Practical Secret Key Agreement for Full-Duplex Near Field Communications

Rong Jin  
University of Michigan, Dearborn, MI 48128  
jinrong@umich.edu

Xianru Du  
University of Michigan, Dearborn, MI 48128  
duxianru@umich.edu

Zi Deng  
University of Michigan, Dearborn, MI 48128  
zdeng@umich.edu

Kai Zeng  
University of Michigan, Dearborn, MI 48128  
kzeng@umich.edu

Jing Xu  
Huazhong University of Science and Technology  
xujing@hust.edu.cn

ABSTRACT
Near Field Communication (NFC) is a promising short distance radio communication technology for many useful applications. Although its communication range is short, NFC alone does not guarantee secure communication and is subject to security attacks, such as eavesdropping attack. Generating a shared key and using symmetric key cryptography to secure the communication between NFC devices is a feasible solution to prevent various attacks. However, conventional Diffie-Hellman key agreement protocol is not preferable for resource constrained NFC devices due to its extensive computational overhead and energy consumption. In this paper, we propose a practical, fast and energy-efficient key agreement scheme, called RIWA (Random Bits transmission with Waveform shaking), for NFC devices by exploiting its full-duplex capability. In RIWA, two devices send random bits to each other simultaneously without strict synchronization or perfect match of amplitude and phase. On the contrary, RIWA randomly introduces synchronization offset and mismatch of amplitude and phase for each bit transmission in order to prevent a passive attacker from determining the generated key. A shared bit can be established when two devices send different bits. We conduct theoretical analysis on the correctness and security strength of RIWA, and extensive simulations to evaluate its effectiveness. We build a testbed based on USRP software defined radio and conduct proof-of-concept experiments to evaluate RIWA in a real-world environment. It shows that RIWA achieves a high key generation rate about 26kbps and is immune to eavesdropping attack even when the attacker is within several centimeters away from the legitimate devices. RIWA is a practical, fast, energy-efficient, and secure key agreement scheme for resource-constrained NFC devices.

Keywords
near field communication, practical key agreement, energy efficient, USRP

1. INTRODUCTION
Near Field Communication (NFC) provides a convenient proximity radio communication technology for many useful applications [1, 2, 3, 4], including contactless payment [5], identification, data exchange, and simplified device pairing [6]. Although the communication range of NFC is short, NFC alone does not prevent data modification and eavesdropping [1, 7]. Unfortunately, ISO standard offers no security technique for NFC [8]. Applications may use higher-layer cryptographic mechanisms to deal with security threats.

Previous works show that using symmetric key cryptography to establish a secure channel between two NFC devices is a feasible way to protect against various attacks [1]. A conventional mechanism used to generate a symmetric key between two devices is the Diffie-Hellman (D-H) key agreement protocol. However, since NFC devices are usually powered by battery with limited computation and communication capability, D-H key agreement protocol is not preferable for NFC devices due to its high computational overhead and energy consumption [9].

Recently, a low-cost key agreement scheme exploiting duplex capability of NFC devices has been proposed in [1, 2]. The basic idea is illustrated in Fig. 1 showing the baseband signal with Manchester encoding. Both NFC devices, say Alice and Bob, simultaneously send random bits of 0’s or 1’s. While sending, one (or both) of the devices can detect the receiving signal. When Alice and Bob send different bits (one sends 0 and the other sends 1), an eavesdropper only overhears the superposed signal as a flat voltage, so cannot figure out who sent the 0 or who sent 1. Since Alice and Bob know what they have sent and received, they can use the bits sent at either side to establish a shared key.

Unfortunately, this scheme is impractical due to the strict requirements of perfect synchronization [1] and perfect match of signal amplitude and phase, which are very hard to be satisfied in practice due to the following reasons. First, it is well known that time synchronization for distributed devices is a difficult problem [10] and suffers from clock drift and skew. Second, at RF front-end, digital baseband signal is convert-
ed to analog, up-converted to the carrier frequency, scaled to the right power, and sent. Due to the impairment of the device, the RF signal being transmitted is in fact distorted relative to the digital baseband representation [11]. As a result, it is nearly impossible for a device to exactly control what it is transmitting and making perfect synchronization and absolute match of amplitude and phase.

Once there is a slight synchronization offset or mismatch of the amplitude or phase, the key can be broken. An intuitive way to break the key is shown in Fig. 2 under non-perfect synchronization. An eavesdropper can determine the bit sent by Alice and Bob by observing the peak or valley in the superposed RF signal. Suppose Bob is ahead of Alice. If eavesdropper detects a valley, he would know that Alice sends 0 and Bob sends 1 and vice versa.

