Secured Bilateral Rendezvous using Self Interference Cancellation in Wireless Networks

Eun-Kyu Lee and Mario Gerla
Computer Science Department, UCLA
Los Angeles, CA, USA
{eklee, gerla}@cs.ucla.edu

***Abstract*—Secure transmission among mobile nodes in hostile environments has often been achieved by trading off security with communication latency. However, in time critical applications (e.g., tactical wireless networks, civilian law enforcement operations, homeland security missions, etc.), latency is more critical than any other parameters. Long latency is generally attributed to the time required by authentication in the presence of eavesdroppers and jammers. To address the latency issue, this paper proposes a Secured Bilateral Rendezvous (SBR) protocol that reduces the communication latency while guaranteeing protection. SBR performs neighbor detection before authentication. During detection, nodes exchange a common secret key in plaintext that helps speed up the authentication process. The plaintext message is protected against jammers by allowing nodes to transmit simultaneously and to recover the partner's packets using self interference cancellation technique. Simulation results show that the proposed SBR protocol can effectively withstand various jamming attacks.**

***Index Terms*—wireless network; jamming attack; interference cancellation; rendezvous**

I. Introduction

Data protection has been a critical issue in MANETs exposed to attacks such as vehicular networks engaged in civilian defense operation and; military networks. The shared property of the wireless medium makes communications more vulnerable to external threats. To achieve secure transmission, a network often sacrifices communication efficiency. Say, in a jamming attack scenario, two nodes must communicate in the presence of adversaries. Without prior knowledge, the two nodes exercise a random Spread Spectrum (SS) modulation (either Frequency Hopping, FH, or Direct Sequence, DS) to authenticate each other [11, 13, 20]. Then, they activate another SS modulation for data transmission in which they use the common secure key that was exchanged and agreed upon during authentication.

However, with the random SS scheme (say FH), they can exchange packets successfully and thus authenticate only when they encounter by chance. This random rendezvous aggravates the latency performance as nodes are required to exchange rather large authentication messages. Capar *et al.* addressed this problem by proposing a Physical-Layer-Enhanced Key Exchange (PEK) scheme [2]. In PEK, a sender transmits a common key in plaintext. Randomness in the physical layer eventually creates a situation where the receiver gets the packet but the jammer does not. When that occurs, the two nodes exchange authentication messages via FH modulation using the key. This way, PEK achieves fast authentication, which reduces the overall communication latency. However, nodes still rely on the randomness of wireless channel condition, and the unilateral scheme is vulnerable to reactive attacks like eavesdropping.

This paper proposes a *Secured Bilateral Rendezvous (SBR)* protocol that reduces the communication latency while maintaining data security. Like PEK, the nodes send out the common secret key in plaintext, thus improving latency. As a difference, however, SBR allows two nodes to transmit their keys at the same time, namely bilateral transmission. Since their signals interfere in the air, jammers in the radio range misinterpret the mixed signal as a noise. To enable each node to recover packets of interest from the interfered signal, SBR exploits self-interference cancellation (SIC) [16]. Once two keys are recovered, two nodes generate a new key from those keys, which is used for authentication. The possibility of using SIC is motivated by the observation that a node in a tactical network is large enough that it could install two antennas with separation and accomplish a full-duplex wireless communication with help from the SIC technique. In addition to security purpose, the feasibility of full-duplex introduces additional advantages. In VANET, say, it resolves the hidden terminal problems as well as helps mitigate the broadcast storm problem due to periodic control packets. This paper examines the feasibility through simulations and shows that the proposed SBR protocol achieves secure and efficient communications in the presence of various jamming attacks.

Contribution of this paper is three folds. First, we develop an efficient anti-jamming communication protocol that minimizes communication latency while it does not sacrifice security. Second, the paper applies SIC to a MANET for the first time (to the best of authors' knowledge) and investigates the feasibility of full-duplex communication. Last, we inspect jammer's behaviors in detail and evaluate the proposed SBR protocol against them by showing fine-grained performance results.

