Wireless-Assisted Key Establishment Leveraging Channel Manipulation

Song Fang, Ian Markwood, Yao Liu

Abstract—Wireless communication is easily eavesdropped due to the broadcast nature of the wireless medium. This has spurred extensive research into secret key establishment using physical layer characteristics of wireless channels. In all these schemes, the secret keys directly originate from the physical features of the real wireless channel, which is highly dependent on the communication environment nearby. Also, previous schemes require performing information reconciliation, which increases both the costs and the risk of key leakage. In this paper, we exhibit a novel wireless key establishment method allowing the transmitter to specify arbitrary content as the key and cause the receiver to obtain the same key leveraging a channel manipulation technique. We furthermore enable the transmitter to apply error-correction code to the key, so that the receiver can automatically correct any mismatched bits without sending key-related information back to the transmitter over the public channel. Experimental results demonstrate that our key establishment method reaches a success rate as high as 91.0% for establishing a 168-bit key between the transmitter and the receiver, and meanwhile the chance that the eavesdropper can infer the key in meter-order range of the receiver is subdued into the range of 0$\sim$0.10%.

Index Terms—Secret key, eavesdropping, error-correction code, key establishment chain.

1 Introduction

Wireless key establishment has been widely studied in the past years (e.g., [1]–[13]) for its easy implementation, low computational requirement, and small energy consumption. The common intuition is to establish a shared key utilizing the fact that the transmitter and receiver of one wireless link can observe the same channel simultaneously, a property known as *wireless channel reciprocity*. Next, the *spatial uncorrelation property* of the wireless channel provides the security basis for these wireless key establishments, i.e., a receiver will observe differing channels between transmitters in different locations. Hence, an eavesdropper able to receive the signal sent by transmitters Alice or Bob will be unable to decrypt it, as the extracted channel characteristic will differ from that visible to Alice and Bob, so long as the eavesdropper is not co-located with either character [14].

Existing wireless key establishment techniques normally entail three steps to share a secret key between Alice and Bob, namely *quantization*, *reconciliation*, and *privacy amplification*. Quantization involves both parties sampling the channel characteristic and then quantizing the sampled data into initial binary bit sequences. Unfortunately, channel noise may cause the quantization step to render some moderately different bit sequences for Alice compared to Bob. Reconciliation schemes aim to correct these mismatched bits through information exchanges. Finally, each communicator performs privacy amplification to confuse a malicious listener from deducing the secret bit sequence.

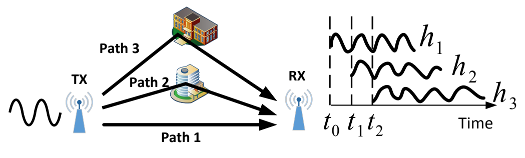
This workflow naturally imparts two major drawbacks. First, the shared key originates from the actual channel, and the communicators cannot control such a key. The key depends on the wireless channel dynamics, and a static environment results in an established key of low entropy [3]. Second, Alice and Bob must exchange multiple messages over the public channel to agree on an identical key during reconciliation. These messages contain sensitive key-related information and not only create an opportunity for an attacker to capture the exchanged information and infer the key [2], but also highly decrease the efficiency of key establishment. In this paper, we evolve fundamental aspects of existing key establishment techniques to eliminate these aforementioned deficiencies, by removing the mutual message exchange process heretofore universally required. Specifically, we enable Alice to deliver a key to Bob by sending a one-time message, once Alice receives a key establishment inquiry from Bob. Bob is then passive and no longer needs to make any unprotected communications.

Intuitively, this sort of key establishment scheme can be achieved if the transmitter can manipulate the channel characteristics observed by the receiver. This transmitter will select any random content as the key and generate channel characteristics equal to the key for the receiver to observe. Because the transmitter can specify any content as the key, the transmitter can further encode the key with error-correction code, and accordingly the receiver will be able to automatically correct any mismatched bits without sending key-related information to the transmitter over the public channel.

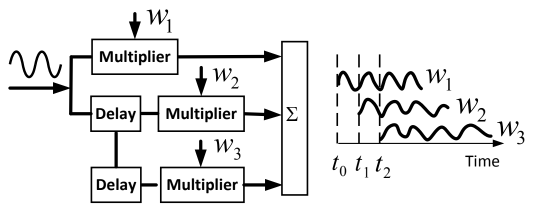
The spatial uncorrelation property is caused by the multipath propagation phenomenon, where a signal travels in the air over multiple paths due to signal reflections, diffractions, and scatterings. Geographically separated transmitter and receiver pairs encounter different multipaths and necessarily different channel characteristics. Hence, if we can create an "artificial multipath" effect, then we can manipulate the channel char-

• *Song Fang is with the School of Computer Science, University of Oklahoma, Norman, OK, 73019. E-mail: songf@ou.edu.*
Ian Markwood is with Cyber Development, BlackHorse Solutions, Herndon, Virginia, 20171. E-mail: imarkwood@mail.usf.edu.
Yao Liu is with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL, 33620. E-mail: yliu@cse.usf.edu.

An earlier version of the work was published in IEEE CNS'17.



(a) Real multipath propagation



(b) Artificial multipath propagation

Fig. 1. Real v.s. artificial multipath effect.

acteristics observed by the receiver, and specify "artificial" channel characteristics equal to the encoded key at the receiver.

Figure 1(a) shows a simple multipath propagation example involving three paths. The transmitter sends a wireless signal. The receiver observes the superposition of three signal copies, each of which is distorted by the corresponding path. We use h_i to denote the distortion introduced by Path i. In Figure 1(a), the vector $[h_1, h_2, h_3]$ represents the channel characteristics of the 3-path channel and this vector is referred as the *channel impulse response* [15]. The shared key is normally quantized from the channel impulse response [2], [8], [10]. The signal copy distorted by Path i can be usually modeled by $h_i x$, where x is the original, undistorted signal [15]. It is easily understood then that multipath propagation in the real world can be simulated using simple delay and multiplication operations. Figure 1(b) illustrates how this may be achieved. Here, the transmitter sends the original signal as well as its time-delayed copies of this signal to mimic the different arrival times of each multipath component. The original signal and all copies are multiplied against coefficients w_i to mimic the distortion caused by each path i. Consequently, the receiver observes an aggregation of one signal plus time-delayed copies, each undergoing a certain path distortion, and thus obtains a channel impulse response vector of $[w_1, w_2, w_3]$ corresponding to the expected multipath effect but created entirely by the transmitter. The delay (e.g., $t_1 - t_0$) is the arrival time difference of two consecutively received signals. The challenges for building the proposed scheme are summarized below.

First, to specify the chosen channel impulse response at the receiver, the transmitter must cancel the real multipath effect without compromising the spatial uncorrelation property, which serves as the security foundation for all wireless key establishment techniques. To address this, we create a customized channel manipulation technique.

Second, upon receiving the signal, the receiver estimates and quantizes the channel impulse response to obtain the key. Hence, the transmitter must represent the key in the form of channel impulse responses, and find an appropriate key mapping method to enable the receiver to achieve a low error rate after the quantization. Accordingly, we design a key mapping technique to translate a key into manipulated channel impulse responses of a multipath channel.

Third, an important question is whether an eavesdropper can decode the key with the help of the same error correction code. When an eavesdropper's channel is uncorrelated with the receiver's channel, the key observed by the eavesdropper exhibits more deviation from the actual key than the receiver, allowing the ability of the communicators to choose a code type, which can resolve a number of bit errors larger than the receiver typically encounters but smaller than the number by the eavesdropper. For an eavesdropper located 4 meters away from the receiver, a 168-bit key can be established between two parties with a success rate as high as 91.0% at the receiver, while the probability that the eavesdropper can break the key is subdued into the range of 0$\sim$0.1%.

We lastly consider the scenario where Bob changes locations. We do not assume a physical location will be protected after a legitimate user leaves (i.e., an eavesdropper may approach the location). This means when Bob moves to a new location, he meets a dilemma of whether to reuse the previously generated key. If he does, he and Alice can continue their previous unfinished communication, as Alice still associates him with the key, but the attacker can discover that key by occupying the old location. On the other hand, if he establishes another key with Alice for the sake of security, their ongoing communication will be disrupted, as Alice can no longer recognize Bob and has to regard him as another entity with whom to initiate secure communication. We propose a key establishment chain technique to solve the dilemma and achieve both confidentiality and continuity of communications.

2 Preliminaries

Channel Estimation. Channel impulse response quantifies the effect of the multipath environment in wireless communications. Each path imposes a time delay, magnitude attenuation, and phase shift on the signal traveling along it. Channel is usually estimated based on training bit sequences and received signal samples. Physical layer channel estimation can be processed in either frequency or time domains which are inter-convertible due to their linear relationship [15].