In this paper, aiming at a practical key agreement scheme tackling imperfect synchronization and mismatch of signal amplitude and phase, we propose RIWA (Random bIts trans-mission with Waveform shAkIng), which uses a novel idea of waveform shaking shown in Fig. 3. Suppose there is a slight synchronization offset of $\varepsilon$ between Alice and Bob. Rather than compensating $\varepsilon$, RIWA introduces another random synchronization offset $\gamma$ by shifting the waveform. $\gamma$ differs from $\varepsilon$ in that it has a random value and can be positive or negative, corresponding to right or left shifting. $\gamma$ disturbs the start moment of the bits and covers up $\varepsilon$. Attackers who use $\varepsilon$ to distinguish the RF signals of Alice and Bob will make mistakes. For amplitude and phase mismatch, we propose similar ideas of adding randomness to cover it up.

Our main contributions are as follows:

- We propose a key agreement scheme, RIWA, for NFC devices by exploiting its duplex capability without strict requirement of perfect time synchronization or absolute match of signal amplitude and phase.
- We prove the correctness of RIWA analytically, and show its effectiveness by simulations.
- We build a testbed based on USRP software defined radio and conduct proof-of-concept experiments to evaluate the effectiveness and security strength of RIWA in a real-world environment.

We make the following findings:

- RIWA can generate a shared key between two NFC devices at a high rate about 26kbps.
- RIWA is immune to eavesdropping attacks even when the attacker is a few centimeters away from the legitimate users.

2. SYSTEM MODEL
We consider two legitimate parties, say Alice and Bob, who want to extract a shared secret key via communication without any pre-shared secret. They are equipped with active
NFC devices which are full duplex enabled. That is, they can transmit random signals to each other while detecting the received signal. As specified by the ISO standard, Alice and Bob use Manchester encoding and ASK modulation [8]. They are close to each other within several centimeters, and operate at 13.5MHz carrier frequency.

2.1 Attacker Model
We consider a powerful passive attacker, Eve, who tries to compromise the generated key by eavesdropping on the communication between Alice and Bob. Eve possesses a high quality wireless channel in eavesdropping. Eve may have multiple antennas, and may be in any location with respect to the legitimate NFC devices. She may use standard or custom-built hardware. She can capture baseband and RF signals with high sensitivity and sampling rate. In this case, she can sense slight out of synchronization and the mismatch of signal amplitude and phase. Eve also has powerful computational ability. She can store all the overheard signals and conduct sophisticated signal processing or data analysis.

2.2 Signal Model
We assume Alice and Bob send random bits of 0’s and 1’s to each other simultaneously. The bit streams sent by Alice and Bob are $C_a$ and $C_b$, respectively.

$$C_a = \{C_a(1), C_a(2), \ldots, C_a(N_a)\}$$

$$C_b = \{C_b(1), C_b(2), \ldots, C_b(N_b)\}$$

(1)

where $C_a(i)$ and $C_b(i)$ ($1 \leq i \leq N$) are the $i$th bit sent by Alice and Bob, respectively.

We give the representation of the signal received at Alice, Bob, and Eve as follows.

2.2.1 Signal Model of Legitimate Users
The RF signals transmitted by Alice and Bob are

$$x_a(t) = h_a^r r^t(C_a) + n_a^r(t)$$

$$x_b(t) = h_b^r r^t(C_b) + n_b^r(t)$$

(2)

where $r^t(C_a)$ and $r^t(C_b)$ are baseband representation of $C_a$ and $C_b$. $x_a(t)$ and $x_b(t)$ are the RF waveform at carrier frequency. $h_a^r$ and $h_b^r$ represent the modulated transmission at the RF front-end. $n_a^r(t)$ and $n_b^r(t)$ are channel noises. Fig. 4 shows an example of ideal NFC modulation method of Manchester coding with 100% ASK [8, 3] is assumed.

Due to proximity, the RF signals received by Alice and Bob are inductive coupling of each other [3], and can be written as

$$y_a(t) = h_a^e x_a(t) + \eta x_b(t)$$

$$y_b(t) = h_b^e x_b(t) + \eta x_a(t)$$

(3)

where $\eta$ is coupling coefficient [12]. $h_a^e$ and $h_b^e$ are the channels of NFC devices’ self interference [13].

At receiving end, RF Signals are down converted to baseband

$$r_a^r = h_a^r y_a(t) + n_a^r(t)$$

$$r_b^r = h_b^r y_b(t) + n_b^r(t)$$

(4)

where $h_a^r$ and $h_b^r$ represent down converted channels. $n_a^r(t)$ and $n_b^r(t)$ are channel noises.

2.2.2 Signal Model of Attacker
Similar to (3), the RF signals received by Eve can be written as

$$y_e(t) = h_{ea} x_a(t) + \eta_{ea} x_b(t)$$

where $\eta_{ea}$ is the coupling coefficient between Eve and Alice. $\eta_{eb}$ is the coupling coefficient between Eve and Bob. The baseband envelop is:

$$r_e^r = h_e^r y_e(t) + n_e^r(t)$$

(5)

where $h_e^r$ is Eve’s down converted channel. $n_e^r(t)$ is the channel noise.

3. BASIC SCHEME IS NOT SECURE IN PRACTICE
In order to motivate our solution, RIWA, we describe the basic key generation scheme proposed in [2] below, and show that it is not secure in practice under non-perfect synchronization and mismatch of signal amplitude and phase.

