DyWCP: Dynamic and Lightweight Data-Channel Coupling towards Confidentiality in IoT Security

Shengping Bi
sbi@nmsu.edu
New Mexico State University

Tao Hou
taohou@usf.edu
University of South Florida

Tao Wang
taow@nmsu.edu
New Mexico State University

Yao Liu
yliu@cse.usf.edu
University of South Florida

Zhuo Lu
zhuolu@usf.edu
University of South Florida

Qingqi Pei
qqpei@mail.xidian.edu.cn
Xidian University

ABSTRACT

As Internet of Things (IoT) is more and more pervasive and deployed in critical applications, it's becoming increasingly important to preserve the confidentiality of sensitive data when IoT devices communicate with each other. However, traditional cryptography is usually time and energy consuming. It may not be applicable to IoT devices with limited computational capability or limited power. In this paper, we propose a lightweight encryption scheme named Dynamic Wireless Channel Pad (DyWCP) inspired by one-time pad encryption. One-time pad encryption achieves perfect secrecy but has been rarely used in practice due to the inconvenience of key negotiation. Our research discovers that in the wireless context it is possible to design a one-time pad encryption scheme without key negotiation. Towards the realization of DyWCP, we create techniques to utilize the additive feature of wireless channel to encrypt messages, to integrate modular operations at wireless physical layer, and to defend against multiple eavesdroppers. We implement a prototype of the proposed scheme using Universal Software Defined Radio Peripherals (USRP), and conduct a suite of experiments to evaluate the performance of the proposed scheme.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

Wireless Security, IoT Confidentiality, Data-channel Coupling, Lightweight Encryption

ACM Reference Format:
Shengping Bi, Tao Hou, Tao Wang, Yao Liu, Zhuo Lu, and Qingqi Pei. 2022. DyWCP: Dynamic and Lightweight Data-Channel Coupling towards Confidentiality in IoT Security. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3507657.3528550

WiSec '22, May 16–19, 2022, San Antonio, TX, USA.

ACM ISBN 978-1-4503-9216-7/22/05…$15.00
https://doi.org/10.1145/3507657.3528550

1 INTRODUCTION

With the evolvement of sophisticated and low-cost chips and the ubiquity of wireless networks, Internet of Things (IoT) has been extensively deployed in various domains, including smart home, wearable, health care, and manufacturing [1]. However, it also incurs a range of security concerns [2–4]. As IoT devices may gather geographical information, monitor users' privacy activities, and record clients' biometric features [5], one of the critical concerns is how to preserve the confidentiality of such sensitive data. Specifically, because of the broadcast nature of wireless signal, conversation between IoT devices are usually vulnerable to eavesdropping attacks [6]. It is essential to secure the communication between IoT devices.

Intuitively, cryptography encryption methods can be applied to encrypt all the conversation between IoT devices. However, as IoT devices are usually featured with limited computational capacity and limited power, they may not afford expensive cryptography operations by conventional encryption methods like AES or RSA. For example,

- Implantable medical devices (IMDs): Modern IMDs usually come with wireless connectivity to allow remote monitoring of a patient's vital signs. The wireless connectivity should be protected to prevent unauthorized accesses and data transmissions. Since IMDs are battery powered, simply applying conventional cryptography may considerably reduce life time of IMDs.
- Energy harvesting devices: Featured with low-end computational chips or only integrated with RFID tags, energy harvesting devices usually cannot support compute-intensive cryptography encryption schemes.

To cope with the limited resources of IoT devices, multiple lightweight schemes have been proposed to achieve efficient data encryption. While most designs gain energy efficiency by a security trade-off (e.g., smaller key size, simplified key schedule and more elementary operations) [7–9]. Other designs, such as [10–14], utilize either customized hardwares, or specialized softwares to facilitate their encryption schemes for IoT applications. However, such designs also limit their possible deployment scenarios.

Alternatively, we propose a novel encryption scheme, named Dynamic Wireless Channel Pad (DyWCP), that takes advantage of dynamic signal variation in wireless context to achieve the confidentiality of sensitive data. The scheme has four features: 1) Lightweight, the scheme only consumes few computational resources.

Encryption works along with the baseband equalization processing, while decryption is hassle-free without any computational consumption; 2) Secure, the scheme is inspired by the one-time pad encryption, simple but with high secrecy; 3) Compatible, the scheme works independently from application layer. It is complementary to traditional cryptography scheme and can work together to further improve the security. 4) Ubiquitous, the scheme can be applied to most IoT applications with wireless connectivity.

DyWCP is inspired by one-time pad encryption. Among existing encryption schemes, one-time pad encryption is a typical lightweight method, because it only uses basic modular, exclusive or, or addition operations. Although lightweight and simple, one-time pad encryption has been proven of the perfect secrecy property [15]. Nevertheless, one-time pad encryption has been rarely used in practice since it was invented several decades ago. It is believed that there is no point in using one-time pad encryption, because the encryption requires a key to be of the same length as an original message. If one canfi nd a way to pass the key in a secret way, then one should also be able to directly send the message in the same secret way without encryption [16].

In this research, by proposing DyWCP, we make one-time pad encryption a practical method to protect IoT applications. Key negotiation creates the essential hurdle of applying one-time pad encryption, which requires a key to be used only once and different messages should be encrypted by different keys. If a system generates thousands of messages, the same number of keys should be generated and negotiated. In our design, DyWCP aims to address this hurdle by completely removing the step of key negotiation to enable the practical use of one-time pad encryption.

Although this goal may be difficult to achieve under a generic situation, our investigation found that in the wireless context it is possible to design a one-time pad encryption scheme without key negotiation. We note that the encryption of one-time pad schemes can adopt modular addition operations. We also note that wireless channel is additive, i.e., wireless signals sent from different antennas add up at a receiver. Intuitively, DyWCP can utilize the additive feature of wireless channel towards the realization of a one-time pad scheme. For example, Alice transmits $d(t)+k(t)$ and $d(t)-k(t)$ using two antennas respectively, where $d(t)$ and $k(t)$ are original and random signals respectively. Wireless channel is additive and wireless signals sent from different antennas add up at a receiver. Thus, Bob receives $d(t)+k(t)+d(t)-k(t)$, which is equal to $2d(t)$. Bob can easily obtain the original signal $d(t)$ by simply dividing received message by 2 and $k(t)$ is canceled at Bob.

Nevertheless, a closer examination reveals that this example design does not accomplish one-time pad encryption, since it seems that an eavesdropper is also able to decrypt a message as long as it is located within Alice's signal coverage range. However, in reality, neither Bob or the eavesdropper can receive the original signal by using the example design.

Specifically, wireless signals are distorted when they propagate in the air. The distortion imposed by wireless channel can be usually considered as multiplicative [17]. A received signal can thus be represented by $d(t)h(t)$, where $h(t)$ is the channel distortion. A process called channel estimation is utilized by the receiver to remove the channel distortion and enable a receiver to correctly decode messages. For Alice with two antennas, we use $h_1(t)$ and $h_2(t)$ to denote the channel distortions imposed to the signals transmitted by two antennas respectively. Therefore, the signal received by Bob is indeed $h_1(t)(d(t)+k(t))+h_2(t)(d(t)-k(t)) = (h_1(t)+h_2(t))d(t)+(h_1(t)-h_2(t))k(t)$. Signals sent from different antennas experience distinct channel distortions due to the different reflection, refraction, and diffraction introduced by different propagation paths [18]. Hence, $h_1(t) \neq h_2(t)$ and $k(t)$ is not canceled at Bob. By utilizing channel estimation technique, Bob can know $h_1(t)$ and $h_2(t)$ but he is still unable to decode the message, because $k(t)$ is random and $(h_1(t)-h_2(t))k(t)$ is indeed a random variable to Bob. For the same reason, the eavesdropper cannot decode the message either.

We need to refine the DyWCP design to equalize the channels (i.e., channel distortions) between Alice and Bob, such that $k(t)$ can be canceled at Bob but not at the eavesdropper. Accordingly, we propose to create a dynamic channel cipher, which is customized based on the channels between Alice and Bob and can enable the cancelation of $k(t)$ at Bob but not at an eavesdropper. With the refined design, DyWCP can achieve one-time pad encryption by simply using different key signal $k(t)$ to encrypt different original signals. The negotiation of $k(t)$ is not required for encryption and decryption, since $k(t)$ is canceled at Bob. Nevertheless, DyWCP still face the following challenges to achieve a practical one-time pad encryption scheme.