The rest of the paper is organized as follows. In Section II, we review jamming attacks and show our motivation. Section III investigates the feasibility of self-interference cancellation. Section IV describes the proposed Secured Bilateral Rendezvous protocol, SBR. Section V analyzes jamming attack models and evaluates the proposed scheme along various attack patterns. Finally, we conclude the paper in Section VI.

978-1-4577-0899-2/11/$26.00 ©2011 IEEE

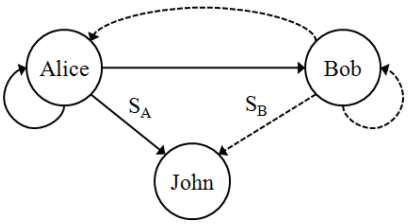

Fig. 1. Two nodes, A and B, transmit and receive packets at the same time.

II. Anti-Jamming Communication

A. Jamming Attack

A jamming attack is a malicious behavior disturbing legitimate communication in a wireless network. Unlike conventional security threats, a jammer disrupts wireless communications by simply injecting false messages or noise signals into the wireless medium. The shared nature of the wireless medium empowers the jammer to disable all data transmissions within the radio range. Gummadi *et al.* [8] investigated its impact on 802.11 networks and discovered that the 802.11 network is vulnerable even to weak interference. The jamming attack is broadly categorized into three models: *eavesdropping*, *noising*, and *inserting* [22]. First, a jammer eavesdrops legitimate communication to obtain valuable information. The jammer can launch a replay attack by using the obtained data. Second, a jammer emits noise signals to saturate the wireless channel. Legitimate nodes cannot even commence packet transmission. A proactive jammer continuously transmits noise signals whereas a reactive jammer listens to the channel and begins transmitting noise only when detecting signals. Third, a jammer inserts false messages. For instance, the jammer impersonates a legitimate node and tries to establish a wireless communication with another node.

To protect nodes against above attacks, an authentication process is preferred before data transmission. Over the years, various approaches have leveraged randomized SS modulation to enable two nodes having no prior knowledge to authenticate each other [18, 19, 20]. In Uncoordinated Frequency Hopping (UFH) [20], a sender and a receiver randomly switch over multiple frequencies[1]. Upon meeting on the same frequency by pure chance, they exchange authentication messages. Thus, the latency performance relies on the probability of encounters. However, UFH takes considerable time: it takes almost $40s$ to exchange two authentication messages when the jammer's attack probability is 60%. The slow operation is primarily attributed to the fact that a successful data transmission depends on accidental rendezvous in the random FHSS.

B. Sending a common key in plaintext

In addition to the accidental rendezvous latency problem, the large size of a typical authentication message affects latency adversely. Note that to prevent the jammer from synchronizing on the frequency and detecting legitimate signals, the slot duration, during which a node dwells on one frequency in FH, must be short. Thus, the authentication message must be generally fragmented into a set of small packets each fitting into one slot. Increasing number of fragments increases the communication latency. Our approach is that a node sends a common SS spreading key in plaintext that is small enough to be delivered in one packet (i.e., one slot). Once the other nodes receive it, they initiate a SS demodulation using the common key, which converges faster than the randomized FH scheme.

The primary concern of transmitting the plaintext is how to protect the message against jammers placed between A and B. To solve the problem, we propose to conceal the packets in their own interference. In Fig. 1, when two nodes, Alice (A) and Bob (B), are within radio reach, they transmit their short packets in an open channel *at the same time*. Intuitively, this seems to be very counterproductive because two radio signals collide on the channel and the mingled signal is received as a noise. For instance, both Bob (B) and John (J), an adversary, receive the interfered signal, $S_R = S_A + S_B$, and handle it as an erroneous packet. We assume here that each node has two separate antennas, one to transmit and the other to receive. On the receive antenna, B is able to subtract its own signal S_B from S_R, and thus can extract A's packet. This is possible since B has stored a copy of its own signal S_B whereas J does not know it. So message from A is protected from adversary Js eavesdropping.