The received signal $r(t)$ can be denoted as the convolution of the transmitted signal $s(t)$ and the channel impulse response $h(t)$ (we omit the noise term for the sake of simplicity): $r(t) = s(t) * h(t)$. In the frequency domain, we have $R(f,t) = S(f,t)H(f,t)$, where $R(f,t)$, $S(f,t)$ and $H(f,t)$ are the Fourier transforms of $r(t)$, $s(t)$ and $h(t)$, respectively. Thus, with knowledge of the transmitted and received signals, we easily obtain $H(f,t)$ and perform the inverse Fourier transform operation on it to find the corresponding channel impulse response $h(t)$, denoted as $h(t) = \mathcal{F}^{-1}\{\frac{R(f,t)}{S(f,t)}\}$, where $\mathcal{F}^{-1}\{\cdot\}$ indicates the inverse Fourier transform. We sample the received signal with a symbol period of T_s and obtain the following sampled impulse response vector with a length of L: $h = [h_1, \cdots, h_L]$, where $h_i = h((i-1)T_s)$, $i \in \{1, \cdots, L\}$.

Error Correction Code (ECC). ECC is developed to correct errors in data transmission and commonly takes the form of block or convolutional ECC [15]. Reed-Solomon (RS) ECC [16] is a typical block ECC with very strong error-correction capacity, especially against the burst errors inherent to wireless communications. Without loss of generality, we choose RS ECC to encode and reconstruct our keys.

3 Assumptions and attack model

In a general scenario, Alice wants to establish a secret key with Bob. We assume that Alice and Bob reside within each other's communication range and have the ability to do channel estimation. Also, we assume the training sequence for channel estimation is public, which conforms with the design of many commercial wireless communication systems [17]. We further assume that the channel impulse response is stable in a short period of time (e.g., a packet duration), which is a common assumption for designing wireless communications [18]–[20].

Besides, before launching channel manipulation based key establishment, we assume Alice knows the actual channel impulse response between herself and Bob. This can be achieved by estimating the channel impulse response from the wireless signals (e.g., key establishment inquiries) emitted by Bob. Note that in many cases, wireless communications are built upon communication protocols like 802.11 and TCP/IP and thus they are two-way. Alice and Bob can hear each other. For one-way communications (Alice $\rightarrow$ Bob), Alice cannot hear the wireless signals from Bob and thus she is unable to launch channel manipulation. In this case, all previous wireless key establishments would fail as the generated key at Alice depends on the received signal sent by Bob. Finally, the RS code enforced by Alice is assumed publicly available so that Bob can do the corresponding decryption.

In our attack model, we consider that an attacker Eve aims to derive the secret key established between Alice and Bob, and presume that Eve has the ability to 1) do channel estimation; 2) know the secret key quantization algorithm and the RS code that the transmitter utilizes.

4 Key Establishment Scheme

4.1 Scheme Outline

The transmitter first chooses a key, encodes it with RS ECC. The selection of RS code is based on the length of the chosen key and other environmental factors which are studied in Section 5.5. After that, the transmitter maps the encoded key into an artificial channel impulse response, denoted as $h_m(t)$. To launch the channel manipulation, the transmitter needs then to calculate the aggregate signal $s_m(t)$ to send. Correspondingly, the signal $r_m(t)$ received from the transmitter can be represented by $r_m(t) = s_m(t) * h(t)$, where $h(t)$ is the actual channel impulse response between the transmitter and the receiver. The receiver uses $r_m(t)$ and the public probe signal $s(t)$ to do channel estimation. With $h_{est}(t)$ denoting the estimated channel impulse response at the receiver, we have $s(t) * h_{est}(t) = s_m(t) * h(t)$. Hence, the transmitter should construct the aggregated signal $s_m(t)$ to make $h_{est}(t) = h_m(t)$. We give the detailed calculation in Section 4.3.

With $h_{est}(t)$, the receiver first obtains a bit sequence via quantization, and then feeds it into a RS decoder. As long as the number of symbol errors in the quantized bit sequence does not exceed the error correction capability of the chosen RS code, the receiver can successfully recover the secret key specified by the transmitter. Figure 2 shows the flow chart of the proposed key establishment.

4.2 Key Mapping

Key mapping is the process of encrypting the selected binary key bits with RS code, and converting the encrypted key into an artificial channel impulse response. A channel impulse response is actually composed of a sequence of complex numbers. To simplify the conversion process, the channel impulse response used here refers to its magnitude, which is a vector of decimal numbers. Thus, key mapping can be regarded as a binary-decimal conversion.

The transmitter can arbitrarily specify a sequence of bits as the secret key, and then input it into an RS encoder. An RS(n, k) code has n symbols of s bits each, the first k of which are symbols comprised of selected key bits and any required padding and the rest calculated based on the RS algorithm and the k-symbol input. Given a symbol size s, the maximum length of the encoded message is $m = 2^s - 1$, so $n \leq m$ should hold. Finally, the RS decoder can correct any $(n-k)/2$ symbol errors in the encoded message.

The next step is to convert the encoded message with $n * s$ bits into channel impulse responses, each a length-L vector of decimal numbers, where L denotes the maximum number of manipulated resolvable multipath components that can be observed by the receiver. Generally, the transmitter can use two different conversion strategies, an absolute value based and a relative value based. In the former, a bit sequence is regarded as a binary number and directly translated into a decimal number denoting the value of a path response. In the latter case, a bit is translated into a quantitative relationship of two path responses, and this relationship may be the comparison result of the value size or the ratio. In this paper, we focus on the relative value based method, mapping the bits into the relation (difference) between path response values.

Let $h_m = [h_{m1}, h_{m2}, \cdots, h_{mL}]$ denote the generated channel impulse response. The differences between the first path response value and the others are $\Delta_1 = h_{m2} - h_{m1}$, $\Delta_2 = h_{m3} - h_{m1}$, $\cdots$, $\Delta_{L-1} = h_{mL} - h_{m1}$ respectively. With these $(L-1)$ path response value differences, we can compute another $\lfloor (L-1)/2 \rfloor$ differences, defined by $\Delta_{(L-1)+i} = |\Delta_{2i-1}| - |\Delta_{2i}|, i \in \{1, 2, \cdots, \lfloor (L-1)/2 \rfloor\}$. In fact, further levels could be utilized, but we use two for the scope of this paper. Hence, a bit sequence with length $L_b = L - 1 + \lfloor (L-1)/2) \rfloor$ can generate a channel impulse response with length L using each of these differences to define one bit. In order to reduce the error, the transmitter uses a quantum q as a positive or negative *path differential* to distinguish these differences for constructing an artificial multipath response. Based on the selected key and this q, the manipulated channel is assembled such that

$$\begin{cases} \Delta_i = q_+, & if\ key\ bit\ is\ 1 \\ \Delta_i = q_-, & if\ key\ bit\ is\ 0, \end{cases} \tag{1}$$

where $i \in \{1, 2, \cdots, L_b\}$. In Section 5.4, we show that when the path differential q is well chosen, the bits extracted by the receiver are identical to the bits specified by the transmitter with high probability. In this manner, the encoded key of $n * s$ bits can be mapped into $\lceil (n * s)/L_b \rceil$ channel impulse responses, which are then launched in the channel manipulation step which follows.

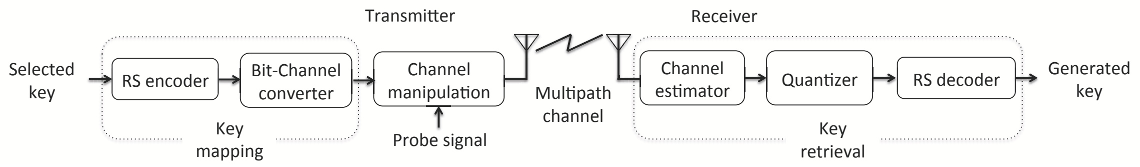


Fig. 2. Flow chart of manipulatable wireless key establishment.

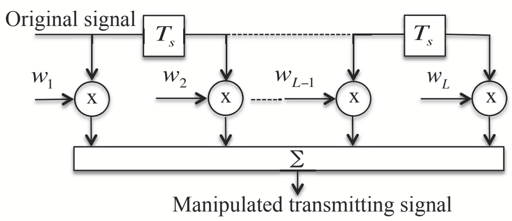


Fig. 3. Channel manipulation filter.

4.3 Channel Manipulation

The goal of channel manipulation is to make the channel impulse response estimated at the receiver equal to the artificial channel impulse response generated through key mapping, i.e., $h_{est} = h_m$. As mentioned earlier, the transmitter intends to emulate the real multipath effect by sending aggregated weighted, delayed copies of the original transmitting signal.