First, the previous discussion omits the steps of modular addition to facilitate understanding and presentation. However, modular operations are always essential and important to one-time pad encryption. Towards modular addition, it seems that we can directly add the i-th bit of an original message to the i-th bit of a key, modulo the addition result by a modulus of 2 (a binary bit is either 1 or 0), and then convert the modular output from a discrete bit sequence to a continuous signal for transmission over the wireless channel. This intuitive method, however, cannot be adopted by a practical wireless communication system, because the transmission unit at wireless physical layer is a symbol instead of a binary bit. Hence, we need to perform modular addition at symbol level instead of binary bit level. Otherwise, the key signals may not be canceled and consequently the receiver cannot recover the original message (an example about the decoding failure is given in Section 5).

Second, it has been demonstrated that multiple eavesdroppers may launch known plaintext attacks tofi nd the channels between Alice and Bob [19]. With the channel information, they canfigure out the key signal $k(t)$, andfi nally decode the original signal $d(t)$. This collaboration attack requires attackers to estimate wireless channel from predefined information like training sequences, preambles or synchronization codes in a message [17]. We thus need to seek methods that can disable the capability of the eavesdroppers to estimate the wireless channel from the predefined information. Ideally, we would like to 'poison' the wireless channel estimated by the eavesdropper, such that the eavesdroppers always obtain fake and deceptive channels that are quite different from the original one.

In DyWCP, we propose approaches to address these challenges, and our contributions are summarized below:

(1) We propose a lightweight encryption scheme that only requires simple modular additions and channel estimations.

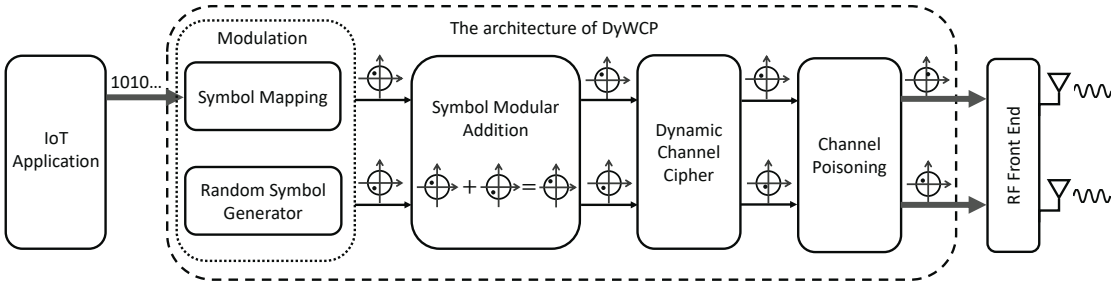

Figure 1: An overview of DyWCP.

(2) We propose the design of an dynamic channel cipher that can enable the cancelation of $k(t)$ at Bob.
(3) We propose a symbol level modular addition method that adds symbols at the wireless physical layer to allow the successful recovery of original signals.
(4) We propose a channel 'poisoning' approach to prevent cooperative eavesdroppers from deriving the channels between target communicators.
(5) We implement a prototype of the proposed scheme using USRPs [20], and conduct a suite of experiments to assess the effectiveness of the proposed scheme.

2 RELATED WORK

In this section, we summarize the literature related to our research in this paper.

Lightweight Cryptography Encryption: There exist multiple lightweight cryptography algorithms that enforce the security for low-resource IoT devices [7–14, 22]. To achieve the energy efficiency, these designs tend to adopt a smaller key size, rely more on elementary operations (e.g., XOR, AND) or adopt a simplified key schedule. For example, a lightweight block cipher named PRESENT is proposed in [22], adopting a simple SPN network with key sizes of 80 or 128 bits. The authors in [9] design a hardware oriented block cipher, KATAN, which follows a Feistel structure with a simplified key scheduling mechanism. Another lightweight block cipher, TWINE is presented in [7]. It employs the Type-2 generalized Feistel structure (GFS) to achieve hardware efficiency while minimizing the hardware-oriented design. Other designs, such as [10–14], utilize either customized hardwares, or specialized softwares to facilitate their encryption schemes. However, such designs also limit their possible deployment scenarios. DyWCP is inspired by one-time pad encryption, simple but with a high secrecy. Meanwhile, DyWCP can be ubiquitously applied to most IoT applications with wireless connectivity. It takes advantage of wireless physical layer property and works independently from application layer. Thus DyWCP is complementary to existing lightweight cryptography algorithms and can work together to further improve the security of IoT devices.

Physical Layer Security: There exist recent works that achieve secret wireless communication by utilizing zero-forcing beamforming techniques and multi-user, multiple-input, multiple-output (MU-MIMO) systems [23–26]. The basic idea of these work is to add artificial noise to an original message, and encode a transmit signal based on the channel between a transmitter and a receiver, such that the noise is canceled at the receiver but not at an eavesdropper.

It seems that these works are similar to the proposed one. Nevertheless, there are two essential differences between them. First, the existing work [23–26] encrypts an original message using general arithmetic and an original message is directly added to a noise signal. However, for one-time pad encryption, it is essential to utilize modular arithmetic to avoid information leakage and fully preserve confidentiality of original messages [27]. The scheme proposed in this paper applies modular arithmetic, and we create a symbol-level modular addition method to enable the incorporation of the proposed scheme into practical wireless systems.

Second, [19] demonstrated that the encryption schemes proposed in the existing work are vulnerable to the known plaintext attack. Specifically, cooperative eavesdroppers may utilize publicly known information of a message to infer the channels between a transmitter and a receiver, and then use the inferred channels to decode the original messages. The proposed scheme is resilient against such attacks. We propose the channel poisoning method to alter the channels during the transmission time of the publicly known information. In this way, the eavesdroppers can only obtain poisoned channels that are quite different from real ones, and thus fail to decode original messages.

3 THREAT MODEL AND SYSTEM OVERVIEW

We consider a transmitter of multiple antennas and assume that all antennas of the transmitter share the same external clock source. To ensure signals transmitted from different antennas can arrive at the receiver simultaneously, we utilize the widely-used reference broadcast synchronization to compensate the processing and transmission delay difference between the receiver and antennas [28**?** , 29].

We assume antennas are separated by enough distance that channels between the receiver and each antenna are uncorrelated. Theoretically, when two transmit antennas are separated by more than a half wavelength, a receiver can observe uncorrelated channels from two antennas. A wireless channel is normally static for a short time, which is referred to as the coherence time [30]. We assume that a packet can be transmitted within the coherence time of the channel.

Threat Model: We focus on preventing an original message from being retrieved by eavesdroppers. We consider two types of eavesdroppers. First, we consider a traditional eavesdropper with a single antenna. Such an eavesdropper can be at any location except

for the exact position of the receiver or transmitter. Second, we consider cooperative eavesdroppers with multiple antennas. In this scenario, multiple eavesdroppers at different positions collaborate to infer transmit messages.

System Overview: We incorporate DyWCP into physical layer processing to enable data encryption. As shown in Figure 1, DyWCP consists of four modules: 1) Modulation: it converts incoming bit sequence into continuous wireless symbols, and produces arbitrary keys for symbol encryption. 2) Symbol modular addition: it performs symbol wise modular addition to encrypt incoming messages by the random generated keys. 3) Dynamic channel cipher: it calibrates encrypted symbols to accommodate channel variation ensuring that the key always cancels at receiver but remains at an eavesdropper. 4) Channel poisoning: it disables the capability of the eavesdroppers to estimate the wireless channel from the predefined information.

Since symbol mapping and random symbol generator can be easily implemented by existing techniques [31, 32], we focus on the implementation of the rest three modules. To facilitate understanding, wefi rst introduce dynamic channel cipher in Chapter 4 and then describe details of symbol modular addition in Chapter 5. In Chapter 6 and 7, we present a comprehensive security analysis in the scenario of both single and cooperative eavesdroppers. We also illustrate the design of channel poisoning in Chapter 7. Experiment and evaluation are discussed in Chapter 8.

4 DYNAMIC CHANNEL CIPHER

The purpose of the dynamic channel cipher is to ensure that the key $k(t)$ always cancels at Bob but remains at an eavesdropper. In this section, we propose the method for constructing the dynamic channel cipher.

4.1 Mathematical Modeling

We propose to design the dynamic channel cipher as multiplicative, since the channel distortion is multiplicative to wireless signals. Figure 2 shows an example of the proposed scheme for Alice with two antennas. The dynamic channel cipher has cipher characteristic functions of $f_1(t)$ and $f_2(t)$ for thefi rst and second antennas, respectively. After passing the multiplicative channel cipher, the transmit signal becomes $f_1(t)(d(t) + k(t))$ and $f_2(t)(d(t) - k(t))$.