Another concern of protecting the plaintext against jammers is to hide the common key information. A jammer behind A and further removed from B can only hear S_A without interference from B. Thus, it can eavesdrop A's key and launch an attack. To solve the problem, we propose that nodes A and B generate a new key based on two plaintext messages from both A and B. In other words, they both send own keys from which a new one is created. As such, the jammer behind A cannot recover the new key because it does not receive S_B from B. The following sections describe our proposed scheme that addresses the above two concerns.

III. Self Interference Cancellation

To conceal information in the interference, we exploit self-interference cancellation technique. This section examines its feasibility in a tactical MANET. To this end, we begin discussing wireless communication fundamentals, and then describe cross correlation and antenna configuration to subtract self signal out.

A. Wireless Communication Fundamentals

In wireless communications, a signal is represented as a sequence of discrete complex symbols each of which are mapped from the bits of a packet via modulation process. A transmitter generates a symbol in a fixed interval T [seconds]. We denote x[n] as n^{th} transmitted complex symbol from the transmitter. The received signal is also represented as a series of complex symbols after sampled with interval T at a receiver. The corresponding n^{th} symbol received is represented as y[n]. Then, their relationship is approximated as:

$$\mathrm{y}[n] = \mathbf{H}\mathrm{x}[n] + \mathrm{I}[n] + \mathrm{w}[n] \quad (1)$$

[1]We use "channel" and "frequency" interchangeably

where **H** is the channel coefficient, I[n] represents the combined interfering signal, and w[n] is a noise. **H**=$\alpha e^{-j\beta}$, a complex number, represents the channel attenuation (α) and the phase shift (β). If the node could recover x[n] from y[n], then data is successfully transmitted.

In communications theory, the standard Signal-to-Interference-plus-Noise-Ratio (SINR) model is used to determine the ability of successful data recovery. The SINR value is computed as the ratio of the received signal power $P(X)$ to the combined power of interference and noise $P(I) + P(W)$ at the receiver.

$$SINR = \frac{P(X)}{P(I) + P(W)} \geq \tau \tag{2}$$

The value is used to compute the bit error rate (BER): a large SINR implies a stronger signal and thus few bit errors. Then, the BER is used to calculate the packet error rate (PER) that defines the average fraction of transmitted packets that are not detected correctly. The receiver can decode the transmitted packet if the SINR value is above a given threshold τ. The noise term $P(\mathrm{w})$ in (2) is Gaussian in nature and is approximated to about -100 dBm in 22MHz 802.11b[2] or 20MHz 802.11g. To achieve a PER of 1% [3] in 1Mbps rate, a signal-to-interference ratio (τ) of at least 10 dB above the noise threshold is required. When taking into account processing gain (e.g., Barker coding in 802.11b), τ theoretically goes down to 0 dB.

Interference cancellation makes use of the fact that interfering signals, unlike noise, have structure determined by data that they carry [9]. Its objective is to mitigate the harmful effects of interference, removing the known signal structure from the interfered signal and improving the effective SINR of the signal of interest. To search for the structure in the incoming signal, cross correlation is performed, a popular technique to detect known signal patterns in wireless networks [3]. When the pattern is present in the incoming signal, their cross correlation would yield a spike, a high correlation value. Recent researches have applied the correlation technique to detect collision by verifying the presence or absence of predefined preambles in the incoming signal [7, 17].