Again, channel manipulation can be regarded as a delay-weight-sum module, implemented by a finite impulse response (FIR) filter as shown in Figure 3. The delay is set to one symbol duration, the transmission time of one symbol as denoted by T_s. The impulse response of the channel manipulation process can be represented by $w(t) = \sum_{i=1}^{L} w_i\delta(t - t_0 - (i-1)T_s)$, where t_0 is the arrival time of the first arrived signal copy. Thus, the manipulated transmission signal is the convolution of the transmitted signal $s(t)$ with the impulse response $w(t)$. This manipulated transmission signal $s(t) * w(t)$ is sent to the receiver through the real multipath channel. Hence, the corresponding signal obtained by the receiver is $r_m(t) = (s(t) * w(t)) * h(t)$. The receiver then utilizes the received signal and the public probe signal $s(t)$ to estimate the channel impulse response, described as $r_m(t) = s(t) * h_{est}(t)$. Based on the associativity of convolution, we obtain $\hat{h}(t) = w(t) * h(t)$. Channel manipulation aims to make $h_{est}(t) = h_m(t)$, where $h_m(t)$ is the value of h_m at a given time. Therefore, we have $h_m(t) = w(t) * h(t)$, and in the frequency we get $H_m(f,t) = W(f,t)H(f,t)$, where $H_m(f,t)$ and $W(f,t)$ are the Fourier transforms of $h_m(t)$ and $w(t)$. Therefore, we can solve the impulse response $w(t)$ of channel manipulation as

$$w(t) \quad = \quad \mathcal{F}^{-1}\{\frac{H_m(f,t)}{H(f,t)}\} = \mathcal{F}^{-1}\{\frac{R_m(f,t)}{R(f,t)}\}, \qquad (2)$$

where, $R_m(f,t)$ is the Fourier transform of $r_m(t)$. Thus, the transmitter can set the parameter $\mathbf{w} = [w_1, \cdots, w_L]$, enabling the receiver to obtain the generated channel impulse response.

4.4 Key Retrieval

With the estimated channel impulse response, the receiver runs quantization to generate bits. This key retrieval process is an inverse process of key mapping, i.e., a decimal-binary conversion. The quantizer we use is defined as

$$Q(\Delta_i) = \begin{cases} 1, & if\ \Delta_i > 0 \\ 0, & otherwise. \end{cases} \qquad (3)$$

Note that since we remove the exchange of reconciliation information, missing bits are not handled in quantization. The quantizer defined above always returns a value, so the receiver will obtain a bit sequence of length L_b after applying quantization to an estimated channel impulse response.

- With obtained channel estimates $h_{est} = [\hat{h}_1, \hat{h}_2, \cdots, \hat{h}_L]$, the receiver first calculates the $L-1$ differences between the first estimated path response and each of the others with the equation $\Delta_i = \hat{h}_{i+1} - \hat{h}_1$, where $i \in \{1, 2, \cdots, L-1\}$. For each calculated difference, the receiver quantizes it using $Q(\cdot)$.
- The receiver next computes the $\lfloor (L-1)/2 \rfloor$ differences between every pair of differences calculated in the previous step, using $\Delta_{L-1+i} = |\hat{\Delta}_{2i-1}| - |\hat{\Delta}_{2i}|$, where $i \in \{1, 2, \cdots, \lfloor (L-1)/2 \rfloor\}$. Again, the receiver quantizes each newly obtained difference with quantizer $Q(\cdot)$. Between these two steps, the total number of differences calculated is $L_b = L - 1 + \lfloor (L-1)/2 \rfloor$.
- Once the length of the obtained bits reaches the required length $n * s$ of the chosen RS(n,k) decoder, the receiver decodes them, and the first k symbols in the decoded message are then extracted as the secret key of $k * s$ bits.

To more clearly demonstrate the key retrieval process, we pick $L = 3$ as an example. At the receiver side, the estimated channel impulse response is denoted as $h_{est} = [\hat{h}_1, \hat{h}_2, \hat{h}_3]$, the corresponding differences are calculated by $\Delta_1 = \hat{h}_2 - \hat{h}_1$, $\Delta_2 = \hat{h}_3 - \hat{h}_1$ and $\Delta_3 = |\Delta_1| - |\Delta_2|$. The receiver then obtains 3 bits by quantizing. This process is repeated, reassembling the encoded key 3 bits at a time, until all (possibly including padding) have arrived. The bits are finally decoded according to the selected RS ECC scheme into the secret key.

4.5 Moving Scenarios

When Bob moves to a new location, Alice and Bob cannot reuse the "old" key they established when Bob was at the old location. An eavesdropper may occupy Bob's old position and infer the old key so long as the channel between that location and Alice remains largely unchanged. Also, Bob needs to prove his identify to Alice if he wishes to re-establish a secret key and continue communications. In the following, we first show insecurity of the old key, and then discuss the key re-establishment between Alice and Bob. Lastly, we generalize the problem of Bob's movements and introduce a new "key establishment chain" technique to solve the challenge.

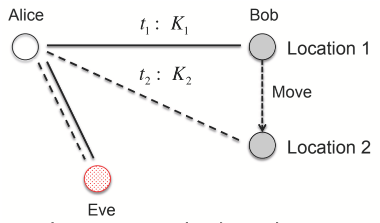


Fig. 4. The receiver moves its location.

4.5.1 Insecurity of the Old Key

As shown in Figure 4, at time t_1, Alice establishes a shared secret key K_1 with Bob at Location 1, and we use $h_{1_{t_1}}$ and $h_{m_{t_1}}$ to denote the actual channel and the manipulated channel between Alice and Location 1, respectively. At time t_2, Bob moves to Location 2, and we use $h_{1_{t_2}}$ to represent the actual channel between Alice and Location 1.

It is often assumed that an eavesdropper is not in the same physical location as a legitimate user [1], [21], as the exposure risk for the eavesdropper is extremely high. However, the eavesdropper may stalk a legitimate user [22], namely, follow Alice or Bob without sharing a location with either when they are there. Thus, when Bob moves to Location 2 at t_2, we assume that eavesdropper Eve (or her helper node) is able to occupy Location 1 and obtain $h_{1_{t_2}}$ correspondingly. As a result, if the variation of the channel between Alice and Location 1 during the period between t_1 and t_2 is small, i.e., $h_{1_{t_1}} \approx h_{1_{t_2}}$, the eavesdropper is able to compute the manipulated channel $h_{m_{t_1}}$ and then infer the old key K_1.

Lemma 1 demonstrates that when $h_{1_{t_1}} = h_{1_{t_2}}$ satisfies and the eavesdropper occupies Bob's old location, she is able to calculate the manipulated channel $h_{m_{t_1}}$. Therefore, the old key K_1 established at time t_1 will become insecure at time t_2, verifying that an old key cannot be reused once Bob moves.

Lemma 1. *Let $h_{AE_{t_1}}$ and $h_{AE_{t_2}}$ denote the real channels between Alice and Eve at time t_1 and t_2 respectively. $s_{a_{t_1}}$ and $s_{a_{t_2}}$ denote the manipulated transmission signals at time t_1 and t_2 respectively. Given $h_{1_{t_1}} = h_{1_{t_2}}$, Eve is able to compute $h_{m_{t_1}}$ with the knowledge of $h_{1_{t_2}}$, $h_{AE_{t_1}}$, $h_{AE_{t_2}}$.*

Proof. Let s denote the public transmission signal. We have

$$s_{a_{t_1}} * h_{1_{t_1}} = s * h_{m_{t_1}}, \tag{4}$$

Meanwhile, the received signal $r_{e_{t_1}}$ at Eve is

$$r_{e_{t_1}} = s_{a_{t_1}} * h_{AE_{t_1}}. \tag{5}$$

As Bob moves to Location 2, Eve can occupy Bob's old location, i.e., Location 1, by putting a helper node there. The observed signals at Location 1 and Eve are then represented with $r_{1_{t_2}}$, $r_{e_{t_2}}$ respectively,

$$\begin{cases} r_{e_{t_2}} = s_{a_{t_2}} * h_{AE_{t_2}} \\ r_{1_{t_2}} = s_{a_{t_2}} * h_{1_{t_2}}. \end{cases} \tag{6}$$

Based on Equation 6, Eve can solve $h_{1_{t_2}}$ with $r_{1_{t_2}}$, $r_{e_{t_2}}$ and $h_{AE_{t_2}}$. Since $h_{1_{t_1}} = h_{1_{t_2}}$ holds, and $s_{a_{t_1}}$ can be obtained with $r_{e_{t_1}}$, $h_{AE_{t_1}}$ based on Equation 5, Eve can then calculate the manipulated channel $h_{m_{t_1}}$ based on Equation 4. □

With $h_{m_{t_1}}$, Eve can utilize the key retrieval process as described in Section 4.4 to recover the old key K_1.

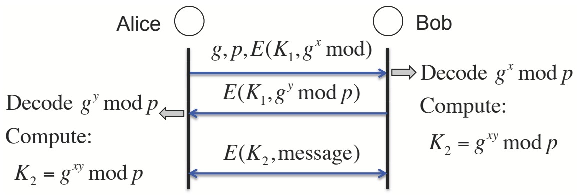


Fig. 5. Authenticated Diffie-Hellman key exchange.