3.1 Basic Scheme
The basic scheme with baseband signal representation is illustrated in Fig. 1 assuming perfect synchronization and Manchester encoding. There are four possible cases when Alice and Bob send bits simultaneously:

- Case 1: Alice sends 0, Bob sends 1;
- Case 2: Alice sends 1, Bob sends 0;
- Case 3: both Alice and Bob send 1;
- Case 4: both Alice and Bob send 0.

For case 3, the superposed signal at Eve is the double of 1’s. This does not help because Eve knows that both devices sent a 1. The same thing happens for case 4. Alice and Bob will discard the bits of these two cases.

It gets interesting for case 1 and case 2. Manchester coding has symmetric (flipped) waveform for 0 and 1. The superposed RF signals at Eve are exactly the same under case 1 and case 2; thus Eve cannot distinguish them.
Bob can make use of the bits at either side as the shared bits to establish a key.

Unfortunately, the basic scheme is not secure in practice where there are inevitable inconsistency on time, amplitude, and phase between Alice and Bob’s RF signals due to non identical devices, which gives Eve clues to distinguish case 1 and case 2.

The inconstancy exists in practice mainly due to the following two reasons:

1) Perfect transmission synchronization requires clock synchronization. It is well known that synchronization is a difficult problem for distributed devices [10] and suffers from clock drift and skew.

2) In fact, Alice or Bob does not know exactly what they are transmitting. What Alice or Bob does know is the clean digital representation of the signal in baseband \( r^i(C_a) \) or \( r^i(C_b) \). However, once the signal is converted to analog and up-converted to the carrier frequency and transmitted, the transmitted signal \( x_a(t) \) or \( x_b(t) \) will be distorted by the channel and noise randomly as modeled in Eq. (2). As a result, neither Alice nor Bob can make accurate compensation for inconsistency.

We identify three types of inconsistency between Alice and Bob that Eve can exploit to break the key.

1) Out of synchronization

\[
x_a(t) = x_b(t + \varepsilon)
\]  

(7)

2) Amplitude mismatch

\[
x_a(t) = \alpha x_b(t)
\]  

(8)

3) Phase mismatch

\[
\text{Hilbert}[x_a(t)] = e^{j\theta} \text{Hilbert}[x_b(t)]
\]  

(9)

where \( \text{Hilbert}() \) represents Hilbert transform of a signal to present its analytic representation. NFC has a high bit rate of 106kbps. Since the bit duration is very small, \( \varepsilon, \alpha \) and \( \theta \) can be considered stable during the key generation [14]. Eve can use them to break the key.

### 3.2 Key Compromise Under Out of Synchronization

As illustrated in Fig. 2, we assume Bob is ahead of Alice. If Eve detects a valley in the middle of a bit, she would know that Alice sends 0 and Bob sends 1. If Eve detects a peak, she would know that Alice sends 1 and Bob sends 0.

### 3.3 Key Compromise Under Amplitude Mismatch

As illustrated in Fig. 5, we assume Bob’s signal is stronger than Alice. If Eve detects a higher voltage at the first half bit and lower voltage at the second half bit, she would know that Alice sends 0 and Bob sends 1. If Eve detects a lower voltage at the first half bit and higher voltage at the second half bit, she would know that Alice sends 1 and Bob sends 0.

### 3.4 Key Compromise Under Phase Mismatch

As illustrated in Fig. 6, we assume Bob has postponed phase. If Eve detects a positive phase change in the middle of a bit, she would know that Alice sends 0 and Bob sends 1. If Eve detects a negative phase change, she would know that Alice sends 0 and Bob sends 1.

As a conclusion, the basic scheme is not secure in practice due to inevitable synchronization offset and mismatch of signal amplitude and phase. A more practical key agreement scheme is needed.

**Figure 5: Superposed RF signal’s envelop under non perfect amplitude match**

4. RIWA

In this section, we propose RIWA to prevent an eavesdropper from breaking the key when there are inevitable non-perfect synchronization and mismatch of signal amplitude and phase. The basic idea of RIWA is to add randomness into Alice’s and Bob’s RF signal \( (x_a(t) \) and \( x_b(t) \)) to make the signals received by Eve indistinguishable under case 1 and case 2. The randomness we introduce include 1) time shifting, 2) amplitude scaling, and 3) phase shifting, which are used to tackle out of synchronization, amplitude mismatch, and phase mismatch, respectively.

In the following subsections, we will describe these three types of waveform shaking methods and show their effectiveness.