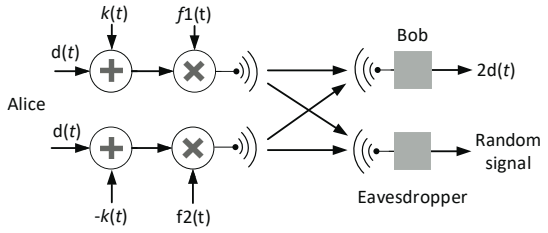

Figure 2: Basic idea of dynamic channel cipher.

Assume Alice has N antennas. Let $\mathbf{F} = [f_1(t), f_2(t), ..., f_N(t)]$ denote the dynamic channel cipher with cipher characteristic function $f_n(t)$ for the n-th antenna ($1 \le n \le N$). We denote the message and key processed by $f_n(t)$ and sent by the n-th antenna as $d'_n(t)$ and $k'_n(t)$ respectively. The dynamic channel cipher is multiplicative, and hence $d'_n(t) = d_n(t)f_n(t)$ and $k'_n(t) = k_n(t)f_n(t)$, where $d_n(t)$ and $k_n(t)$ are the original message and the key associated with the n-th antenna. The actual signal transmitted by the n-th antenna is $d'_n(t) + k'_n(t)$.

Let $h_n(t)$ denote the channel between Bob and the n-th antenna of Alice. Bob receives $\sum_1^N h_n(t)[d'_n(t) + k'_n(t)]$. As mentioned earlier, a wireless channel can be measured by a process called channel estimation. Common channel estimation algorithms include Least-square and MMSE estimation [33, 34]. The channel estimation process normally needs to be performed periodically, such that the communicators can cope with the channel changes.

4.2 Construction Condition

We expect Bob to receive an original message $d(t)$. Moreover, we expect the component that consists of key information is canceled at Bob. We thus need equations $\sum_1^N h_n(t)d'_n(t) = \sum_1^N d_n(t)$ and $\sum_1^N h_n(t)k'_n(t) = 0$ to hold. Equivalently, the following equation should hold.

$$\begin{cases} d_1(t)f_1(t)h_1(t) + ... + d_N(t)f_N(t)h_N(t) = d(t) \\ k_1(t)f_1(t)h_1(t) + ... + k_N(t)f_N(t)h_N(t) = 0 \end{cases}$$

Let $h_n(t)' = f_n(t)h_n(t)$, $\mathbf{H'} = [h_1(t)', ..., h_N(t)']$, and $\mathbf{K} = [k_1(t), ..., k_N(t)]$. To make the above equation to hold, we need $\mathbf{K} \cdot \mathbf{H'} = k_1(t)h_1(t)' + ... + k_N(t)h_N(t)' = 0$. This means that $\mathbf{K}$ is orthogonal to $\mathbf{H'}$. We use $\perp$ to denote the orthogonal relationship, and the construction of the dynamic channel cipher requires $\mathbf{K} \perp \mathbf{H'}$. Further let $\mathbf{D} = [d_1(t), ..., d_N(t)]$. We also need $\mathbf{D} \cdot \mathbf{H'} = d_1(t)h_1(t)' + ... + d_N(t)h_N(t)' = d(t)$. In MIMO system, the data vector $\mathbf{D}$ is usually chosen to be co-directional with the channel vector $\mathbf{H'}$ to achieve the maximum amplitude at the receiver [35]. We use $\parallel$ to denote the co-directional relationship, and the construction of the dynamic channel cipher also requires $\mathbf{D} \parallel \mathbf{H'}$. We refer to $\mathbf{K}$ and $\mathbf{D}$ as the key vector and message vector respectively. From $\mathbf{K} \perp \mathbf{H'}$ and $\mathbf{D} \parallel \mathbf{H'}$, we conclude that $\mathbf{D} \perp \mathbf{K}$, i.e., the message vector should be orthogonal to the key vector.

4.3 Constructing the Dynamic Channel Cipher

Our ultimate goal is tofi nd the cipher characteristic functions $f_1(t)$, $f_2(t)$,..., $f_N(t)$. For $f_n(t)$, it must satisfy the equation $h_n(t)' = f_n(t)h_n(t)$. In an ideal case, with the knowledge of $h_n(t)'$ and $h_n(t)$, the cipher characteristic function $f_n(t)$ for the n-th transmit antenna can be computed by $f_n(t) = h_n(t)'/h_n(t)$.

The channel $h_n(t)$ can be measured by the channel estimation process, and $h_n(t)'$ can be determined based on the aforementioned construction condition that $\mathbf{K} \perp \mathbf{H'}$ and $\mathbf{D} \parallel \mathbf{H'}$, where $\mathbf{H'} = [h_1(t)', ..., h_N(t)']$. Note that this condition also implies that $\mathbf{D} \perp \mathbf{K}$. Accordingly, instead offi nding a $\mathbf{H'}$ that needs to satisfy both the orthogonal and co-directional relationships, we can relax the requirement for $\mathbf{H'}$ byfi rstfi xing the relationship between $\mathbf{D}$ and $\mathbf{K}$, and thenfi nding a $\mathbf{H'}$ that satisfies either the orthogonal relationship with $\mathbf{K}$, or the co-directional relationship with $\mathbf{D}$. Without loss of generality, wefi nd a $\mathbf{H'}$ to satisfy the co-directional relationship with $\mathbf{D}$.

We would like $\mathbf{D}$ and $\mathbf{K}$ to be orthogonal. Towards this end, we randomly generate a pair of orthogonal vectors, each of which

is of length N and consists of all positive numbers. Such vectors can be generated by using well-known mathematical tools like the Gram-Schmidt algorithm. Assume that both orthogonal vectors are represented by $\mathbf{u} = [u_1, u_2, ..., u_N]$ and $\mathbf{v} = [v_1, v_2, ... v_N]$. We then set $\mathbf{D} = \mathbf{u}d(t)$ (i.e., $\mathbf{D} = [u_1 d(t), u_2 d(t), ..., u_N d(t)]$) and $\mathbf{K} = \mathbf{v}k(t)$ (i.e., $\mathbf{K} = [v_1 k(t), v_2 k(t), ..., v_N k(t)]$), or $\mathbf{D} = \mathbf{v}d(t)$ and $\mathbf{K} = \mathbf{u}k(t)$. Because $\mathbf{u}$ and $\mathbf{v}$ are orthogonal, $\mathbf{D}$ and $\mathbf{K}$ are also orthogonal as shown below.

$$\mathbf{DK} = (u_1 v_1 + u_2 v_2 + ... + u_N v_N) d(t) k(t) = 0$$

Without loss of generality, we let $\mathbf{D} = [u_1 d(t), u_2 d(t) ..., u_N d(t)]$. As discussed, $\mathbf{H}'$ should be co-directional to $\mathbf{D}$. This can be achieved by setting $\mathbf{H}' = \frac{\mathbf{u}}{||\mathbf{u}||}$, where $||\mathbf{u}|| = u_1^2 + u_2^2 + ... + u_N^2$. For example, assume that $\mathbf{u} = [1, 1, \sqrt{2}]$ and $\mathbf{D} = [d(t), d(t), \sqrt{2}d(t)]$. $\mathbf{H}'$ can be set as $[\frac{1}{4}, \frac{1}{4}, \frac{\sqrt{2}}{4}]$. We can see that elements in $\mathbf{H}'$ are proportional to those in $\mathbf{D}$, i.e., $d(t) : d(t) : \sqrt{2}d(t) = \frac{1}{4} : \frac{1}{4} : \frac{\sqrt{2}}{4} = 1 : 1 : \sqrt{2}$. We can further verify that $\mathbf{DH}' = d(t)$.

5 SYMBOL LEVEL MODULAR ADDITION

Our discussion so far has omitted modular addition to facilitate understanding. Note that modular addition is an essential and indispensable step of one-time pad encryption. In this section, we propose methods that can achieve modular addition at wireless physical layer.

In the following discussion, to facilitate understanding, wefirst assume ideal wireless communication channel that does not distort wireless signals, and the transmitter directly transmits signals without multiplying them with dynamic channel cipher. We assume the transmitter has two antennas, transmitting (message + key) and (message - key), respectively.

5.1 Modular Addition at Symbol Level

Intuitively, we can directly perform bit-wise modular addition on the original message. Specifically, we add the i-th bit of an original message to the i-th bit of a key, and then modulo the addition result by a modulus. This intuitive method, however, cannot be adopted by a practical wireless communication system, because it does not consider the wireless physical layer modulation, which is indispensable for modern wireless systems.

The transmission unit at the wireless physical layer is a symbol instead of a binary bit. The purpose of the wireless physical layer modulation is to convert information binary bits into symbols. We focus on 2-dimensional modulation like QPSK, 16-QAM, 64-QAM, and 256-QAM, because they are dominantly used by existing wireless systems [36]. In 2-dimension modulation, a group of binary information bits are usually mapped to a coordinates on a 2-dimension plane, which is referred to as a constellation [17]. Figure 3 gives an example of a 16-QAM constellation.