4.5.2 Generation of the New Key

Section 4.5.1 demonstrates that the old key K_1 becomes insecure when Bob moves to Location 2. Thus, Alice and Bob needs to re-establish another key, denoted with K_2, in order to continue the current communication securely. There exist two intuitive ways to generate K_2, i.e., traditional cryptography (e.g., Diffie-Heffman key change) and another round of the proposed key establishment, and these are discussed below.

Authenticated Diffie-Hellman Key Exchange: First, if Alice and Bob have sufficient computational capability, they can launch Diffie-Hellman key exchange [23] to generate K_2. However, traditional Diffie-Hellman algorithm cannot be directly used due to its vulnerability to man-in-the-middle attacks; Alice or Bob might be inadvertently establishing a secret key with an adversary.

To defend against man-in-the-middle attacks, when Bob is still at Location 1, we enable Alice and Bob to encrypt the Diffie-Hellman key exchange with the established key K_1, which can be utilized to later authenticate each other. Figure 5 shows the corresponding procedures. Alice and Bob pick random numbers x and y respectively. Alice firstly sends Bob the chosen numbers p and g, as well as $g^x \bmod p$ encrypted with K_1. With K_1, Bob can decode the message and obtain $g^x \bmod p$. Similarly, $g^y \bmod p$ can be obtained by Alice. As a result, either Alice or Bob can calculate the secret key $K_2 = g^{xy} \bmod p$, and use it to enable secure communications between Alice and Bob wherever Bob moves to subsequently.

After K_2 is established and Bob moves to a different location, the old key K_1 may be disclosed, and thus an eavesdropper may successfully decode $g^x \bmod p$ and $g^y \bmod p$ with K_1. However, the adversary is still unable to extract the newly established key K_2 as it is computationally infeasible to compute g^{xy} from g^x and g^y.

Re-launching the Proposed Key Establishment: Now we consider situations where each communicator has limited computational capability and cannot launch Diffie-Hellman key exchange. Thus another around of manipulatable wireless key establishment can be utilized to generate the new secret key K_2. For K_2, Bob actually has two choices.

- K_2 is arbitrarily selected. Alice and Bob are required to restart communication as Alice will not know Bob's identity due to his unexplained appearance at a new location. The previous secure communication begun when Bob is at Location 1 would not therefore continue immediately.
- K_2 is carefully selected in order to let Alice identify Bob as a previous communication participant. In this case, they will be able to continue the previous communication.

The first choice only requires Bob to launch another round of the proposed channel manipulation scheme with Alice. This is obviously not efficient, as ongoing communications between

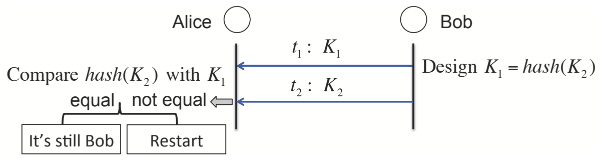


Fig. 6. Alice distinguishes Bob.

Alice and Bob will be disrupted each time Bob moves. In contrast, the second choice enables Bob to maintain continuity of communications. However, while Alice should be able to identify Bob using K_2, we must also ensure Eve cannot compute K_2 after deriving K_1.

We craft keys such that $K_1 = hash(K_2)$, where $hash$ denotes a selected one-way hash function. Therefore, as shown in Figure 6, at time t_2, Alice can verify Bob's identity by comparing K_1 and the cryptographic hash value of K_2. If Bob is not verified, they must restart the communication process as Alice will see Bob as a new contact. Meanwhile, even when Eve successfully recovers the old key K_1, she is still unable to infer the current key K_2 as K_1 does not disclose useful information about K_2.

4.5.3 Key Establishment Chain

Now we consider the general situation, where Bob continuously moves to different locations. Note that if the authenticated Diffie-Hellman key exchange is used to generate a new key, this key will not be leaked as Bob moves, and so can be reused perpetually. Here, we focus on communicators with limited computational ability, where, because of the potential stalking threat, Bob must re-establish a new key with Alice upon each movement. Also, so Alice can identify him, Bob needs to select a new but identifiable secret key once he moves.

We generalize the solution presented in Section 4.5.2 and propose a key establishment chain technique to not only continuously verify a moving Bob to Alice but also guarantee the confidentiality of each newly selected key. Here, Bob pre-computes a series of keys which are chained together by way of a one-way hash function, and whenever he moves to another location, Bob will send each successive key in the chain to Alice using the channel manipulation method. Specifically, Bob chooses the value N to be the maximum number of potential locations for which he will need to establish a new secret key with Alice. Alice and Bob choose a public hash function h, which Bob uses to pre-compute the chain of keys, beginning with the key to be used last (K_N) and hashing each key to compute the one to be used before it. The newly computed keys in order of their use satisfy

$$K_i = hash(K_{i+1}), i \in \{1, \cdots, N-1\}. \tag{7}$$

In all, $K_1 = hash(K_2) = hash(hash(K_3)) = \cdots = hash^{N-1}(K_N)$. In this manner, whenever Bob moves, Alice can verify that he is establishing a new key with her by matching the old key with the hash of the new, and thereby the two can maintain contiguous secure communications. However, due to the one-way hash property, it is computationally infeasible for an eavesdropper to derive the new key from the old. Note that the value of N can be large, and a chain of keys can be constructed prior to Bob even knowing he wishes to communicate with Alice. This prevents any delay at the time of their introduction, and not all keys need be used.

4.6 Security and Performance Analysis

4.6.1 Eavesdropping Attacks

The correlation coefficient ρ between the two channels observed at the receiver and the eavesdropper respectively can be present with $J_0(2\pi\frac{D}{\lambda})$, where $J_0(\cdot)$ is the first kind Bessel function of order zero, and D is the distance between the receiver and the eavesdropper [24], [25]. When $D/\lambda \geq 1/2$, ρ approaches 0. That means, when the eavesdropper and the receiver are spaced at least a half wavelength apart, it yields zero correlation. However, in the real world, there is poor scattering and/or a strong line-of-sight component, the spatial separation between the receiver and the eavesdropper should be longer (e.g., several wavelengths) in order to obtain uncorrelated channels [26]. Generally, the eavesdropper may employ two different schemes to obtain the manipulated channel (and thus extract the secret key):

Direct Observation: It is unlikely the eavesdropper could occupy the same location with the transmitter or receiver, as the exposure risk would be dramatically increased. Due to the spatial uncorrelation property, a small location difference (e.g., several wavelengths) would cause an observed channel change, and following discrepancies in the extracted keys.

Indirect Observation: A further concern is whether it is possible for the eavesdropper to calculate the manipulated channel when she is not at the same location with the receiver. As mentioned in Section 4.3, the manipulated channel $h_m(t) = w(t)*h(t)$. So to learn $h_m(t)$, an eavesdropper should not only know the impulse response $w(t)$ of the channel manipulation process, but also the real channel impulse response $h(t)$ between the transmitter and the receiver.

Let $r_e(t)$ denote the signal received by the eavesdropper when the transmitter launches channel manipulation, and let $h_e(t)$ denote the real channel impulse response between the transmitter and eavesdropper. Thus we have $r_e(t) = s(t) * w(t) * h_e(t)$. Therefore, to learn $w(t)$, the eavesdropper must learn $h_e(t)$. However, the transmitter can always hide its real channel or stay silent before launching key establishment, thus the eavesdropper would fail to obtain $w(t)$. Besides, we allow the transmitter to randomize the channel manipulation process and thus the value of $w(t)$ can be updated at any time, so that the eavesdropper will require far more effort. Even $w(t)$ is disclosed, $h(t)$ is also unknown to the eavesdropper due to the same reason as in the first scheme (i.e., the eavesdropper is unable to put a helper node co-located with the transmitter to measure the real channel). Therefore, this attack requirement is more stringent than the previous.

4.6.2 RS Code Impact

Due to the noise and hardware differences, the quantized bit sequence at the receiver may have bit discrepancies with the key K_a selected by the transmitter. With RS code, the receiver can obtain the secret key K_b and $K_b = K_a$. One concern is whether the enforced RS code can help the eavesdropper to correct those mismatched bits as well, leading that her extracted key K_e is same with K_a. We define a term *channel proximity*, denoted with d_{ij}, to quantify the difference between two channels i and j, which equals the Euclidean distance between two obtained channel impulse responses, i.e., $d_{ij} = ||h_i - h_j||$. Channel proximity between respectively observed channels at an eavesdropper and a receiver highly depends on the distance between them.