In order to differentiate the shaken waveform from the original ones, we use \( X_a(t) \) and \( X_b(t) \) to denote shaken RF waveform. We also use \( Y_e(t) \) to replace \( y_e(t) \) in Eq. (5). We further use \( Y_e^{c1}(t) \) and \( Y_e^{c2}(t) \) to denote the superposed signal received by Eve under case 1 and case 2. Denote the set of all the possible \( Y_e^{c1}(t) \) as \( S_{c1} \) and \( Y_e^{c2}(t) \) as \( S_{c2} \). If we can make \( S_{c1} \) and \( S_{c2} \) have a common set,

\[
S_{c1} \cap S_{c2} \neq \emptyset
\]  

(10)

Eve will not be able to distinguish case 1 and case 2 based on the received signal \( z(t) \) if the following condition is satisfied

\[
\exists Y_e^{c1}(t) = Y_e^{c2}(t) = z(t), \quad z(t) \in S_{c1} \cap S_{c2}
\]  

(11)

That is, \( z(t) \) could be an instance of \( Y_e^{c1}(t) \) or an instance of \( Y_e^{c2}(t) \). Therefore, as long as we can ensure (10), the effectiveness and security of RIWA can be ensured.
of Alice and Bob under case 1 has the total length shifted by \( n \). The waveform is shown in Fig. 7. For each bit, the waveform is randomized to hide Alice and Bob under case 1: \( \gamma_0 \) is fixed. For forward shifting, the waveform of previous bit will be extended by \( \gamma_0 \). For backward shifting, the waveform of previous bit will be shortened by cutting off its last part. For backward shifting, the waveform of previous bit will be shortened by cutting off its last part. For backward shifting, the waveform of previous bit will be extended by \( \gamma_0 \). For backward shifting, the waveform of previous bit will be extended by \( \gamma_0 \).

We conclude lemmas 1-4 to get lemma 5. Lemma 5: if the current bits of Alice and Bob are opposite, they are invisible to Eve under the condition that the previous bits of Alice and Bob are the same, as well as the following bits of Alice and Bob are the same. We use \( \gamma_0^3 \) to denote the shifting of Alice at case 1. For a small out of synchronization \( \varepsilon \ll T \), as long as \( \varepsilon_0 > \varepsilon/2 \), we can easily find instances satisfying \( 2\varepsilon = \gamma_0^3 + \gamma_0^5 - \gamma_0^1 - \gamma_0^2 \).

So that

\[
\gamma_0 = \varepsilon + \gamma_0^1 - \gamma_0^5 = -\varepsilon + \gamma_0^2 - \gamma_0^2
\]

\[
(16)
\]

We denote \( I_n \) as the set of instances. Using lemma 1 and (7), we can prove \( I_n \subset \mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset \). So that (10) is proved.

We conclude lemmas 1-4 to get lemma 5. Lemma 5: if the current bits of Alice and Bob are opposite, they are invisible to Eve under the condition that the previous bits of Alice and Bob are the same, as well as the following bits of Alice and Bob are the same.

Fig. 8 and Fig. 9 show the envelop of the superposed RF waveforms under case 1 and case 2, where the previous bits of Alice and Bob are the same. It can be seen that the superposed RF waveforms under two different cases are perfectly matched.
4.1.1 Adding Guard Bits

We must be cautious to design RIWA dealing with out of synchronization, because it is unable to prove the safety of current bits if previous bits of Alice and Bob are different. Fig. 10 shows such situation. It can be seen from the figure that, indeed there might be a small difference at \( Y_c(t) \) between case 1 and case 2 due to the Gibbs phenomenon. A very powerful attacker may use it to crack some of the bits.

To avoid it, we add guard bits. The bits that Alice and Bob send are

\[
\begin{align*}
C_a & = [0, C_a(1), 0, C_a(2), 0 \cdots 0, C_a(N)] \\
C_b & = [0, C_b(1), 0, C_b(2), 0 \cdots 0, C_b(N)]
\end{align*}
\]

(17)

Figure 10: Superposed RF waveform (previous bits of Alice 1 and Bob 0)

where only \( C_a(k) \) and \( C_b(k) \) are random bits. In this way, the previous and following bits of \( C_a(k) \) and \( C_b(k) \) can be guaranteed identical. The key generation rate however reduces by 1/2. Considering 50% probability that a bit is effective, the key generation rate is about 1/4 of bit rate. For 106 kBaud rate, the key generation rate of RIWA is about 20kbps, which is fast enough in practice.

4.1.2 Discarding Bits that Eve may Discover

Another situation should be noticed is when the actual out of synchronization is larger than a threshold:

\[ v > 2\gamma_0 - \varepsilon \]  

(18)

Substituting (16) into (18), it can be seen that case 2 cannot cause such a large out of synchronization \( (\gamma_a^2 - \gamma_b^2 > 2\gamma_0) \). Eve will know that Alice sends 1 and Bob sends 0 (case 1).

A powerful attacker can crack some bits from the above situation. So in RIWA, we can choose to detect the actual out of synchronization. If it is larger than a threshold estimated by (18), we discard it in advance. Due to this discard, the bits left may have more zeros than ones at Alice side. Finally, randomness amplification is optional.

4.2 Random Amplitude Scaling against Amplitude Mismatch

Similar to out of synchronization, for amplitude mismatch, the amplitude of waveform is multiplied by a random coefficient \( \beta \) uniformly between \([1 - \beta_0, 1 + \beta_0] \) at each bit.