Because the wireless physical layer transmits original messages in a symbol-by-symbol way rather than a bit-by-bit way, we need to perform modular addition at symbol level.

5.2 The Naive Method

Assume that we obtain a sequence of symbols s_1, s_2,...,s_n after modulation, and we randomly generate a key signal denoted by

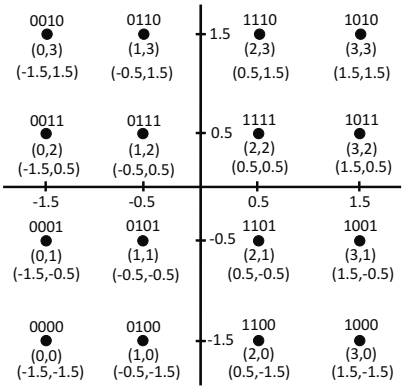

Figure 3: Bit sequence mapping for 16-QAM.

a sequence k_1, k_2,...,k_n of symbols. In a naive method, we can directly perform the symbol level modular addition by calculating $s_i + k_i \bmod m$ and $s_i - k_i \bmod m$ for all $1 \le i \le n$. Specifically, let $s_i = (a_i, b_i)$ and $k_i = (c_i, d_i)$. $s_i + k_i \bmod m$ can be calculated by $((a_i, b_i) + (c_i, d_i)) \bmod m = ((a_i + c_i) \bmod m, (b_i + d_i) \bmod m)$, and $s_i - k_i \bmod m$ can be calculated by $((a_i - c_i) \bmod m, (b_i - d_i) \bmod m)$.

Note that the modular addition result $((a_i + c_i) \bmod m, (b_i + d_i) \bmod m)$ will be transmitted to the channel and should be a valid symbol on the constellation. For example, for the 16-QAM shown in Figure 3, the x-coordinates of all symbols form the set $\mathcal{X} = \{-1.5, -0.5, 0.5, 1.5\}$, and the y-coordinates of all symbols form the set $\mathcal{Y} = \{-1.5, -0.5, 0.5, 1.5\}$. After modular addition, we need to enable $(a_i + c_i) \bmod m \in \mathcal{X}$, and $(b_i + d_i) \bmod m \in \mathcal{Y}$. Similarly, we need to enable $(a_i - c_i) \bmod m \in \mathcal{X}$, and $(b_i - d_i) \bmod m \in \mathcal{Y}$. However, this requirement is difficult to achieve, because the sets $\mathcal{X}$ and $\mathcal{Y}$ are not closed under the modular addition operation (a set is called closed under an operation if that operation returns a member of the set when evaluated on members of the set [37]).

In this example, we can see that $\mathcal{X} = \mathcal{Y}$. Indeed, for all aforementioned 2-dimensional modulations, the set formed by x-coordinates and that formed by y-coordinates of all symbols are identical. To facilitate presentation, in the following, we refers to the set $\mathcal{X}$ only, since $\mathcal{X} = \mathcal{Y}$.

5.3 The Indexing Method

The naive method fails, because the set $\mathcal{X}$ is not closed under the modular addition operation. Intuitively, if we can revise the naive method in a way that we can deal with a set that is closed under the modular addition, then we may be able to calculate symbol level modular addition. We observe that the consecutive integer set $\mathcal{Z} = \{0, 1, 2, 3, ...\}$ is closed under the modular addition with a modulus of $||\mathcal{Z}||$, where $||\mathcal{Z}||$ denotes the number of elements in $\mathcal{Z}$. This observation inspires us to revise the naive method byfirst mapping the elements of the set $\mathcal{X}$ to the consecutive integer set $\mathcal{Z}$, and then performing modular addition on the set $\mathcal{Z}$.

Specifically, we achieve this mapping by indexing the elements of the set $\mathcal{X}$. For example, for the 16-QAM modulation, $\mathcal{X} = \{-1.5, -0.5, 0.5, 1.5\}$, and we index them by 0, 1, 2, and 3, respectively.

Thus, $\mathcal{Z} = \{0, 1, 2, 3\}$ and $||\mathcal{Z}|| = 4$. We give an example to illustrate how the indexing method works.

An example of indexing method: Assume that Alice would like to encrypt information bits 1101 using the 16-QAM modulation. Alicefi rstfi nds that the symbol for 1101 is (0.5, -0.5) on constellation, and the corresponding index is (2,1). Alice then randomly chooses a symbol (1.5, -0.5) as the key, and the corresponding index is (3, 1). The modulus $m = ||\mathcal{Z}|| = 4$. Alice computes ((2,1) + (3,1)) mod 4 = (1, 2), and ((2,1) - (3,1)) mod 4 = (3, 0). The symbols with indexes of (1, 2) and (3, 0) are (-0.5, 0.5) and (1.5, -1.5), respectively. Alice transmits each of the two symbols with its own antenna. Since an ideal channel does not introduce distortion to radio signals, Bob thus receives the sum of two symbols, i.e., (-0.5, 0.5) + (1.5, -1.5) = (1, -1). Bob divides the received content by 2 (the number of antennas) and the result is (0.5, -0.5), which will be demodulated as the original message 1101.

5.4 Correctness Analysis and Decoding Shift

We point out that the indexing method may fail under some situations. As a result, the sum of symbols received by Bob does not generate original bits. For example, if the previously discussed symbol of index (2,1) is encrypted by a symbol of index (3,2), then symbols with indexes (1,3) and (3,3) are transmitted, i.e., (-0.5, 1.5) and (1.5, 1.5). The received symbol is (0.5,1.5), which is demodulated into 1110 instead of the original bits 1101. In the following discussion, we investigate the correctness of the indexing methods and propose advanced techniques to improve this method.

5.4.1 Correctness Analysis. Let $(In(d)+In(k))$ mod m and $(In(d)-In(k))$ mod m denote the indexes of transmit symbols at two antennas respectively, where d and k denote the message and the key symbols, and $In(d)$ and $In(k)$ denote the corresponding indexes, e.g., $In(0.5, -0.5) = (2,1)$.

Lemma5.1. *The index of the symbol received by Bob is (1/2)*$((In(d) + In(k))\ mod\ m + (In(d) - In(k))\ mod\ m)$.

Proof: Let s_1 and s_2 denote the symbols sent by two antennas respectively. We have $In(s_1) = (In(d) + In(k))$ mod m, and $In(s_2) = (In(d) - In(k))$ mod m. Note that the coordinate of a symbol can be represented by the index of this symbol subtracting a constant coordinate. For example, (0.5, -0.5) = $In(0.5, -0.5)$ - (1.5, 1.5). Let (δ,δ) denote the constant coordinate. We thus have $s_1 = In(s_1)$ - (δ,δ) and $s_2 = In(s_2)$ - (δ,δ). Bob receives $s_1 + s_2 = In(s_1) + In(s_2)$ - $2(\delta,\delta)$. Bob divides the received content by 2 and the received symbol is $1/2(In(s_1) + In(s_2))$ - (δ,δ). The corresponding index of the received symbol is $1/2(In(s_1) + In(s_2)) = 1/2((In(d) + In(k))$ mod $m + (In(d) - In(k))$ mod $m)$. □

Let $In(d)_x$ and $In(d)_y$ denote the x and y coordinate of $In(d)$. Further let $In(k)_x$ and $In(k)_y$ denote the x and y coordinate of $In(k)$. We classify the conditions for correct and incorrect decoding of the x coordinate in Table 1. For the decoding of the y coordinate, the same conditions apply.

5.4.2 Decoding Shift. Incorrect decoding happens for the last two conditions given in Table 1. When incorrect decoding happens, the x or y coordinate of the index of received symbol is shifted by $\pm\frac{1}{2}m$ from that of the original message symbol. This shift is caused by the unbalanced modulo operations on the indexes of transmit symbols.

Table 1: Decoding results of x coordinate.

Condition	Result
$0<In(d)_x+In(k)_x<m \&\& 0<In(d)_x-In(k)_x<m$	$In(d)_x$
$In(d)_x+In(k)_x>m \&\& In(d)_x-In(k)_x<0$	$In(d)_x$
$In(d)_x+In(k)_x>m \&\& 0<In(d)_x-In(k)_x<m$	$In(d)_x-\frac{1}{2}m$
$0<In(d)_x+In(k)_x<m \&\& In(d)_x-In(k)_x<0$	$In(d)_x+\frac{1}{2}m$

5.5 Dealing with Decoding Shift

When the decoding shift happens, the x or y coordinate of the index of a received symbol is decoded as $In(d)_x \pm \frac{1}{2}m$ or $In(d)_y \pm \frac{1}{2}m$. Intuitively, if we canfi nd a way such that the x and y coordinates of the index of a received symbol are decoded as $In(d)_x \pm m$ and $In(d)_y \pm m$. We can then apply modular operation to remove decoding shift.