Suppose C denotes the count of mismatched symbols in the quantized symbol sequence (e.g., $\{s_1, \cdots, s_n\}$) for a corresponding n-symbol codeword (e.g., $\{s^1_{enc}, \cdots, s^n_{enc}\}$) selected by the transmitter. We construct a function $M(\cdot)$ to model the relationship between the channel proximity d_{ij} with the count C, i.e., $C = M(d_{ij}) = \sum_{j\in\{1,\cdots,n\}}(s^j \oplus s^j_{enc})$ (if s^j and s^j_{enc} have mismatched bits, $s^j \oplus s^j_{enc}$ equals 1, otherwise 0). Theoretically, when $d_{ij} = 0$, i.e., the two observed channels are totally the same, the generated bit sequences from them should be the same, i.e., $C = 0$. While if $d_{ij} = \infty$, i.e., the two target channels are totally different, the worst case is that all symbols in the generated two bit sequences are different, i.e., $C = n$. We utilize d_{tr} and d_{te} to denote the channel proximities between the estimated channels at the transmitter and the receiver, and at the transmitter and the eavesdropper, respectively. In the experiment (i.e., Section 5.4), we show that $M(\cdot)$ is a monotonically increasing function in practice. Lemma 2 presents that with an appropriately selected RS code, the transmitter and the receiver are able to establish a secret key in the presence of an eavesdropper.

Lemma 2. *Suppose that $M(\cdot)$ is monotonically increasing, when the selected $RS(n,\ k)$ satisfies $d_{tr} \leq M^{-1}((n-k)/2) < d_{te}$, $K_a = K_b \neq K_e$ can be achieved.*

Proof. The error correction capability of $RS(n,\ k)$ is to correct $(n-k)/2$ symbols per codeword. We set a channel proximity threshold d_0, above which the two channels are regarded as different, while below we regard them are identical. Due to the property of wireless channel reciprocity, the estimated channels at the transmitter and the receiver would be almost the same, i.e., $d_{tr} \approx 0$. When $d_{tr} \leq M^{-1}((n-k)/2)$, we have $M(d_{tr}) \leq (n-k)/2$. Thus, the errors at the receiver can be corrected by the RS code, i.e., $K_a = K_b$.

On the other hand, due to the channel uncorrelation property, the estimated channels at the eavesdropper (e.g., more than several wavelengths away from the receiver) and the transmitter would be totally different, i.e., $d_{te} > d_0$. When $M^{-1}((n-k)/2) < d_{te}$, we have $M(d_{te}) > (n-k)/2$. Thus, the errors at the eavesdropper are unable to be corrected by the RS code, i.e., $K_e \neq K_a$. □

4.6.3 Active Attacks

An active adversary may try to interrupt, intercept, block or overwrite the transmit signals to disrupt the legitimate receiver extracting the secret key specified by the transmitter. For example, an adversary may jam the communication between the transmitter and the receiver so that the receiver may obtain an incorrect signal. Those attacks are not unique to our scheme, and all previous wireless key establishments are vulnerable to them. For example, Eberz *et al.* [27] propose a practical man-in-the-middle attack, in which an active attacker is able to impersonate both participants by injecting spoofed packets so Alice and Bob agree on a common key which the attacker knows. Also, an active eavesdropper is able to launch pilot contamination attack by transmitting the training sequence at the same time as Bob transmits [28]. The impact of this attack is that it reduces the accuracy of Alice's estimate of her outgoing channel to Bob and thus may fail traditional wireless key establishments due to the compromise of the wireless channel reciprocity. In our scheme, however, any injection or jamming by the attacker only changes the key the receiver extracts but is unable to sabotage the key at the transmitter, because the key at the transmitter is always specified by the transmitter itself, instead of extracted from the received signal. As a result, those active attacks would cause discrepancies easily detectable by the transmitter and the receiver. Besides, some techniques have been proposed to remove such active attacks. For example, to defend against jamming attacks, researchers have proposed spread spectrum approaches like Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) [29], [30]. Also, researchers have proposed pilot contamination detection methods like a self-contamination technique [31] and an uncoordinated frequency shift (UFS) scheme [32]. We can combine those techniques and the proposed scheme in order to successfully establish a key under active attack scenarios.

4.6.4 Key Establishment Rate Analysis

It has been explored that key establishment via physical layer characteristics can achieve information-theoretical security [2], [11], [14], [33]. Let $\mathbf{h}_{est_E}$ denote the estimated channel vector at the eavesdropper when Alice launches the channel manipulation. For any $\epsilon > 0$, and sufficiently long period T, if there exists the specified key K_A at Alice and the extracted key K_B at Bob making the key establishment system satisfy

$$\Pr\left(K_A \neq K_B\right) < \epsilon, \tag{8}$$

$$\frac{1}{T} I\left(K_A; \mathbf{h}_{est_E}\right) < \epsilon, \tag{9}$$

$$\frac{1}{T} H\left(K_A\right) > R - \epsilon, \tag{10}$$

$$\frac{1}{T}\log|\mathcal{K}| < \frac{1}{T} H\left(K_A\right) + \epsilon, \tag{11}$$

then R in (10) is the achievable key establishment rate, where $I(\cdot)$ denotes mutual information, and $\mathcal{K}$ is the alphabet of the selected key. (8) means Alice and Bob can establish the same key with a high probability. (9) means the manipulated message sent by Alice leaks no information to the eavesdropper, which guarantees the security of the established key. (11) ensures the selected cryptographic key is uniformly distributed.

The upper bound, denoted with R_{Key}, for the key establishment rate is equal to the conditional mutual information between the specified channel vector $\mathbf{h}_m$ at Alice and the estimate channel vector $\mathbf{h}_{est}$ at Bob [33], i.e., $R_{\text{Key}} = I(\mathbf{h}_m; \mathbf{h}_{est}|\mathbf{h}_{est_E})$. When the eavesdropper and Bob do not share a same location and their estimated channels are independent, we then obtain $R_{\text{Key}} = I(\mathbf{h}_m; \mathbf{h}_{est}) = \frac{1}{T} I\left(h_m; h_{est_E}\right)$, where T is the interval within which the eavesdropper obtains the channel estimate h_{est_E}.

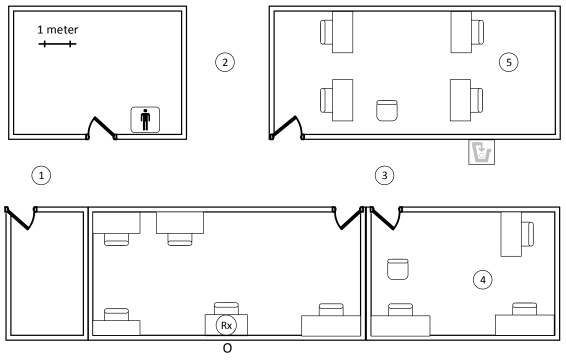


Fig. 7. The floorplan of the key establishment experiment.

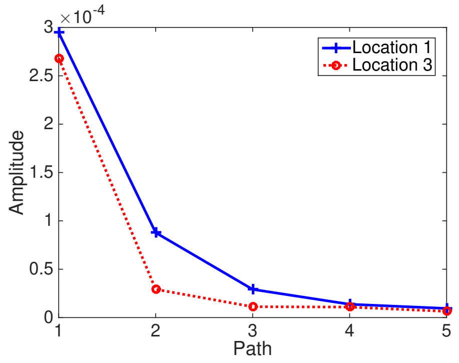


Fig. 8. Channels estimated at two different places.

A static channel refers to a wireless channel with long coherence time. Due to lack of variations in static channels, the generated bits in traditional wireless key establishments may have low entropy, and the corresponding key establishment rate approaches to 0. With the proposed key establishment scheme, however, though the real channel $h(t)$ between the transmitter and the receiver remains unchanged, the manipulated channel $h_m(t)$ can be dynamic since the impulse response $w(t)$ of channel manipulation is arbitrarily selected by the transmitter and $h_m(t) = w(t) * h(t)$ holds. Therefore, the proposed scheme can still generate a high-entropy key.

5 Experimental Evaluation

We implement the proposed key establishment on top of USRPs, which are equipped with AD and DA converters as the RF front ends, and SBX daughter boards operating in the 0.4∼4.4 GHz range as transceivers. The software toolkit is GNURadio [34]. We first describe our testing methodology, then evaluate the key establishment performance.

5.1 Methodology

Our prototype system consists of a transmitter (Tx), a receiver (Rx), and an eavesdropper (Ex). Each is a USRP N210 connected to a PC. Tx and Rx aim to establish a secret key in the presence of Ex. We performed the key establishment in a campus building with small offices and assorted furniture, forming a rich multipath environment. As shown in Figure 7, Rx defines the origin O, and Tx is tested in 5 other locations (L1∼L5). We compare the obtained key and the key specified by Tx to determine whether the key establishment succeeds, and calculate the *success rate* (i.e., the ratio between the

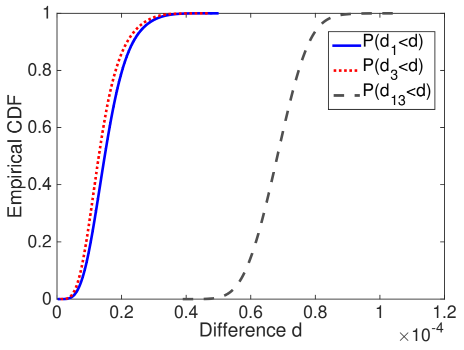


Fig. 9. CDF functions of channel differences.