B. Impact of Self Signal

Similar to interference cancellation, a node can employ self-interference cancellation (SIC) that detects and eliminates self signal (interference). To this end, the node is equipped with two antennas: one for transmitting and the other for receiving. In Fig. 1, B's RX antenna receives S_A from A as well as self signal S_B from own TX antenna. We define two those signals as $\mathrm{y}_A[n]=\mathrm{H}_A\ \mathrm{x}_A[n]$ and $\mathrm{y}_B[n]=\mathrm{H}_B\ \mathrm{x}_B[n]$, respectively. When ignoring additional interference, the received signal y[n] at B is written as (3). By applying (3) to (2), we obtain (4), our SINR model.

$$\mathrm{y}[n] = \mathrm{y}_A[n] + \mathrm{y}_B[n] + \mathrm{w}[n] \tag{3}$$

$$SINR(\mathrm{y}) = \frac{P(\mathrm{y}_A)}{P(\mathrm{y}_B) + P(\mathrm{w})} \tag{4}$$

If the $SINR(\mathrm{y})$ value exceeds τ, then the node B could extract $\mathrm{y}_A[n]$ from y[n] and recover A' packet. In general, the self signal is stronger than A's signal due to the proximity of the TX and RX antennas at B. Thus, $SINR(\mathrm{y}) < \tau$. However, as B knows the self signal, $\mathrm{x}_B[n]$, it can subtract it from y[n]. This eliminates its power contribution to the denominator. Thus, if (5) is satisfied, B can successfully decode $\mathrm{y}_A[n]$. To ensure successful packet reception, $P(\mathrm{y}_B - \mathrm{x}_B) \simeq -100$ dBm is desired.

$$SINR(\mathrm{y} - \mathrm{x}_B) = \frac{P(\mathrm{y}_A)}{P(\mathrm{y}_B - \mathrm{x}_B) + P(\mathrm{w})} \geq \tau \tag{5}$$

To achieve above SIC, there remain two major issues. First, as noted in (1), a wireless signal distorts over the wireless channel; so $\mathrm{y}_B[n] \neq \mathrm{x}_B[n]$. This could yield signal of interest incorrectly. Second, even though $\mathrm{y}_B[n]$ is completely reconstructed, cross correlation technique works properly, only when $P(\mathrm{y}_B) - P(\mathrm{y}_A) \leq \tau_{th}$, where τ_{th} is a threshold value. The next section discusses those issues with two SIC techniques.

C. Self Interference Cancellation

There are two categories of SIC techniques applied to our scheme: cross correlation and antenna configuration.

1) Cross correlation: To subtract the self signal $\mathrm{y}_B[n]$ from y[n] in (3), the node B performs cross correlation. Given a known signal s[n] whose length is l, cross correlation $R(p)$ measures similarity of two signals, s[n] and y[n], as a function of a time-lag; it is a sliding dot product and represented as:

$$\begin{aligned} R(p) &= \sum_{i=0}^{l} \mathrm{s}^*[i]\mathrm{y}[i{+}p] \\ &= \sum_{i=0}^{l} \mathrm{s}^*[i](\mathrm{y}_A[i{+}p] + \mathrm{y}_B[i{+}p] + \mathrm{w}[i{+}p]) \\ &= \sum_{i=0}^{l} \mathrm{s}^*[i]\mathrm{H}_B\mathrm{x}_B[i{+}p] \end{aligned} \tag{6}$$

where p is a lag at a sample-level, and $\mathrm{s}^*[i]$ is the complex conjugate of s[i], an approximation of $\mathrm{y}_B[n]$. Since the pattern of signal s[n] is independent of $\mathrm{y}_A[n]$ and w[n], their terms are close to zero; thus, we obtain (6). If s[n] is same to $\mathrm{y}_B[n]$, then the correlation coefficient $R(p)$ shows a spike, and B can subtract it from y[n]. Since $\mathrm{y}_B[n]$ is the result of several effects such as filter effect, frequency offset, sampling effect, and channel distortion. s[n] must be recreated from $\mathrm{x}_B[n]$ so that $\mathrm{s}[n] \simeq \mathrm{y}_B[n]$. Let $\mathrm{s}[n] = \mathrm{H}_W\mathrm{x}_B[n]$, where H_W is a filter for the self signal over the wire. The goal for accurate cross correlation is to develop a filter H_W such that $\mathrm{H}_W = \mathrm{H}_B$. Because both $\mathrm{y}_B[n]$ and s[n] are generated at the same device, they show the same filter distortion and frequency

[2]The maximum transmit power is limited to 20 dBm (100 mW).