Towards this objective, we propose to deal with the decoding shift by splitting the index $In(d)$ of the original symbol d, and removing the arithmetic division operation. Specifically, we represent $In(d)$ by the modular sum of two indexes q_1 and q_2, i.e., $In(d) = (q_1 + q_2)$ mod m. We use one antenna to transmit a symbol of index $(q_1 + In(k))$ mod m, and the other to transmit a symbol of index $(q_2 - In(k))$ mod m. Since the division operation is removed, the index of a received symbol is indeed $((q_1 + In(k))$ mod m + $(q_2 - In(k))$ mod $m)$.

Let q_{1_x} and q_{2_x} denote the x coordinates of q_1 and q_2, respectively. For thefi rst and second scenarios in Table 1, $((q_{1_x} + In(k)_x)$ mod $m + (q_{2_x} - In(k)_x)$ mod $m)$ mod $m = q_{1_x} + q_{2_x} = In(d)_x$. For the third and fourth scenarios, $((q_{1_x} + In(k)_x)$ mod $m + (q_{2_x} - In(k)_x)$ mod $m)$ mod $m = (q_{1_x} + q_{2_x} + pm)$ mod $m = In(d)_x$, where p is an non-zero integer. Thus the decoding shift on x coordinate isfi xed.

5.6 Baseband Processing

The wireless physical layer needs to further process symbols to generate baseband signals suitable for wireless transmission. Specifically, a quadrature modulator is applied. For a symbol (x, y), a transmitter generates two baseband signals by multiplying x and y with two different carrier signals, which are sine or cosine signals and have a phase difference of $\frac{\pi}{2}$.

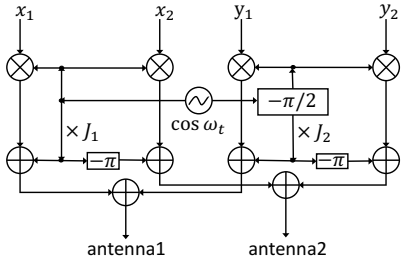

Figure 4: Baseband processing.

To further conceal original message symbols, we propose to add a carrier signal of random amplitude to each baseband signal, such

that we introduce additional randomness to message symbols. As an example shown in Figure 4, symbols to be sent by two antennas are (x_1, y_1) and (x_2, y_2) respectively. x_1 and y_1 are multiplied with the carrier signals $\cos \omega t$ and $\cos(\omega t - \frac{\pi}{2})$ to generate baseband signals $x_1 \cos \omega t$ and $y_1 \cos(\omega t - \frac{\pi}{2})$, respectively. Random numbers J_1 and J_2 are multiplied with the carrier signals to generate random carrier signals $J_1 \cos \omega t$ and $J_2 \cos(\omega t - \frac{\pi}{2})$, respectively. The transmitter then adds baseband signals and random carrier signals together, passes the resulting signal to the dynamic channel cipher, and transmits the outcome using one antenna. Similarly, the transmitter can use the same method to process (x_2, y_2) and generate the signal to be sent by the other antenna. To eliminate random carrier signals at a receiver, we use carrier signals of opposite phases, i.e., for (x_2, y_2), J_1 and J_2 are multiplied with the carrier signals $\cos(\omega t - \pi)$ and $\cos(\omega t - \frac{\pi}{2} - \pi)$, respectively.

6 SECURITY ANALYSIS FOR A SINGLE EAVESDROPPER

Information security can be normally measured by the comparison between two probabilities - the prior probability $P(D)$ of the plaintext D, and the posterior probability $P(D|S)$ of D given the corresponding ciphertext S [38]. Mathematically, we describe information security in the form of entropy, which denotes the average uncertainty of a random variable. For those who do not know the plaintext D, D can be treated as a random variable and its entropy is defined as $H(D) = -\sum_{d \in D} P_D d \log P_D(d)$, where $P_D d$ is the probability when $D = d$. Similarly, the entropy of D conditioned on the ciphertext S is defined as $H(D|S) = \sum_{s \in S} P_S(s) H(D|S = s)$, where $P_S(s)$ is the probability when $S = s$.

We can quantize information security as a form of mutual information, which is defined as the relative entropy between $H(D)$ and $H(D|S)$. In particular, mutual information of D and S is described as $I(D;S) = H(D) - H(D|S)$ [39]. For perfect security, mutual information $I(D;S)$ equals zero. This means that an eavesdropper gains no information of plaintext D from the known ciphertext S.

6.1 Mutual Information at the Eavesdropper

Without loss of generality, we assume the transmitter has two antennas. As mentioned earlier, the index of transmit symbol from one antenna is $(d \pm In(key)) \bmod m$. According to Lemma 5.1, the transmit symbols can be represented by $(d \pm In(key)) \bmod m$ - (δ, δ), which can be written as the sum of two symbols d, k and a constant c (i.e., $s = d + k + c$, where d and k represent the message and key components respectively, and c is a constant represents the remain (δ, δ) and the impact of m). Since c is a deterministic constant that does not contribute to the confidentiality of the message d, we would like to omit it to facilitate the following analysis.

The symbol s is further processed by the baseband and dynamic channel cipher. The ultimate signal to be transmitted by the antenna is $f(t)(d_x \pm k_x) \cos \omega t \pm J_1 \cos \omega t$, where d_x, and k_x are the x coordinate of d and k respectively, and $f(t)$ is the cipher characteristic function. Note that we only consider the transmit symbol of x coordinate, since x and y coordinates are identical and can be processed independently. This expression can be further written as $f(t)((d_x \pm (k_x + J_1)) \cos \omega t$. In following discussion, without loss of generality, we omit $f(t)$, and $cos\omega$ to facilitate presentation, because $f(t)$ is deterministic and generated based on the channel between the transmitter and the receiver, and $cos\omega$ are publicly known carrier signals, and $f(t)$ and carrier signals can be treated as constants. We can thus represent the signal by a simplified form of $d \pm (k + J_1)$. Since k and J_1 are randomly generated, we treat them as an entity and denote $k + J_1$ by k_c.

We represent the signals to be transmitted by both antennas as $d + k_c$ and $d - k_c$, respectively. As mentioned earlier, wireless channel is normally static during a short time (i.e., channel coherence time). Thus, we denote channels between the transmitter and eavesdropper during the transmission time as h_3 and h_4. The eavesdropper receives $r_e = (h_3 + h_4)d + (h_3 - h_4)k_c$. Ideally, for any given r_e, an eavesdropper should obtain no useful information to infer d. We normalize the received content r_e as $r_e = d + r_h k_c$, where $r_h = \frac{h_3 - h_4}{h_3 + h_4}$. Lemma 6.1 gives the mutual information between d and r_e at an eavesdropper.

LEMMA 6.1. *The mutual information $I(D; R_e, R_h)$ between d, r_e, and r_h is less than $\frac{2 \log m}{\pi} \arctan(\frac{d_{range}}{2k_{bnd}}) - \frac{d_{range}}{2k_{bnd}} \ln(\frac{d_{range}^2}{4k_{bnd}^2 + d_{range}^2})$, where $d_{range} = d_{max} - d_{min}$. For any $\epsilon > 0$, we can alwaysfi nd a proper value of k_{bnd} such that $I(D; R_e, R_h) < \epsilon$.*

Proof: r_e, d, r_h, and k_c can be viewed as four random variables R_e, D, R_h and K_c. The CDF $F_{R_e}(r_e|R_h = r_h)$ of R_e is given by $\mathbb{P}(D + R_h K_c \le r_e | R_h = r_h) = \sum_{d \in D} F_{K_c}(\frac{r_e - d}{|r_h|}) P_D(d)$, and the corresponding PDF $f_{R_e}(r_e|R_h = r_h)$ is $\frac{1}{|r_h|} \sum_{d \in D} f_{K_c}(\frac{r_e - d}{|r_h|}) P_D(d)$, which is sectionally uniformly distributed within $(-|r_h|k_{bnd} + d_{min}, |r_h|k_{bnd} + d_{max})$, i.e., it is uniform distributed with any possible value of D over the range $K_u = (-|r_h|k_{bnd} + d_{max}, |r_h|k_{bnd} + d_{min})$. $|r_h|k_{bnd} + d_{min}$ should be larger than $-|r_h|k_{bnd} + d_{max}$, and thus $|r_h|$ is larger than $\frac{d_{range}}{2k_{bnd}}$.