The RF waveforms of Alice and Bob under case 1 and case 2 are

\[
\begin{align*}
X_a^{c1}(t) &= \beta_a^{1}\alpha_{a}^{1}(t) \\
X_b^{c2}(t) &= \beta_b^{2}\alpha_{b}^{2}(t) \\
X_a^{c2}(t) &= \beta_a^{1}\alpha_{a}^{2}(t) \\
X_b^{c1}(t) &= \beta_b^{2}\alpha_{b}^{1}(t)
\end{align*}
\]

(19)

We need to get

\[
\begin{align*}
X_a^{c1}(t) &= Y_b^{c2}(t) \\
X_a^{c2}(t) &= x_a^{c2}(t)
\end{align*}
\]

(20)

to make two cases equal, i.e., \( X_a^{c1}(t) + X_b^{c1}(t) = X_a^{c2}(t) + X_b^{c2}(t) \). With the help of (8), (20) can be simplified as

\[
\begin{align*}
\beta_a^{1}\alpha = \beta_b^{2} \\
\beta_a^{2}\alpha = \beta_b^{1}
\end{align*}
\]

(21)

We can easily find instances satisfying (21) as long as \( \beta_0 > (\alpha - 1)/(\alpha + 1) \). We denote \( I_a \) as the set of instances, \( I_a \subset S_{c1} \cap S_{c2} \neq \phi \). Therefore (10) is proved.

We need to pay attention when actual amplitude mismatch is larger than a threshold

\[ \alpha' > \frac{1 + \beta_0}{\alpha(1 - \beta_0)} \]  

(22)

case 2 cannot cause such a big amplitude mismatch. So in RIWA, we can choose to detect the actual amplitude mismatch. If it is larger than a threshold estimated by (22), we discard it in advance.
4.3 Random Phase Shifting against Phase Mismatch

We also add phase shaking $\varphi$ uniformly distributed between $[-\varphi_0, \varphi_0]$ to deal with phase mismatch. In this case, we need to pay attention to the phase change between bits. It could be a clue for Eve to separate border line of consecutive bits. Recall that we already add guard bits between consecutive random bits. It makes things easier to prevent phase change between bits. The shaking method is shown in Fig. 11. We use previous guard bit to make a smooth connection when transmitting a 1. In this way, the border line of the envelop will not be exposed to Eve.

The RF waveforms of Alice and Bob under two cases are

$$\text{Hilbert}(X_a^{c1}(t)) = e^{j\varphi_1} \text{Hilbert}(x_a^{c1}(t))$$
$$\text{Hilbert}(X_b^{c1}(t)) = e^{j\varphi_2} \text{Hilbert}(x_b^{c1}(t))$$
$$\text{Hilbert}(X_a^{c2}(t)) = e^{j\varphi_3} \text{Hilbert}(x_a^{c2}(t))$$
$$\text{Hilbert}(X_b^{c2}(t)) = e^{j\varphi_4} \text{Hilbert}(x_b^{c2}(t))$$

(23)

We need to get

$$X_a^{c1}(t) = X_b^{c2}(t)$$
$$X_b^{c1}(t) = X_a^{c2}(t)$$

(24)

to make two cases equal $X_a^{c1}(t) + X_b^{c1}(t) = X_a^{c2}(t) + X_b^{c2}(t)$.

With the help of (9), (24) can be simplified as

$$\varphi_1 = \varphi_2 - \theta$$
$$\varphi_3 = \varphi_4 - \theta$$

(25)

We can easily find instances satisfying (25) as long as $\varphi_0 > \theta/2$. We denote $I_p$ as the set of instances, $I_p \subset S_{c1} \cap S_{c2} \neq \phi$. Therefore (10) is proved.

Finally, we show the whole steps to generate shaken RF signal in Fig. 12.

4.4 Key Agreement Procedure of RIWA

The key agreement procedure is shown in Fig. 13. After both sides agree for key generation, at Alice side, she sends a request for simultaneous random bit stream sending. Then she begins to send bit stream while listening. The bit stream has two parts. The first part is normal unshaken bit stream. It is aiming for Alice to make a rough synchronization and amplitude, and phase match. The second part is shaken bit stream. At Bob side, when he receives the request of bit sending, he begins to send bit stream. Same as Alice, first-part bits are normal bits, and the following bits are shaken bits.

After finishing bits sending, Alice tells Bob the positions of bits that can be used. Alice picks her bits as a key: Bob picks his bits and reverse them as a key. Example: Suppose Alice sends $C_a[01000101]$, Bob sends $C_a[00010100]$. Alice listens to the signal, so she knows the 2nd, 4th and 8th bits can be used. She picks the corresponding bits [101] and tells Bob the positions are [2, 4, 8]. Bob picks the corresponding bits on his side [010] and reverse them to get [101]. Eve who only sees superposed RF signal, cannot know the 2nd, 4th and 8th bits that Alice sends.