[3]A PER of 1% corresponds to a BER of 10^{-6} [8]. 802.11b requires less than a PER of 8% and assumes that the network is impaired if the PER is over 20% [16].

offset effects. To compensate the sampling effect, B uses $x_B[n]$ to compute its sampling offset and aligns it with $y_B[n]$. To capture the effects of channel distortion and multipath, filter taps can be calculated from the clear portion of $y_B[\cdot]$ such that $H_W - H_B \simeq 0$. We can additionally use a pilot sequence to take care of channel dynamics over time.

Cross correlation can be performed on both digital and analog signals. First, cross correlation runs on digital symbols after sampled through an Analog-to-Digital Converter (ADC), so named digital cancellation [7, 9, 17]. Its performance relies on the accuracy of the digital samples, which is again affected by the resolution of the ADC. In an 8-bit ADC, if $P(y_B) > P(y_A) + 40$ dB, then $y_A[n]$ is represented only as a 1-bit resolution. This indicates that τ_{th} is at most 40 dB theoretically. Sen *et al*. experimentally showed that digital cancellation reduced $P(y_B)$ by 18 dB with τ_{th} of be up to 34 dB [17]. Halperin *et al*. similarly discovered 20 dB power reduction of the self signal with cross correlation in [9]. Second, an analog cancellation has been designed and evaluated by using QHx220 interference cancellation chip [1]. The QHx220 takes two inputs; a wireless signal y[*n*] and the self signal s[*n*] transmitted through a wire from the TX antenna. The canceller subtracts s[*n*] from y[*n*] before sending the signal through the ADC stage. The design of the noise canceller is described in [16], which showed power reduction of the self signal by 30 dB. Following researches also successfully canceled around 25 dB of power with analog cancellation [5, 15]. In summary, cross correlation can achieve power reduction of ∼50 dB on the self signal in total, and node B can decode A's message even when the self signal is ∼34 dB stronger than A's signal.

2) Antenna configuration: Our goal is to reduce $P(y_B)$ to accomplish $SINR(y) \geq \tau$ in (4). In addition to cross correlation, the power can be reduced by adjusting antennas physically. Previous studies have considered antenna cancellation [5], nulling antenna [15], and antenna orientation [17]. In this work, antenna separation is examined primarily because nodes in SBR are assumed to be large enough; a vehicle in VANET or a communication entity in military environment. To estimate the effect of antenna separation, we examine power attenuation on radio propagation.

In a conventional path loss model, the power attenuation is proportional to the square of the distance between the TX and RX antennas, and also proportional to the square of the frequency of the radio signal. Mathematically, such a simplified free-space path loss is represented by Friis transmission formula; $P_L = 20\log_{10}(\frac{4\pi d}{\lambda})$, where P_L is power loss in dB, d is the TX-RX distance, and λ is the wavelength. Assuming that the transmit power is 10 dBm and analog cancellation cancels 30 dB of power, Fig. 2 plots the power of the self signal received at own RX antenna $P(y_B)$ with varying distances. As explained, digital cancellation works properly when $\tau_{th} \leq 34$ dB. Suppose the worst case where the received power of A's signal $P(y_A)$ is same to the RX sensitivity, -85 dB on average. This implies that the self signal must be lower than -51 dB. However, this does not guarantee a successful packet

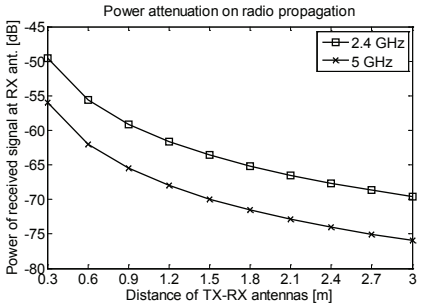

Fig. 2. Power of self signal received at the RX antenna [dB] as a function of the TX-RX antenna distance [m]. Given the transmit power of 10 dBm, power attenuation on propagation is computed from Friis transmission equation, and the signal power of 30 dB is reduced by analog cancellation.