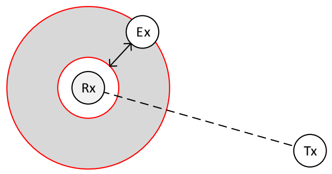


Fig. 10. Setup of eavesdropper (Ex).

number of successful key establishments and the total number of key establishment trials) via a large range of experiments.

Intuitively, the success rate can be affected by the setting of key mapping (i.e., the path differential), the RS ECC in use for key establishment, the manipulated multipath count, and the key length. In the following experiments, we explore how these factors can impact the key establishment performance.

5.2 Measuring Channel Proximities

We first verify the spatial uncorrelation property in the experimental environment. As shown in Figure 7, we place Tx at one of the five locations, Ex at another, and Rx at the origin O, repeating the experiment for every unique pair of locations. Both Tx and Ex estimate the channel impulse response between their respective locations and Rx. Note that existing channel estimation algorithms assume a resolvable multipath, and we usually configure the maximum number of resolvable multipaths to an empirical constant value depending on wireless system setups [15]. In this experiment, we consider a channel with five multipath components for our proof-of-concept implementation. We perform 1000 estimates at each location, and thus obtain 1000 estimation values of the corresponding channel impulse response, denoted by length-5 vectors containing each of the five components. We then calculate the mean of the estimated channel impulse response values for each path in the vector. As an example, Figure 8 shows the mean values of channel estimation results at Locations 1 and 3. The observed channels at two places clearly comprise different shapes, especially in the first three path values.

Figure 9 plots the empirical cumulative distribution functions (CDFs) of the channel proximities d_1 and d_3 between two channel impulse responses estimated at Locations 1 and 3, respectively, as well as the channel proximity d_{13} between one estimated at L1 and one at L3. We can see that the probability that d_{13} is bigger than d_1 or d_3 is almost 100%. This means we can easily distinguish channels observed at L1 and L3, implying validity of the spatial uncorrelation property of wireless channels. Channel estimation for other pairs of the

TABLE 1
Channel proximity ($\times 10^{-4}$) under different f_c

f_c (GHz)	λ (m)	0.25m	0.5m	0.75m	1m
1.2	0.25	0.11	0.35	0.63	0.85
2.4	0.125	0.20	0.51	0.90	0.92

5 locations demonstrate similar results and indicate our key establishment scheme should perform well, as we illustrate through the rest of this Evaluation.

We set the central frequency as 1.2GHz. To explore the relationship between the calculated channel proximity at Ex and the distance between Ex and Rx, we change the distance between Rx and Ex every 0.25m, starting from 0.25m (i.e., a wavelength for 1.2GHz signals), and each time we calculate and record the corresponding average channel proximity between the estimated channel at Ex and that at Tx. Table 1 demonstrates the impact of the distance on the observed channel proximity. We can see with the distance between Ex and Rx increasing, the channel proximity becomes larger, and then maintains a stable high value, which demonstrates that the observed channels at Tx and Ex are uncorrelated.

Besides, we increase the central frequency to 2.4GHz and re-calculate the corresponding channel proximities. A larger central frequency f_c brings a shorter signal wavelength λ (i.e., $\lambda = \frac{3\times 10^8}{f_c}$) in theory and therefore decreases the distance required for obtaining an uncorrelated channel. As shown in Table 1, for the same distance, Ex observes a higher channel proximity when the central frequency is 2.4GHz, which means the observed channel at Ex in this case has bigger differences with the receiver's channel.

To protect the secret key from Ex, we should keep the observed channels at Ex and Rx uncorrelated. Thus, we keep Ex at least 1m away from Rx, as Table 1 shows that 1m can guarantee that the observed channel at Ex is uncorrelated with that at Rx. To measure the confidentiality of the generated secret key, we compute and compare the success rates for Ex in breaking the secret key at varying distances away from Rx. As shown in Figure 10, we draw a circle originating at Rx and place Ex at a radius ranging outward from 1 to 4 meters.

5.3 A Case Study

In this section, we provide a simple walkthrough to illustrate the feasibility of the proposed method. We use a 3-bit sequence (a part of a secret key) as an example. The path differential q is set at 10^{-3}, and we make the first path response $h_{m1} = 3*10^{-3}$. Applying the key mapping mechanism to a sequence "100", we have $h_{m2} = h_{m1} + q$ and $h_{m3} = h_{m1} - 2q$. We thus generate the manipulated channel impulse response $\mathbf{h}_m = [3\ 4\ 1\ 2\ 2] * 10^{-3}$. Note that with such a 3-bit sequence (to which we are not applying ECC), the last two elements of $\mathbf{h}_m$ do not participate in the process and may be arbitrary values.

With the manipulated and real channel impulse responses between Tx and Rx, Tx generates and transmits the manipulated signal. On the other end, Rx estimates the channel impulse response utilizing the received signal from Tx and the public probe signal. Then Rx quantizes the estimated channel impulse response and retrieves a bit sequence.

We place Tx at Location 1 and Ex at Location 3. Figure 11 shows the corresponding normalized channel impulse results.

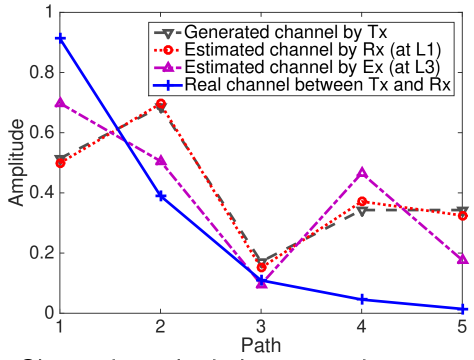


Fig. 11. Channel manipulation example.

The estimated channel at Rx is quite similar to the specified channel at Tx as desired. Using this estimate, Rx extracts a bit sequence of "100" based on the quantizer $Q(\cdot)$, a correct match with that which Tx sent. Meanwhile, applying the quantizer $Q(\cdot)$ on the channel impulse response estimated by Ex, the bit sequence "000" is returned, which differs from that Tx sent. The transmitter and the receiver have successfully established a 3-bit sequence while evading the eavesdropper.

5.4 Path Differential Selection

In the key mapping stage, we set the normalized value of the first path of the manipulated channel impulse response to 0.5, and vary the value of the path differential q from 0.05 to 0.2, with increments of 0.05. For each q, we perform 1000 attempts of establishing a 3-bit key not involving RS ECC between Tx and Rx. Figure 12 shows the success rate of 3-bit key establishment for Rx and Ex when Rx is placed at L1 and Ex is varied from 1 to 4 meters away. We can observe three major tendencies. First, larger distances between Rx and Ex generally result in a lesser ability for Ex to extract the key. When Ex is 1 meter away from Rx, the success rate to break the 3-bit key ranges from 23.7% to 33.6%, while when the distance increases to 4 meters, the success rate falls in the range of 12.8% to 19.3%, which is almost equivalent to the success rate of a random guess (the chance for a random guess to hit the correct 3-bit key is 12.5%). This appears due to the channels for Rx and Ex diverging with increased distance between them. This demonstrates the count of mismatched symbols normally increases with the distance between Rx and Ex, or the observed channel proximity at Ex, i.e., the function $M(\cdot)$ is really monotonically increasing in practice. Second, the success rate at Rx is always much higher that that at Ex. For example, when q value is 0.1, the success rate at Rx and Ex 4 meters away are 83.3% and 14.0% respectively. Finally, the success rate of key establishment increases with the increasing q. This is because larger q further separates the path responses generated by Tx to better overcome interference.

Though the success rate of establishing a 3-bit key at Rx can reach as high as around 88.9%, this success rate may not be reliable enough to secure private communications in practice. In order to eliminate the bit inconsistence between the specified key bits at Tx and the quantized bits at Rx, we introduce RS ECC to encode the secret key. As $q = 0.1$ achieves the largest difference between the success rate at Rx and that at the surrounding Ex in this initial test for success rate, we employ $q = 0.1$ for the following discussions.

5.5 RS Code Integration

In this section, we investigate the success rate after incorporating RS code and explore how to select an appropriate RS code. Intuitively, the chosen RS code should make sure that the success rate at Rx can be increased close to 100%, without improving the success rate at Ex and rendering insecure the established key between Tx and Rx.