4.5 Discussion

We briefly compare RIWA with the Elliptic Curve Diffie-Hellman key exchange method (ECDH) [15].

4.5.1 Energy Consumption

For energy consumption, RIWA only needs simple computations of bit comparisons. The main energy consumption is bit transmissions. To generate 128 bit key, considering 50% probability that a bit is effective and the discard of guard bits, the number of bits transmitted is about 512. ECDH needs about 1 kbit transmission (both overheads of ECDH and RIWA are not considered), which is a little more than RIWA. More importantly, ECDH has high computational complexity; the energy consumed by computation of the cryptographic operations is at least one order of magnitude larger than transmissions [15].

4.5.2 Time Consumption

For time consumption, to generate 128 bit key, RIWA needs $512\text{bit}/106\text{kbps} = 4.8\text{ms}$ assuming 106 kBaud rate. For ECDH, computations alone cost $9.1\text{~s}$ to $15.1\text{~s}$ [15].

4.5.3 How about MIMO Attacker

For NFC, the operating distance is within 10cm, significantly smaller than half of a wavelength (11m). In this case, it is
well known that MIMO techniques cannot separate their signals. Hence, the eavesdropper cannot exploit MIMO to decode the RFID’s data [13]. We believe our method deals with major threats in NFC, and is generally applicable.

5. SIMULATIONS

In the following, we conduct numerical simulations to investigate our method. We set the baud rate to 106 kBaud. The modulation is Manchester coding with 100% ASK. We add random channel noises at Alice and Bob sides $n_a(t)$, $n_b(t)$ with the SNR of 20dB. We assume antenna efficiency of Alice and Bob are $\eta_a = \eta_b = 1$.

5.1 RIWA’s Performance against Out of Synchronization

In order to analyse RIWA’s performance against out of synchronization, we generate a delay at Alice side, $\varepsilon = 0.1\mu s$. To deal with this delay, Alice and Bob both generate a waveform shaking of $\gamma_0 = 0.5\mu s$, (the amount of shaking is about 1/10 of one bit duration, which is too small to cause bit errors for Alice and Bob). For each simulation, Alice and Bob both send 200 random bits. The chance that one user send 1 and the other send 0 is 50%; so the number of effective bits is approximately 100. Alice and Bob discard ineffective bits and generate the key with effective bits using the method introduced in section 4.1.

At Eve’s side, she uses the envelop of superposed RF waveform to guess the key. The scheme is shown in Fig. 14. She first detects whether the bit is an effective bit. If most of the sample points (70% in the simulation) is within the center part (upper and lower threshold in the simulation are 0.4 and 0.6 respectively). She takes it for an effective bit that one user sends 1, and the other sends 0. She then observe the middle part of the bit. If there is a peak (detect threshold in the simulation is 0.7) he takes it as case 1, where Alice sends 1, Bob sends 0. If there is a valley (detect threshold in the simulation is 0.3), she takes it as case 2, where Alice sends 0, Bob sends 1. If there is no peak or valley detected, she will make a random guess.

We did the simulation 10000 times. The BER histogram at Eve is shown in Fig. 15(left). Eve’s BER is 42% on average, with a standard deviation of 6%. The result closes to a random guess where BER is 50%.

We also investigate the impact of different degrees of out of synchronization on BER. We vary $\varepsilon$ from 0.1$\mu s$ to 1$\mu s$. It makes the ratio of out of synchronization to shaking, $\varepsilon/2\gamma_0$, vary from 10% to 100%. The result is shown in Fig. 15(right). It can be seen that as $\varepsilon$ increases, the number of bits that Eve makes correct guess increases. When $\varepsilon = 2\gamma_0$, $\varepsilon$ is too large that $\gamma$ can not cover it. As a result, Eve can completely determine the bit. Section 4.1 also proposed a method to deal with large $\varepsilon$ by discarding the bits that Eve may discover. However, this will cause a drop on key generation rate. Fig. 16(left) shows the key generation rate
under different degrees of out of synchronizations. As expected, when $\varepsilon$ increases, the key generation rate decreases. For $\varepsilon/2\gamma_0 < 0.3$, the key generation rate is larger than 20kbps. Fig. 16(right) shows the impact of SNR to key generation rate. For SNR $> 16$dB, the key generation rate is larger than 20kbps.

5.2 RIWA’s Performance against Amplitude Mismatch

In this simulation, we generate an amplitude mismatch $\alpha = 1.1$. To deal with it, Alice and Bob both generate an amplitude shaking of $\beta_0 = 0.4$. Manchester coding has half high voltage and half low voltage (0 for 100% ASK). At Eve’s side, she will see half bit of Alice’s high voltage and half bit of Bob’s high voltage. She takes the higher one as Alice’s high voltage and the other as Bob’s high voltage.