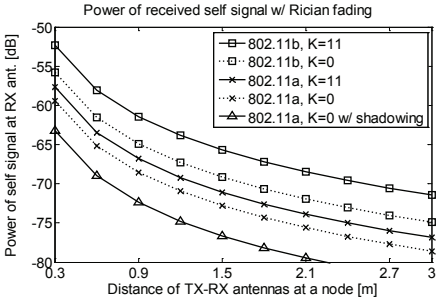

Fig. 3. Fading also affects power attenuation on the radio. Rician fading model is considered, and power of self signal received at the RX antenna [dB] is measured using QualNet simulator. The transmit power is set to 10 dBm, and the power reduction by analog cancellation is also included.

reception. To achieve a PER of 1%, i.e., $SINR(y) \geq 0$ dB in (4), the self signal must be lower than -65 dB so that digital cancellation reduces additional 20 dB of power. Fig. 2 shows that this can be achieved when two antennas are separated by $0.9m$ and $1.8m$ at $5GHz$ and $2.4GHz$ band, respectively.

To look into the fading effect, we consider the Rician fading model with two values of K factor, i.e., K=0 and K=11. Using QualNet simulator, we measure $P(y_B)$ with IEEE 802.11b and 802.11a specification. The transmit power is set to 10 dBm, and the lowest data rates were used. As shown in Fig. 3, when two antennas are apart by $1.2m$, the power of the self signal is below -65 dB, our threshold. With 802.11a, $0.8m$ of separation can achieve the same level of power reduction. If shadowing should be taken into account, the distance can be more decreased (see the curve with triangle mark). Considering that two antennas are installed on a vehicle, these separations are reasonable. Moreover, with other antenna configurtion options, the SIC strategy in MANET can be achieved.

IV. Secured Bilateral Rendezvous

This section proposes *Secured Bilateral Rendezvous (SBR)* protocol. SBR defines three steps for protected rendezvous and secure communication: neighbor detection, authentication, and data transmission. SBR leverages SIC for neighbor detection and adopts SS modulation additionally for both authentication and data transmission. We describe SBR with FHSS examples

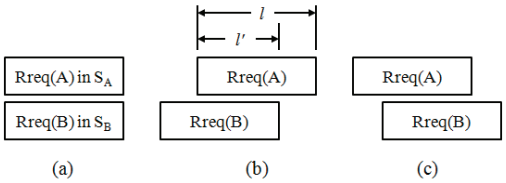

Fig. 4. A node receives two signals, S_A and S_B. (b) represents that the node receives S_B and then S_A; thus, l parts of the signals are overlapped (i.e., interfered).)

$Seq(A)$	3	7	1	4	6
$Seq(B)$	7	8	5	3	6
$\overline{Seq}$	0	5	6	7	2
$\overline{NSeq}$	0	5	7	8	5

Fig. 5. Hopping sequences used in FHSS for authentication. One box denotes one time slot during which a node sits on a frequency that is specified by the index number inside the box. (n=10)

given that a set of n sub frequencies (i.e., sub-channels) in the spectrum is known.