Our selection criteria for enforced RS code is that its error-correction capability should exceed the $failure\ rate$ (i.e., $1-$ $success\ rate$) of key establishment at Rx while remaining below the failure rate at Ex. Thus, the success rate at Rx should increase to almost 100% while that at Ex stays low. As shown in Figure 12, when $q = 0.1$, the failure rate at Rx is 16.7%, while that at Ex placed at various distances from Rx is at most 73.5%. So we utilize RS(7, 3) which is able to correct 2 symbol errors for a 7-symbol codeword, which is more than the expected 1.2 symbol errors from a 16.7% failure rate at Rx but less than the expected 5.1 symbol errors from Ex's 73.5% failure rate. To further explore the effect of RS code choice, we enlarge the length of a RS codeword and select another three RS codes under the guidance of the aforementioned selection criteria: RS(15, 7) (4 bits per symbol), RS(31, 17) and RS(31, 15) (5 bits per symbol). For each RS code, we perform 5000 attempts of key establishment using five RS codewords as an example and record the success rate at Rx and the Ex around. Note that communicators may establish keys from multiple channel estimations and combine the bits from each to form an aggregate secret key.

Figure 13 plots the success rate of key establishment for Rx, we can observe that the success rate for Rx decreases as the code efficiency (i.e., k/n for RS(n, k)) increases. For example, the code efficiency of RS(7, 3) is 42.9% and its success rate is as high as 97%, while the code efficiency of RS(31, 17) is 54.8% and the corresponding success rate decreases to 78.7%. However, for RS code of higher code efficiency, its encoded key size becomes shorter. A RS(7, 3) codeword can encode a key of length $3 \times 3 = 9$ bits whereas a RS(31, 17) codeword can encode a key of length $17 \times 5 = 85$ bits.

Figure 14 shows the success rate at Ex of different distance away with Rx. Compare with Rx's high success rate (78.7%$\sim$97%), the success rate at Ex is markedly low, ranging from 0.09% to 0.47%. The chosen RS code is unable to correct most errors incurred at Ex, so that while ECC allows the correct decoding of the key at Rx, Ex is unable to use it to decode the key as Rx does. This is because the channels of Ex and Rx are strongly uncorrelated, and the artificial channel impulse response observed by Ex exhibits more bits mismatched with the actual key than Rx. These extra mismatched bits overflow the decoding capability of ECC, thereby leading to the incorrect decoding of the key at Ex.

Choosing the RS Code: In general, when the channel of the eavesdropper is uncorrelated with that of the receiver, it is highly likely that a typical RS code would allow the receiver to reconstruct the key with a high success rate and meanwhile yield a very low success rate at the eavesdropper. To further refine the code selection, an empirical profile may be built to reveal the relationship between the number of bit errors and the distance from the receiver. Assuming that the eavesdropper is at least d meters away from the receiver, the communicators can then look up the profile to determine an appropriate ECC with an error correction capability ranging between x and y, where x is the number of bit errors typically encountered by the receiver and y is the number of bit errors measured d meters away from the receiver.

TABLE 2
Success Rate (%) vs. key length (bits)

Key length	Rx	Ex-1m	Ex-2m	Ex-3m	Ex-4m
112	96.0	1.56	1.00	1.21	0.94
140	94.8	0.42	0.29	0.39	0.32
168	91.0	0.10	0.06	0.04	0

TABLE 3
Success Rate (%) vs. value of L

Value of L	Rx	Ex-1m	Ex-2m	Ex-3m	Ex-4m
3	94.8	0.42	0.29	0.39	0.32
4	92.2	0.34	0.29	0.39	0.30
5	86.3	0.31	0.31	0.41	0.27

5.6 Key Length and Manipulated Path Count

In order to provide effective protection for private communications, the secret key that the transmitter and the receiver intend to establish should be suitably long. Table 2 shows the relationship between the success rate of key establishment and key length when we utilize RS(15, 7) code to encode the key. This appears as one would expect when varying the length of typical cryptographic keys, demonstrating a clear divergence between success rates for the receiver vs. the eavesdropper with higher key length. Indeed, the success rate for eavesdroppers lowers to 0% with a 168-bit key, but key establishment between legitimate users is still successful.

At the key mapping stage, the transmitter maps a bit sequence with length L_b into a manipulated channel impulse response vector with size L. Correspondingly, the receiver will generate L_b bits as the secret key from an estimated channel impulse response. Previous experiments discuss the situation when $L = 3$, and we now analyze the effect of different L on the performance. Essentially, the number L of manipulated paths in a channel impulse response determines the number of bits that the transmitter can include per channel impulse response. Based on the relationship $L_b = L - 1 + \lfloor L - 1 \rfloor / 2$ mentioned in Section 4.2, when L is set to 5, L_b would be 6. Table 3 shows the relationship between the success rate of 140-bit key and the number of manipulated paths, using RS(15, 7) code. We see that the success rate at the receiver decreases with the value of L increasing. When $L = 5$, the success rate moves to below 90%; with more paths to manipulate, the key establishment is more susceptible to channel noise. The success rates at the surrounding eavesdroppers, however, do not decrease notably and varies between 0.27% and 0.42%, again demonstrating that the success rate at the eavesdropper mainly depends on the uncertainty of its observed channel.

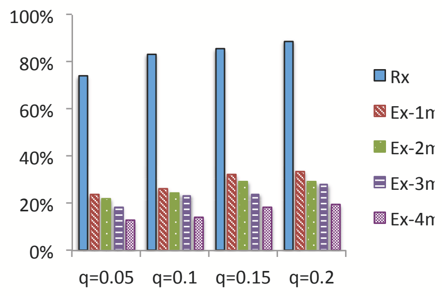


Fig. 12. Success rate vs. value of q.

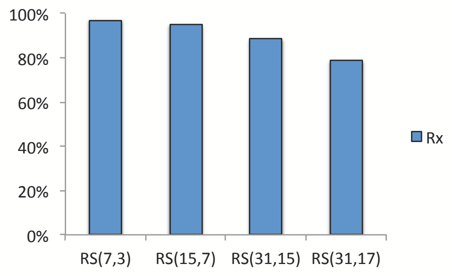


Fig. 13. Success rate vs. RS code type.

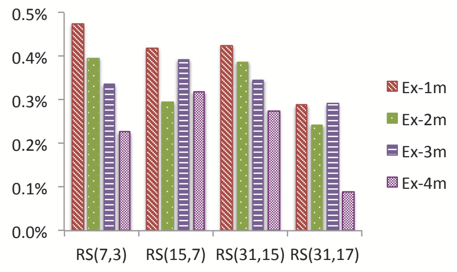


Fig. 14. Success rate for Ex.

TABLE 4
Success Rate (%) of Key Establishment for a Wide Range of Locations

	L2			L3			L4			L5		
L	3	4	5	3	4	5	3	4	5	3	4	5
Rx	91.1	87.5	86.0	92.2	87.6	81.2	92.4	89.4	88.1	91.1	85.5	82.9
Ex-1m	0.12	0.12	0.10	0.14	0.10	0.08	0.12	0.08	0.06	0.12	0.10	0.08
Ex-2m	0.10	0.08	0.10	0.12	0.10	0.08	0.10	0.06	0.06	0.14	0.08	0.06
Ex-3m	0.06	0.04	0.02	0.08	0.04	0.04	0.06	0.02	0.02	0.06	0.04	0.02
Ex-4m	0.02	0	0	0	0	0	0	0.02	0	0	0	0

5.7 Overall Performance

Table 4 shows the success rates at the receiver and surrounding eavesdroppers when Tx is placed at each of the locations other than L1 and establishes a 168-bit key utilizing RS(15, 7) code. The number L of manipulated paths ranges from 3 to 5. The success rate at the receiver (ranging from 81.2%$\sim$92.2%) is visibly much higher than those at the eavesdroppers (0%$\sim$0.14%), and successively larger separation causes the eavesdropper's success rate to continue dropping, especially when Ex lies in 4 meters distance away from Rx, it is almost impossible for her to break the key established between Tx and Rx. Through the whole experiment, we show

- ECC improves the success rate for Rx while causing no increase in Ex's effectiveness. This follows from the ability to choose an RS code resolving a number of bit errors larger than Rx typically encounters but smaller than the number by Ex.
- Increasing key length to sizes typical of modern keys reduces Ex to near zero efficacy without decreasing the success rate of key establishment.
- The performance holds steady at all our tested locations and so is concluded to be robust in practice.