We run the simulation for 10000 times. The BER histogram at Eve is shown in Fig. 17(left). Eve’s BER is 44% on average, with a standard deviation of 6%. The result closes to a random guess. As the degree of amplitude mismatch increases, the number of bits that Eve makes correct guess increases. In this case, we discard the bits that Eve may discover. As a result, the key generation rate decreases. Fig. 17(right) shows the key generation rate under different degrees of amplitude mismatch. For $\alpha < 1.2$, the key generation rate is larger than 15kbps.

5.3 RIWA’s Performance against Phase Mismatch

In this simulation, we generate a phase mismatch at Bob’s side $\theta = 1^\circ$, where Alice’s initial phase is $0^\circ$, and Bob’s initial phase is $1^\circ$. To deal with it, Alice and Bob both generate a phase shaking with $\varphi_0 = 5^\circ$. Eve will see two phases for one bit duration. She chooses the RF waveform with a larger initial phase as Bob’s RF waveform.

We run the simulation 10000 times. The BER histogram at Eve is shown in Fig. 18(left). Eve’s BER is 48% on average, with a standard deviation of 5%. The result closes to a random guess. As the degree of phase mismatch increases, the number of bits that Eve makes correct guess increases. In this case, we discard the bits that Eve may discover. As a result, the key generation rate decreases. Fig. 18(right) shows the Key generation rate under different degrees of phase mismatch. For $\theta < 3^\circ$, the key generation rate is larger than 20kbps.

6. EXPERIMENTS

We conduct experiments to prove the concept of RIWA. We use software defined radios, USRP n210 to simulate the channels of NFC devices. We use BasicTX and BasicRX Daughter board to send RF signals at 13.56MHz. We set the baud rate to 106 kBaud. The modulation is Manchester coding with 100% ASK. The sample rate is set to 2MHz, that is 20 sample points a bit. The antennas are the DLP-RFID-ANT antennas. We use a power splitter to implement full-duplex on single antenna. The trunk port is connected to antenna, and two branch ports are connected to TX and RX respectively.

In the experiment, we aim at showing the effectiveness of RIWA dealing with the existence of out of synchronization. The experiment set up is shown in Fig. 19. Eve’s antenna is in the middle of Alice and Bob’s, so she can hear clearly on the communication between Alice and Bob. For simplicity, we set the bit streams that Alice and Bob send as $[0, 1, 0, 1 \cdots]$ and $[1, 0, 1, 0 \cdots]$, respectively. According to the analysis in section 5.1, we add guard bits to the bit streams.

$$C_a = [0, 0, 0, 0, 1, 0, 0, 1, 0 \cdots]$$
$$C_b = [0, 1, 0, 0, 0, 1, 0, 0, 0 \cdots]$$

Fig. 20 shows the RF envelop that Alice receives while Bob is sending. Blue line shows the waveform for normal bits;
Figure 20: Normal waveform and shaken waveform

Figure 21: Superposed RF envelop at Alice’s side

red line shows the waveform for shaken bits. Fig. 21 shows the RF envelop that Alice receives, while Alice and Bob are simultaneous sending (without synchronization). It can be seen that, the waveform is approximately the superposed waveform of Alice and Bob.

We then make a rough synchronization for Alice and Bob (there exists a small out of synchronization of about $0.5\mu s$). Fig. 22 shows the received RF envelop at Eve. For convenience, we temporarily do not take amplitude mismatch into consideration. It can be seen from the figure that, due to out of synchronization, Eve can separate Alice from Bob. By assuming that Alice is ahead of Bob (opposite assumption leads to a mirror symmetric result). Eve can tell that when there is a peak in the middle of RF envelop, it is the case that Alice sends 0 and Bob sends 1. When there is a valley in the middle of RF envelop, it is the case that Alice sends 1 and Bob sends 0.

To cover up the out of synchronization, we add a random shaking ($-1\mu s$ to $1\mu s$ in the experiment) on both Alice and Bob. Fig. 23 shows a snapshot of the received RF envelop at Eve. It can be seen from the figure that, due to shaking, a peak is turned into a valley. If Eve still uses the above method to separate Alice and Bob, she will make a mistake.

Figure 22: Superposed RF envelop at Eve with the existence of out of synchronization

Figure 23: Superposed RF envelop at Eve after shaking

Curves could be applied to establish symmetric keys. Many works [19, 20] focus on finding the least expensive protocol for NFC and RFID. However, standard key agreement protocols intrinsically have extensive computational overhead and energy consumption. They are not preferable for resource constrained NFC devices.

The works in [2, 16] proposed specific key agreements. The idea is for both devices to send random bits at exactly the same time. For the case when two parties send different bits, the attacker will not be able to identify which device sent the 0 and which device sent the 1. However, this concept is difficult to be implemented in practice due to the strict requirement of perfect synchronization as well as amplitude and phase match. This issue has been discussed in detail in this paper.