When r_e falls within K_u, the probability that D is uniformly distributed is $P_D(D|R_e = r_e, R_h = r_h) = \frac{1}{m}$. The corresponding entropy is $H(D|R_e = r_e, R_h = r_h) = \log m$, and we can approximate the upper bound of the entropy $H(D|R_e, R_h = r_h)$ as $\log m \frac{2|r_h|k_{bnd} - d_{max} + d_{min}}{2|r_h|k_{bnd}}$.

With $H(D|R_e, R_h = r_h)$ we further derive the entropy $H(D|R_e, R_h)$ conditioned on R_e and R_h. Towards this end, we treat h_3 and h_4 as two random variables H_3 and H_4, and thus $R_h = \frac{H_3 - H_4}{H_3 + H_4}$. By dividing H_3 on both numerator and denominator, R_h can be rewritten as $\frac{1 - (H_4/H_3)}{1 + (H_4/H_3)}$. H_3 and H_4 are two independent zero mean Gaussian random variables with the same variance σ^2. The pdf of $H_c = H_4/H_3$ is $\frac{1}{\pi} \frac{1}{h^2 + 1}$. Since $R_h = \frac{1 - H_c}{1 + H_c}$, the corresponding PDF of R_h is $f_{R_h}(r_h) = \frac{d}{dr_h} F_{R_h}(r_h) = \frac{1}{\pi} \frac{1}{r_h^2 + 1}$.

Because $|r_h|$ is larger than or equal to $\frac{d_{range}}{2k_{bnd}}$, we can obtain $H(D|R_e, R_h) \ge \int_{|r_h| > \frac{d_{range}}{2k_{bnd}}} f_{R_h}(r_h) H(D|R_e, R_h = r_h) dr_h = \log m - \frac{2 \log m}{\pi} \arctan(\frac{d_{range}}{2k_{bnd}}) + \frac{d_{range}}{2k_{bnd}} \ln(\frac{d_{range}^2}{4k_{bnd}^2 + d_{range}^2})$.

Based on the lower bound of $H(D|R_e, R_h)$, the lower bound of mutual information $I(D; R_e, R_h)$ is computed by $\frac{2 \log m}{\pi} \arctan(\frac{d_{range}}{2k_{bnd}}) - \frac{d_{range}}{2k_{bnd}} \ln(\frac{d_{range}^2}{4k_{bnd}^2 + d_{range}^2})$. □

Conclusion: Lemma 6.1 indicates that a single eavesdropper can hardly infer any message from the received signals even with the knowledge of original channel information.

7 SECURITY ANALYSIS FOR COOPERATIVE EAVESDROPPERS

In this section, we investigate the security of the proposed scheme in the presence of multiple eavesdroppers. Assume that Alice is equipped with N antennas and there are N_e eavesdroppers. Let h_{ij} denote the channel between the i^{th} antenna of Alice and the j^{th} eavesdropper. We assume that the eavesdroppers know h_{ij}, but they cannot know the channels between Alice and Bob in a passive way due to the channel spatial uncorrelation property [40]. (In later section, we describe an active attack that can be launched by eavesdroppers to learn the channels between Alice and Bob, and then we propose the countermeasures). The signals received by the eavesdroppers can be modeled by

$$\begin{bmatrix} r_{e1} \\ r_{e2} \\ . \\ r_{eN_e} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{21} & . & h_{N1} \\ h_{12} & h_{22} & . & h_{N2} \\ . & . & . & . \\ h_{1N_e} & h_{2N_e} & . & h_{NN_e} \end{bmatrix}, \begin{bmatrix} s_1 \\ s_2 \\ . \\ s_N \end{bmatrix},$$

where s_i denotes the symbol sent by the i^{th} antenna, and r_{ej} denotes the symbol received by the j^{th} eavesdropper. To facilitate the analysis, we rewrite the model into a compact form of $\mathbf{R_e} = \mathbf{H_e S}$. Because $\mathbf{H_e}$ is known to the eavesdroppers, they can derive the transmit signals by computing $\mathbf{S} = \{\mathbf{H_e H_e}^H\}^{-1}\mathbf{H_e}^H\mathbf{R_e}$, where H denotes the complex conjugate transpose operator. The number of eavesdroppers must be larger than the number of Alice's antennas (i.e. $N_e \geq N$) to ensure that $\mathbf{H_e H_e}^{\mathbf{H}}$ is invertible to compute $\mathbf{S}$. Nevertheless, as mentioned earlier, each symbol is indeed encrypted and expressed as $d_i + k_{c_i}$. Without the knowledge of k_{c_i}, the eavesdroppers still cannotfi gure out the original information.

7.1 Known Plaintext Attack

A message may include predefined information like training sequences, preambles and synchronization codes, which are known to the public. By utilizing predefined information, eavesdroppers may apply existing Least Mean Squares algorithms to derive the channels between Alice and Bob [41]. This attack is referred to as a known plaintext attack [42]. For the proposed scheme, with the knowledge of the channels between the communicators, it is possible for the eavesdroppers tofi gure out the dynamic channel cipher of each antenna and k_{c_i}, andfi nally decode d_i. We refer to the symbols that correspond to the portion of predefined information in a message as the *predefined symbols*. To successfully launch the known plaintext attack, the eavesdroppers must satisfy the following two conditions, 1) The number of eavesdroppers must be larger than the number of Alice's antennas; 2) The number of predefined symbols must be larger than the number of Alice's antennas.

Intuitively, we may increase the number of Alice's antennas or reduce the size of the predefined information to defend against this attack. However, increasing the number of antennas may complicate the wireless system implementation and increase the implementation cost, and reducing the size of the predefined information may lower the message decoding accuracy. In what follows, we present an alternative scheme to defend against known plaintext attack. Unlike the intuitive methods, the proposed scheme does not rely on change of hardware or communication protocols.

7.2 Channel Poisoning

The fundamental reason for a successful known plaintext attack is that the channels between Alice and Bob remain consistent within the transmission time of a message. Due to this reason, the eavesdroppers can apply the channels estimated from the predefined information to equalize the channels of the entire message to retrieve this message.

Based on this observation, we propose to "poison" the wireless channel during the transmission of predefined information, such that the eavesdroppers obtain poisoned channels that are quite different from the original channels. Figure 5 illustrates a simple example of channel poisoning. A training sequence is transmitted at the training stage, and the message payload is prefixed by a preamble. In wireless communication, a receiver usually utilizes a channel estimation algorithm to estimate the wireless channel from a training sequence, and detect the beginning of a message based on a preamble. For the i-th antenna, Alice poisons the wireless channel estimated by the eavesdroppers by multiplying the predefined symbols, including training sequence symbols $d_{i_t}(t)$ and preamble symbols $d_{i_p}(t)$, with a random signal $c_i(t)$.

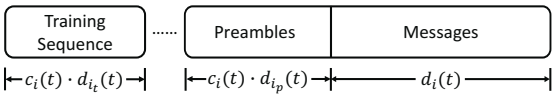

Figure 5: Channel poisoning.

When the transmit signal arrives at the eavesdroppers, as mentioned earlier, they may cooperate together to launch the known plaintext attack to derive the channel from the received predefined symbols. However, because the predefined symbols are multiplied with a random signal $c_i(t)$, the derived channel $h_i(t)'$ between the i^{th} antenna of Alice and Bob is not equal to the actual channel $h_i(t)$ any more. Instead, it becomes a function of both $c_i(t)$ and $h_i(t)$. Specifically, we can easily prove that $h_i(t)' = h_i(t)/c_i(t)$. Since $c_i(t)$ is a random signal, the eavesdroppers cannot get the actual channel $h_i(t)$ from $h_i(t)/c_i(t)$ without the knowledge of $c_i(t)$, and consequently cannot decode the original message $d_i(t)$.

It seems that the legitimate receiver Bob may also estimate poisoned wireless channel and thus fails to decode the original message. Nevertheless, note that we multiply a random signal with the predefined symbols only and the message payload is not changed. Because of the dynamic channel cipher, the key is canceled at the receiver, and the receiver receives altered training sequence symbols $c_i(t)d_{i_t}(t)$ (because the transmitter multiplies $d_{i_t}(t)$ with $c_i(t)$), altered received preamble symbols $c_i(t)d_{i_p}(t)$, and the original message payload symbols. Bob can estimate $c_i(t)$ from the received training sequence and equalize the preamble with $c_i(t)$ for the purpose of the packet synchronization. The message payload is received in the original form, and thus the receiver can recover the original message $d_i(t)$.

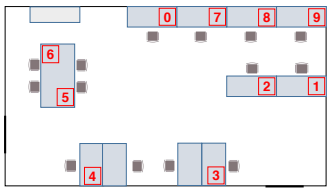

Figure 6: Floor plan.

