H. Yang, H. Liu, C. Luo, Y. Wu, W. Li, A. Y. Zomaya, L. Song, and W. Xu, "Vehicle-key: A secret key establishment scheme for lora-enabled iov communications," in *IEEE ICDCS*, 2022.

[12] H. Ruotsalainen, J. Zhang, and S. Grebeniuk, "Experimental investigation on wireless key generation for low-power wide-area networks," *IEEE IoT-J*, 2019.

[13] Y. Ren, L. Liu, C. Li, Z. Cao, and S. Chen, "Is lorawan really wide? fine-grained lora link-level measurement in an urban environment," in *IEEE ICNP*, 2022.

[14] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *IEEE INFOCOM*, 2013.

[15] S. N. Premnath, P. L. Gowda, S. K. Kasera, N. Patwari, and R. Ricci, "Secret key extraction using bluetooth wireless signal strength measurements," in *IEEE SECON*, 2014.

[16] Q. Wang, H. Su, K. Ren, and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *IEEE INFOCOM*, 2011.

[17] R. Dautov and G. R. Tsouri, "Establishing secure measurement matrix for compressed sensing using wireless physical layer security," in *IEEE ICNC*, 2013.

[18] W. Xue, C. Luo, G. Lan, R. Rana, W. Hu, and A. Seneviratne, "Kryptein: a compressive-sensing-based encryption scheme for the internet of things," in *ACM/IEEE IPSN*, 2017.

[19] Q. Lin, W. Xu, J. Liu, A. Khamis, W. Hu, M. Hassan, and A. Seneviratne, "H2b: Heartbeat-based secret key generation using piezo vibration sensors," in *ACM/IEEE IPSN*, 2019.

[20] C. Li and Z. Cao, "Lora networking techniques for large-scale and long-term iot: A down-to-top survey," *ACM CSUR*, 2022.

[21] Z. Sun, H. Yang, K. Liu, Z. Yin, Z. Li, and W. Xu, "Recent advances in lora: A comprehensive survey," *ACM TOSN*, 2022.

[22] N. Hou, X. Xia, and Y. Zheng, "Cloaklora: A covert channel over lora phy," *IEEE/ACM ToN*, 2022.

[23] Hou, Ningning and Xia, Xianjin and Zheng, Yuanqing, "Jamming of lora phy and countermeasure," in *IEEE INFOCOM*, 2021.

[24] S. Tomasin, S. Zulian, and L. Vangelista, "Security analysis of lorawan join procedure for internet of things networks," in *IEEE WCNCW*, 2017.

[25] J. Kim and J. Song, "A dual key-based activation scheme for secure lorawan," *WCMC*, 2017.

[26] X. Wang, L. Kong, Z. Wu, L. Cheng, C. Xu, and G. Chen, "Slora: towards secure lora communications with fine-grained physical layer features," in *ACM SenSys*, 2020.

[27] A. Al-Shawabka, P. Pietraski, S. B. Pattar, F. Restuccia, and T. Melodia, "Deeplora: Fingerprinting lora devices at scale through deep learning and data augmentation," in *ACM MobiHoc*, 2021.

[28] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for lora using spectrogram and cnn," in *IEEE INFOCOM*, 2021.

[29] W. Xu, Z. Li, W. Xue, X. Yu, B. Wei, J. Wang, C. Luo, W. Li, and A. Y. Zomaya, "Inaudiblekey: Generic inaudible acoustic signal based key agreement protocol for mobile devices," in *ACM/IEEE IPSN*, 2021.

[30] H. Liu, J. Yang, Y. Wang, Y. J. Chen, and C. E. Koksal, "Group secret key generation via received signal strength: Protocols, achievable rates, and implementation," *IEEE TMC*, 2014.

[31] H. Liu, Y. Wang, Y. Ren, and Y. Chen, "Bipartite graph matching based secret key generation," in *IEEE INFOCOM*, 2021.

[32] C. A. Hall and W. W. Meyer, "Optimal error bounds for cubic spline interpolation," *J. Approx. Theory*, 1976.

[33] B. Wei, W. Xu, K. Li, C. Luo, and J. Zhang, "i2key: A cross-sensor symmetric key generation system using inertial measurements and inaudible sound," in *ACM/IEEE IPSN*, 2022.

[34] R. Arablouei, "Fast reconstruction algorithm for perturbed compressive sensing based on total least-squares and proximal splitting," *Signal Process*, 2017.

[35] M. A. Herman and T. Strohmer, "General deviants: An analysis of perturbations in compressed sensing," *IEEE J-STSP*, 2010.

[36] Z. Yang, W. Yan, and Y. Xiang, "On the security of compressed sensing-based signal cryptosystem," *IEEE TETC*, vol. 3, no. 3, pp. 363–371, 2015.

[37] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE TMC*, 2010.

[38] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," DTIC Document, Tech. Rep., 2001.