The works in [17, 18] proposed Physical layer key generation methods by using the wireless channel measurements. By exploiting the reciprocal property of the wireless fading channel between two NFC devices, shared secret key can be generated. The methods have low computational overhead compared with standard key agreement protocols. However, the key generation rate is low. The time interval between two effective measurements of channel randomness is bounded by coherence time. Moreover, the methods have multiple steps including advantage distillation, information reconciliation and privacy amplification, which add computation and transmission overhead and time consumption.

RIWA overcomes major problems in NFC key generation. It is a practical, fast, energy-efficient, and secure key agreement scheme for resource constrained NFC devices.

7. RELA TED WORK

Existing works [1, 2, 16, 17, 18] indicate that using symmetric key cryptography to secure the NFC is a practical solution to protect against many kinds of attacks. It is mentioned in [1] that due to the inherent protection of NFC against Man-in-the-Middle-Attacks, standard key agreement protocols like Diffie-Hellman based on RSA or Elliptic

Figure 20: Normal waveform and shaken waveform

Figure 21: Superposed RF envelop at Alice’s side

red line shows the waveform for shaken bits. Fig. 21 shows the RF envelop that Alice receives, while Alice and Bob are simultaneous sending (without synchronization). It can be seen that, the waveform is approximately the superposed waveform of Alice and Bob.

We then make a rough synchronization for Alice and Bob (there exists a small out of synchronization of about $0.5\mu s$). Fig. 22 shows the received RF envelop at Eve. For convenience, we temporarily do not take amplitude mismatch into consideration. It can be seen from the figure that, due to out of synchronization, Eve can separate Alice from Bob. By assuming that Alice is ahead of Bob (opposite assumption leads to a mirror symmetric result). Eve can tell that when there is a peak in the middle of RF envelop, it is the case that Alice sends 0 and Bob sends 1. When there is a valley in the middle of RF envelop, it is the case that Alice sends 1 and Bob sends 0.

To cover up the out of synchronization, we add a random shaking ($-1\mu s$ to $1\mu s$ in the experiment) on both Alice and Bob. Fig. 23 shows a snapshot of the received RF envelop at Eve. It can be seen from the figure that, due to shaking, a peak is turned into a valley. If Eve still uses the above method to separate Alice and Bob, she will make a mistake.

Figure 22: Superposed RF envelop at Eve with the existence of out of synchronization

Figure 23: Superposed RF envelop at Eve after shaking

Curves could be applied to establish symmetric keys. Many works [19, 20] focus on finding the least expensive protocol for NFC and RFID. However, standard key agreement protocols intrinsically have extensive computational overhead and energy consumption. They are not preferable for resource constrained NFC devices.

The works in [2, 16] proposed specific key agreements. The idea is for both devices to send random bits at exactly the same time. For the case when two parties send different bits, the attacker will not be able to identify which device sent the 0 and which device sent the 1. However, this concept is difficult to be implemented in practice due to the strict requirement of perfect synchronization as well as amplitude and phase match. This issue has been discussed in detail in this paper.

The works in [17, 18] proposed Physical layer key generation methods by using the wireless channel measurements. By exploiting the reciprocal property of the wireless fading channel between two NFC devices, shared secret key can be generated. The methods have low computational overhead compared with standard key agreement protocols. However, the key generation rate is low. The time interval between two effective measurements of channel randomness is bounded by coherence time. Moreover, the methods have multiple steps including advantage distillation, information reconciliation and privacy amplification, which add computation and transmission overhead and time consumption.

RIWA overcomes major problems in NFC key generation. It is a practical, fast, energy-efficient, and secure key agreement scheme for resource constrained NFC devices.

8. CONCLUSION

We propose a practical and energy efficient key agreement method for duplex NFC. In the proposed method RIWA, two users, Alice and Bob simultaneously send random bits of 0 or 1. There are probabilities one of them sends 0 and the other sends 1. An attacker only sees the superposed RF signal cannot figure out which device sent the 0 and which device sent the 1. Alice and Bob themselves know what they have sent and can make use of the advantage to establish a secret key. RIWA also considers various inconsistency which attackers may use to break the method: out of synchronization, amplitude mismatch and phase mismatch. To deal with inconsistency, RIWA introduces random shaking on RF waveform and guard bits between consecutive random bits. The shaking acts like random noise to cover up the inconsistency. The guard bits prevent the attacker determining the bit from the previous one.
Numerical simulations show that, with the existence of out of synchronization, Eve’s BER is 42% on average with $\varepsilon/2\gamma_0 = 10\%$. With the existence of amplitude mismatch, Eve’s BER is 44% on average with $\alpha = 1.1$, $\beta_0 = 0.4$. With the existence of phase mismatch, Eve’s BER is 48% on average with $\theta = 1^\circ$, $\phi_0 = 5^\circ$. Since BERs are close to random guesses, the result confirms that Eve gains nearly no information about the bits. Concept-proof experiments based on software defined radio demonstrate that the shaking added covers up the out of synchronization between Alice and Bob. Eve, trying to use out of synchronization to crack the key, will make mistakes. RIWA is a simple and convenient security method with low energy and time consumption applicable to low cost, resource-constrained NFC devices.

References