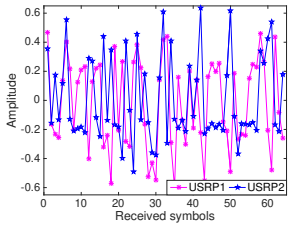

Figure 7: Symbol before cipher.

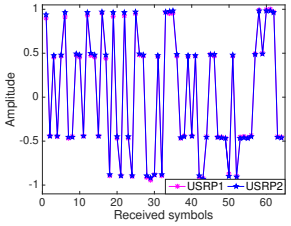

Figure 8: Symbol after cipher.

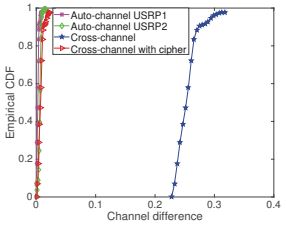

Figure 9: Channel dist. CDF.

8 EXPERIMENT AND EVALUATION

8.1 System Design

We implement a prototype of the proposed DyWCP using the software defined radio toolkit GNURadio and USRPs, which are radio frequency transceivers with high processing capability. The transmitter is built with two synchronized USRPs that are separated by one meter to ensure uncorrelated channels and connected to an external clock source OctoClock-G. The receiver is a standalone USRP.

Our program is built upon the Hardware Driver Library of the GNURadio. The receiver runs the standard receiver program provided by the GNURadio. We redesign the standard transmitter program by adding new modules for modular addition and channel poisoning, as well as new modules to create the dynamic channel cipher and random carrier signals. The prototype system operates at the 2.4 Ghz band and uses 16-QAM modulation. The experiment is conducted in a typical office building with wooden doors, metal and wooden obstacles, and electronic devices. Thefl oor plan is shown in Figure 6. The transmitter and the receiver are at positions 0 and 1, respectively. Eavesdroppers are at position $2 \sim 9$.

8.2 Effectiveness of Dynamic Channel Cipher

The dynamic channel cipher is the prerequisite to enable the proposed DyWCP encryption scheme. Therefore, we would like tofirst evaluate its effectiveness.

The transmitter is constructed by two synchronized USRPs. The dynamic channel cipher should be able to amortize the channel difference between the USRPs and the receiver. To verify this, we use the transmitter to send the same messages to the receiver. Figures 7 and 8 show the examples of signals received from both USRPs of the transmitter. As shown, without the dynamic channel cipher, received signals from the two USRPs are significantly distorted and quite different from each other. On the other hand, when we apply the dynamic channel cipher to the transmit signals, the received signals from two USRPs become clear and exhibit the same shapes.

Channel estimation between the transmitter and the receiver is needed to launch the dynamic channel cipher. In our experiment, the channel is estimated in a training stage, where the receiver broadcasts a beacon signal to the transmitter, and the transmitter then estimates the channels based on the received beacon signals. We estimate the channel 1000 times and record the estimation results. Let h_{1i} and h_{2i} denote the i-th ($1 \leq i \leq 1000$) estimation results for the channel between thefi rst USRP and the receiver, and that between the second USRP and the receiver, respectively. To quantize the difference between both channels, we compute the cross-channel difference by $|h_{1i} - h_{2j}|$ for $1 \leq \forall i,j \leq 1000$. We also compute the auto-channel difference by $|h_{1i} - h_{1j}|$ and $|h_{2i} - h_{2j}|$ for $1 \leq \forall i,j \leq 1000$.

Figure 9 shows the Cumulative Distribution Function of the cross-channel and auto-channel differences without and with the use of the dynamic channel cipher. When the cipher is not in use, the cross-channel difference is much larger than the auto-channel difference, indicating that the channels of both USRPs are significantly uncorrelated. When the cipher is in use, the channels between both USRPs and the receiver are equalized, and hence the cross-channel difference is close to the auto-channel difference.

8.3 NIST Randomness Testing

We use the widely-used randomness testing tool, NIST Test Suite, to evaluate the randomness of bits received [21]. For comparison, we evaluate the randomness of bit sequences received by both the receiver and an eavesdropper. Without loss of generality, we select USRP at position2 as the eavesdropper.

Wefi rst specify the significance level α, which is the probability that a NIST test indicates that a sequence is not random but it is indeed random. α is usually chosen within $[0.001, 0.01]$ and we set it to 0.01 in our experiment. We examine a total of 1000 binary sequences received by the legitimate receiver and eavesdropper, respectively. The outcome of the test can be interpreted by the metric, pass rate (i.e., the proportion of sequences passing a test).

Table 2: NIST test results of pass rate.

Test name	PR	Test name	PR
Frequency	0.991	Block frequency (m=128)	0.993
Forward cumulative sums	0.990	Reverse cumulative sums	0.990
Runs	0.994	Longest runs of ones	0.996
Rank	0.993	Discrete fourier transform	0.992
Non-overlapping templates	0.990	Overlapping templates (m=9)	0.991
Universal	0.990	Approximate entropy (m=10)	0.991
Random excursions (x=+1)	0.988	Random excursions variant	0.989
Serial (m=16)	0.995	Linear Complexity (M=500)	0.993

Pass Rate (PR): The theoretical rate that a pure random bit sequence passes a NIST test is $1 - \alpha$, which is 0.99 in our experiment. In practice, because we test afi nite number (i.e., 1000) of bit sequences, the observed pass rate for a NIST test may slightly deviate from the theoretical one. NIST defines the *confidence interval*, within which a pass rate is acceptable. The confidence interval is given by $[p - 3\sqrt{\frac{p(1-p)}{N_m}}, p + 3\sqrt{\frac{p(1-p)}{N_m}}]$, where $p = 1 - \alpha$, and N_m is the number of tested sequences [43]. The confidence interval in our experiment is $[0.980561, 0.999439]$. The experiment shows

that none of the received sequences at receiver can pass the NIST tests. This is because the key is canceled and the receiver receives original sequences that are deterministic and non-random. Table 2 shows pass rates of sequences at the eavesdropper for all 16 NIST tests. We can see that the maximum and minimum pass rates are 0.996 and 0.988. Both rates fall within the confidence interval. This means that the eavesdropper receives random bit sequences and gains no useful information.

8.4 Evaluation of BER, SER, and Entropy

In addition to randomness, we also use the following evaluation metrics to assess the effectiveness of the scheme.

- **Bit error rate (BER):** Bit error rate is the ratio of number of bit errors to the total number of bits transmitted.
- **Symbol error rate (SER):** Symbol error rate is the ratio of the number of symbol errors to the total number of transmit symbols.
- **Entropy:** Entropy indicates the average uncertainty of a random variable. In our experiment, entropy is referred to as the conditional entropy $H(D|R)$ of a transmit symbol D given a received symbol R.

In our experiment, we would like to examine BER, SER, and entropy experienced by the receiver and the eavesdroppers. To understand the impact of random carrier signals, we set the power ratio between a random carrier signal and a baseband signal to 0, 10, and 100, respectively. A power ratio of 0 indicates that no random carrier signals are generated and added to the baseband signals. Tables 3 and 4 show the evaluation results regarding the three metrics for the two power ratios, respectively.

Table 3: Without random carrier signals.

Pos.	BER	SER	Entropy
1	0.0001	0.0002	0.0003
2	0.4967	0.9559	3.5594
3	0.5217	0.9367	3.7241
4	0.4819	0.8309	3.3152
5	0.5046	0.8484	3.7374
6	0.4785	0.8448	3.4938
7	0.4960	0.8469	3.7697
8	0.4939	0.7893	3.6228
9	0.5045	0.9662	3.6869

Table 4: The power ratio is 10.

Pos.	BER	SER	Entropy
1	0.0001	0.0003	0.0004
2	0.5012	0.9501	3.9065
3	0.4987	0.8775	3.9198
4	0.4703	0.9392	3.8153
5	0.4996	0.8891	3.9429
6	0.4998	0.9363	3.9373
7	0.5001	0.9408	3.9366
8	0.4993	0.9074	3.9694
9	0.5000	0.9352	3.9057

As shown in Table 3, when the power ratio is 0 (random carrier signals are not in use), the receiver at position 1 has low BER and SER that are caused by the natural channel noise between two USRPs. On the other hand, eavesdroppers at positions $1 \sim 9$ all experience significant signal distortions. Specifically, BERfluctuates around 0.5, and SER ranges between 0.7893 and 0.9662. Since a binary bit is either 1 or 0, bit error rate of 0.5 indicates that received bits are totally uncorrelated with transmit bits. We also see that the entropy of the receiver is as low as 0.0003. This means that received symbols are almost deterministic without randomness and uncertainties. Eavesdroppers experience a much higher entropy ranging from 3.3152 to 3.7697, which means received signals are random and hardly to be decoded as original messages.