6 Related Work

The use of physical layer characteristics of a wireless channel for key generation has formed a fruitful research area in recent decades, while existing wireless key establishments focus on generating keys directly from the channel. They select a certain channel metric and quantize it to obtain bits to form a key. A number of metrics have been explored, such as signal envelopes [1], channel impulse response [2], [8], [10], signal phases [11], received signal strength [3], [7], [12], and the frequency-selectivity of channel fading [35]. In these schemes then, the established key is highly dependent upon the selected channel metric while the communicators themselves hold little control. In our work, we enable the transmitter to specify and control the key at will. Thus, our scheme makes it easier for a transmitter to establish a same key with multiple receivers than existing wireless key establishments. For example, the transmitter in our scheme can act a public key server [23] and distribute a same key to different legitimate receivers. Another significant distinction lies in our work, requiring no reconciliation, where previous efforts [1]–[3], [7]–[11] need it to correct mismatched bits. [2] points out that assuming Alice and Bob share an authorized channel during this process is unrealistic, leading to the possibility of spoofing attacks. As we do not perform reconciliation, we have no such concerns.

Note that ECC is often utilized to develop reconciliation schemes for key establishments as it can identify and correct the bit errors [36], [37]. Those schemes still require key-related message exchange between two parties. This is because the secret key is unknown to both parties till the end of the key establishment process, each party cannot determine mismatched bits without message exchange. However, in our work, the secret key is pre-selected by the transmitter so that it can be encrypted with ECC, e.g., RS code, to defend against the transmission errors when it reaches the receiver. Thus, the reconciliation phase of message exchange is not necessary.

Some other work also utilize manipulated channel information to achieve different goals (e.g., [38], [39]). For example, [38] utilized false channel information to enable the transmitter to hide its location or impersonate another user's location while [39] utilized fake channel information to eavesdrop information received by a target user. In our work, however, we construct a manipulated channel and use it to establish secret keys between legitimate communicators.

7 Conclusion

We propose a novel wireless key establishment technique between a transmitter and receiver pair in the presence of an eavesdropper. Our scheme enables the transmitter to specify any content as the secret key and removes the reconciliation process, which is necessary in conventional wireless key establishments. The transmitter encodes the specified key with ECC, and delivers it to the receiver with a manipulated signal. The receiver uses channel estimation to obtain the encoded key, with potential errors, and then performs error correction to retrieve the secret key. We document real-world implementation on the USRP platform, demonstrating the feasibility and reliability of the proposed technique.

References

[1] B. Azimi-sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *Proc. of ACM CCS*, 2007.

[2] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proc. of ACM MobiCom*, 2008, pp. 128–139.

[3] 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 *Proc. of ACM Mobicom*, 2009, pp. 321–332.

[4] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *Proc. of IEEE ICASSP*, 2008, pp. 3013–3016.

[5] J. W. Wallace and R. K. Sharma, "Automatic secret keys from reciprocal mimo wireless channels: Measurement and analysis," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 3, pp. 381–392, 2010.

[6] M. Clark, "Robust wireless channel based secret key extraction," in *Proc. of IEEE Milcom*, 2012, pp. 1–6.

[7] X. Zhu, F. Xu, E. Novak, C. C. Tan, Q. Li, and G. Chen, "Extracting secret key from wireless link dynamics in vehicular environments," in *Proc. of IEEE INFOCOM*, 2013, pp. 2283–2291.

[8] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 2, pp. 240–254, 2010.

[9] Y. Liu, S. C. Draper, and A. M. Sayeed, "Exploiting channel diversity in secret key generation from multipath fading randomness," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 5, pp. 1484–1497, 2012.

[10] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *Proc. of IEEE INFOCOM*, 2013, pp. 3048–3056.

[11] Q. Wang, H. Su, K. Ren, and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *Proc. of IEEE INFOCOM*, 2011, pp. 1422–1430.

[12] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Trans. Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.

[13] K. Zeng, D. Wu, A.Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *Proc. of IEEE INFOCOM*, 2010, pp. 1–9.

[14] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography. i. secret sharing," *IEEE Trans. Inf. Theor.*, vol. 39, no. 4, pp. 1121–1132, 1993.

[15] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[16] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300–304, 1960.

[17] IEEE, "IEEE 802.11: Wireless LANs," 2007.

[18] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *Proc. of ACM MobiCom*, 2007, pp. 111–122.

[19] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Channel-based detection of sybil attacks in wireless networks," *IEEE Trans. on Inf. Forensics and Security*, vol. 4, no. 3, pp. 492–503, Sep. 2009.

[20] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *2010 IEEE Symposium on Security and Privacy*, May 2010, pp. 286–301.

[21] J. Xiong and K. Jamieson, "Securearray: Improving wifi security with fine-grained physical-layer information," in *Proc. of ACM MobiCom*, Miami, Florida, USA, 2013, pp. 441–452.

[22] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *Proc. of IEEE INFOCOM*, March 2012, pp. 927–935.

[23] C. Kaufman, R. Perlman, and M. Speciner, *Network security : private communication in a public world*, ser. Prentice Hall series in computer networking and distributed systems. Prentice Hall, 2002.

[24] G. D. Durgin and T. S. Rappaport, "Effects of multipath angular spread on the spatial cross-correlation of received voltage envelopes," in *49th IEEE Vehicular Technology Conference*, vol. 2, Jul 1999, pp. 996–1000.

[25] J. Salz and J. H. Winters, "Effect of fading correlation on adaptive arrays in digital mobile radio," *IEEE Trans. on Vehicular Technology*, vol. 43, no. 4, pp. 1049–1057, Nov 1994.

[26] X. He, H. Dai, W. Shen, and P. Ning, "Is link signature dependable for wireless security?" in *Proc. of IEEE INFOCOM*, 2013, pp. 200–204.

[27] S. Eberz, M. Strohmeier, M. Wilhelm, and I. Martinovic, "A practical man-in-the-middle attack on signal-based key generation protocols," in *Computer Security – ESORICS 2012*, 2012, pp. 235–252.

[28] X. Zhou, B. Maham, and A. Hjorungnes, "Pilot contamination for active eavesdropping," *IEEE Trans. on Wireless Communications*, vol. 11, no. 3, pp. 903–907, March 2012.

[29] M. Strasser, C. Pöpper, and S. Čapkun, "Efficient uncoordinated fhss anti-jamming communication," in *Proc. of ACM MobiHoc*, 2009, pp. 207–218.

[30] S. Fang, Y. Liu, and P. Ning, "Wireless communications under broadband reactive jamming attacks," *IEEE Trans. on Dependable and Secure Computing*, vol. 13, no. 3, pp. 394–408, May 2016.

[31] J. K. Tugnait, "Self-contamination for detection of pilot contamination attack in multiple antenna systems," *IEEE Wireless Communications Letters*, vol. 4, no. 5, pp. 525–528, Oct 2015.

[32] W. Zhang, H. Lin, and R. Zhang, "Detection of pilot contamination attack based on uncoordinated frequency shifts," *IEEE Trans. on Communications*, vol. 66, no. 6, pp. 2658–2670, June 2018.

[33] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theor.*, vol. 39, no. 3, pp. 733–742, 1993.

[34] "GNU Radio Software," http://gnuradio.org, [Online; accessed in 2014].

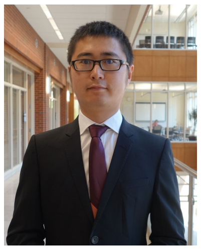
[35] M. Wilhelm, I. Martinovic, and J. B. Schmitt, "Secret keys from entangled sensor motes: Implementation and analysis," in *Proc. of ACM WiSec*, 2010, pp. 139–144.

[36] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, Mar. 2008.

[37] W. WANG, L. Yang, and Q. Zhang, "Resonance-based secure pairing for wearables," *IEEE Trans. on Mobile Computing*, vol. 17, no. 11, pp. 2607–2618, Nov 2018.

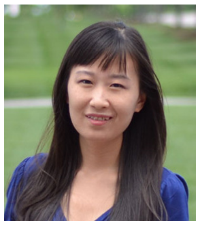
[38] S. Fang, Y. Liu, W. Shen, and H. Zhu, "Where are you from? Confusing location distinction using virtual multipath camouflage," in *Proc. of ACM MobiCom*, 2014, pp. 225–236.

[39] Y.-C. Tung, S. Han, D. Chen, and K. G. Shin, "Vulnerability and protection of channel state information in multiuser MIMO networks," in *Proc. ACM CCS*, Scottsdale, Arizona, USA, 2014, pp. 775–786.



Song Fang received his Ph.D. in computer science from the Univ. of South Florida in 2018. He is now an assistant professor at the School of Computer Science, Univ. of Oklahoma. His research interests are in the area of network security and system security.

Ian Markwood received his Ph.D. in computer science from the Univ. of South Florida in 2018. He is now a security researcher with BlackHorse Solutions, contracting for the US Government.



Yao Liu received her Ph.D. in computer science from the North Carolina State Univ. in 2012. She is now an associate professor at the Dept. of Computer Science and Engineering, Univ. of South Florida. Her research is related to computer and network security, with an emphasis on designing and implementing defense approaches that protect emerging wireless technologies from being undermined by adversaries. Her research interest also lies in cyber-physical systems security, especially smart grid security.