Table 4 shows the evaluation results when random carrier signals are used and the power ratio is set to 10. We observe a trivial but slight increase on BER, SER, and entropy. For example, the eavesdropper at position 5 experiences SER of 0.8891 and entropy of 3.9429, which are larger than the previous SER of 0.8484 and entropy of 3.7374. This is because the random carrier signals introduce additional randomness and further complicate the decoding process at the eavesdropper.

8.5 Channel Poisoning against Cooperative Eavesdroppers

As discussed earlier, corporative eavesdroppers may launch the known plaintext attack to derive channels between the transmitter and the receiver, and then decode original messages. To defend against such attacks, we poison the channels during the transmission of predefined symbols. In the experiment, each packet carries a 1000-bit payload and is prefixed with a 64-bit access code, which is predefined and used for packet synchronization. We poison the modulated symbols of access code by multiplying it with a randomly generated sequence of the same length. We group eavesdroppers into pairs, and each pair of eavesdroppers collaborate to derive the channels, and try to decode the original messages.

Table 5 shows BER, SER, and entropy of the receiver and the eavesdroppers. We can see that BER and SER at the receiver are as low as 0.0002 and 0.0003 respectively. This result indicates that channel poisoning does not affect the message decoding at the receiver. Meanwhile eavesdroppers experience BER around 0.5 and much higher SER, which prevents correct message decoding. For example, the pair of collaborative eavesdroppers at locations 8, 9 encounter BER of 0.5048 and SER as high as 0.9450, indicating that about half of received bits arefl ipped and 94.5% symbols are received in wrong form. The entropy is 3.9670, which is 9,917 times of that of the receiver.

Table 5: Impact of channel poisoning.

Pos.	BER	SER	Entropy
1	0.0002	0.0003	0.0004
2, 3	0.5034	0.8840	3.9192
4, 5	0.5003	0.9412	3.8884
6, 7	0.4998	0.8957	3.9366
8, 9	0.5048	0.9450	3.9670

9 CONCLUSION

In the paper, we propose a lightweight encryption scheme named DyWCP to protect sensitive data in IoT devices. Towards the proposed scheme, we create a dynamic channel cipher to utilize wireless channel features for basic encryption, a symbol-level modular addition method to enables the incorporation of the proposed scheme into practical wireless systems, and a channel poisoning method to address the known plaintext attack for improved security. We implement a prototype of the proposed scheme on top of software defined radio platforms, and evaluate the security performance of the proposed scheme in the presence of a single and multiple eavesdroppers.

10 ACKNOWLEDGEMENT

The work at the New Mexico State University was supported in part by NSF under grant ECCS-2139028.

REFERENCES

[1] Internet of things. https://en.wikipedia.org/wiki/Internet_of_things.

[2] Youyang Qu, Shui Yu, Wanlei Zhou, Sancheng Peng, Guojun Wang, and Ke Xiao. Privacy of things: Emerging challenges and opportunities in wireless internet of things. *IEEE Wireless Communications*, 25(6):91–97, 2018.

[3] Minhaj Ahmad Khan and Khaled Salah. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018.

[4] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[5] Z Berkay Celik, Leonardo Babun, Amit Kumar Sikder, Hidayet Aksu, Gang Tan, Patrick McDaniel, and A Selcuk Uluagac. Sensitive information tracking in commodity iot. In *Proceedings of USENIX Security*, 2018.

[6] Sriram Lakshmanan, Cheng-Lin Tsao, Raghupathy Sivakumar, and Karthikeyan Sundaresan. Securing wireless data networks against eavesdropping using smart antennas. In *Proceedings of IEEE ICDCS*, 2008.

[7] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight block cipher for multiple platforms. In *Proceedings of Springer SAC*, 2012.

[8] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In *Proceedings of Springer CHES*, 2011.

[9] Christophe De Canniere, Orr Dunkelman, and Miroslav Knežević. Katan and ktantan—a family of small and efficient hardware-oriented block ciphers. In *Proceedings of Springer CHES*, 2009.

[10] Stefan Tillich and Johann Großschädl. Instruction set extensions for efficient aes implementation on 32-bit processors. In *Proceedings of Springer CHES*, 2006.

[11] Thomas Wollinger, Jorge Guajardo, and Christof Paar. Security on fpgas: State-of-the-art implementations and attacks. *ACM Transactions on Embedded Computing Systems*, 3(3):534–574, 2004.

[12] Soren Rinne, Thomas Eisenbarth, and Christof Paar. Performance analysis of contemporary light-weight block ciphers on 8-bit microcontrollers. *ECRYPT*, page 33.

[13] George Hatzivasilis, Konstantinos Fysarakis, Ioannis Papaefstathiou, and Charalampos Manifavas. A review of lightweight block ciphers. *Journal of Cryptographic Engineering*, 8(2):141–184, 2018.

[14] Bassam J Mohd, Thaier Hayajneh, and Athanasios V Vasilakos. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *Journal of Network and Computer Applications*, 58:73–93, 2015.

[15] Claude E Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[16] The one-time pad. http://users.telenet.be/d.rijmenants/en/onetimepad.htm.

[17] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[18] Junxing Zhang, Mohammad H Firooz, Neal Patwari, and Sneha K Kasera. Advancing wireless link signatures for location distinction. In *Proceedings of ACM MobiCom*, 2008.

[19] Matthias Schulz, Adrian Loch, and Matthias Hollick. Practical known-plaintext attacks against physical layer security in wireless mimo systems. In *Proceedings of NDSS*, 2014.

[20] Gnuradio. http://gnuradio.org/redmine/projects/gnuradio/wiki.

[21] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, DTIC Document, 2001.

[22] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *Proceedings of Springer CHES*, 2007.

[23] Satashu Goel and Rohit Negi. Secret communication in presence of colluding eavesdroppers. In *Proceedings of IEEE MILCOM*, 2005.

[24] Zang Li, Roy Yates, and Wade Trappe. Achieving secret communication for fast rayleigh fading channels. *IEEE Transactions on Wireless Communications*, 9(9):2792–2799, 2010.

[25] Yunchuan Yang, Wenbo Wang, Hui Zhao, and Long Zhao. Transmitter beamforming and artificial noise with delayed feedback: Secrecy rate and power allocation. *Journal of Communications and Networks*, 14(4):374–384, 2012.

[26] Narendra Anand, Sung-Ju Lee, and Edward W Knightly. Strobe: Actively securing wireless communications using zero-forcing beamforming. In *Proceedings of IEEE INFOCOM*, 2012.

[27] Dirk Rijmenants. The complete guide to secure communications with the one time pad cipher. *Cipher Machines and Cryptology*, 2010.

[28] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[29] Saurabh Ganeriwal, Christina Pöpper, SrdjanČ apkun, and Mani B Srivastava. Secure time synchronization in sensor networks. *ACM Transactions on Information and System Security*, 11(4):23, 2008.

[30] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[31] Edward A Lee and David G Messerschmitt. *Digital communication*. Springer Science + Business Media, 2012.

[32] Random number generation. https://en.wikipedia.org/wiki/Random_number_generation.

[33] Lin Xiao, Stephen Boyd, and Sanjay Lall. A space-time diffusion scheme for peer-to-peer least-squares estimation. In *Proceedings of ACM IPSN*, 2006.

[34] Zhangjie Peng, Wei Xu, Jun Zhu, Hua Zhang, and Chunming Zhao. On performance and feedback strategy of secure multiuser communications with mmse channel estimate. *IEEE Transactions on Wireless Communications*, 15(2):1602–1616, 2016.

[35] Yu-Chih Tung, Sihui Han, Dongyao Chen, and Kang G Shin. Vulnerability and protection of channel state information in multiuser mimo networks. In *Proceedings of ACM CCS*, 2014.

[36] Eldad Perahia and Robert Stacey. *Next Generation Wireless LANS: 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[37] Closed set. https://en.wikipedia.org/wiki/Closed_set.

[38] Christian Cachin. *Entropy measures and unconditional security in cryptography*. PhD thesis, ETH Zurich, 1997.

[39] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.

[40] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Proceedings of ACM MobiCom*, 2008.

[41] Markku Pukkila. Channel estimation modeling. *Nokia Research Center*, 2000.

[42] Charlie Kaufman, Radia Perlman, and Mike Speciner. *Network security: private communication in a public world*. Prentice Hall Press, 2002.

[43] Vinod Patidar, Krishan K Sud, and Narendra K Pareek. A pseudo random bit generator based on chaotic logistic map and its statistical testing. *Informatica (slovenia)*, 33(4):441–452, 2009.