A. Protected Rendezvous

In a mobile scenario, each node proactively looks for other nodes to connect to. To detect neighbors, a node periodically broadcasts Rendezvous request packet ($Rreq$) on a well known, common channel. Say, its period is λ. $Rreq(A)$ is a small control packet that contains a random frequency hopping sequence, $Seq(A) = \{f_1, f_2, ..., f_j\}$ selected by node A, where f_i denotes a frequency index. Note that $Seq(\cdot)$ is not encrypted, so any node can obtain it once receiving $Rreq$. Suppose node A and B are within radio reach (see Fig. 1), and they periodically send $Rreq(\cdot)$ to each other. Fig. 4 shows three cases of signal reception at B according to time alignment amongst nodes. In the case of not interference (i.e., the transmission are not overlapped), the procedure fails, and A and B repeat neighbor detection again. Only when it is interfered, B performs SIC on $Rreq(A)$+$Rreq(B)$ and obtains both $Seq(A)$ and own sequence $Seq(B) = \{f'_1, f'_2, ..., f'_j\}$. Then, it constructs a new frequency hopping sequence, $\overline{Seq}$, via modulo arithmetic which will be used for authentication later.

$$\overline{Seq} = \{\overline{f}_1, \overline{f}_2, ..., \overline{f}_j \mid \overline{f}_i = (f_i + f'_i)\ mod\ n,\ 1 \le i \le j\} \quad (7)$$

In the same way, node A also computes the same $\overline{Seq}$. After sharing three sequences, A and B begin FHSS modulation for authentication. For FHSS, we assume that time is divided into slots, and a node sits on a frequency band during one slot duration (say δ). By default, they hop over the agreed sequence, $\overline{Seq}$, together. Fig. 5 shows an example where both A and B switch over the hopping sequence of $\{0, 5, 6, 7, 2\}$ during which they exchange authentication messages, say M_A and M_B. For extra robustness, in SBR, instead of following the shared $\overline{Seq}$, one of the nodes is allowed to update it dynamically. This is performed at the first slot of $\overline{Seq}$. For instance, B may request to replace the last three indexes in $\overline{Seq}$ with those in $Seq(B)$. Then, as shown in Fig. 5, the updated sequence, $\overline{NSeq}$ becomes $\{0, 5, 7, 8, 5\}$. The property of dynamic update makes authentication process more secure. For authentication, Elliptic Curve Diffie-Hellman (DH) protocol [21] can be considered. The DH protocol allows two nodes to establish another shared secret. The new secret key is, then, used for the FHSS modulation during the data phase. The rest of this section evaluates the security and the overhead of the proposed rendezvous strategy.

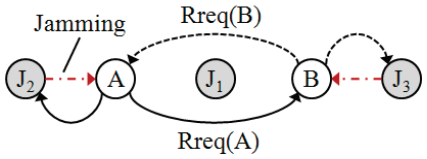

Fig. 6. A network configuration consists of two legitimate nodes, A and B, and one jammer (J) that places on the position of either J1, J2, or J3. J3 is assumed to listen to B, but not A. The solid, dotted, and dash-dotted line represent $Rreq(A)$, $Rreq(B)$, and jamming signal by J, respectively.

B. Security Analysis

This subsection verifies that the proposed SBR scheme is secure against three jamming attack models. Fig. 6 depicts the network configuration used in our analysis. For simple exposition, there is one jammer, J, that launches three kinds of jamming attacks: eavesdropping, noising, and inserting. J is assumed to be in positions J1, J2, or J3 as shown in Fig. 6. The length of $Rreq$ is fixed as l. We assume that nodes use a publicly known coding and modulation schemes. So, the jammer can decode uncorrupted information when it receives a partially collided packet and can identify the start or end of the packet.

First, SBR is robust against the eavesdropping attack. One observes that J must obtain both $Seq(A)$ and $Seq(B)$ to construct $\overline{Seq}$ and to launch its attack during authentication. When J is on J2 or J3, it can only hear either $Rreq(A)$ or $Rreq(B)$, not both. Thus, it cannot construct $\overline{Seq}$. J1, on the other hand, hears both packets whose reception is represented in Fig. 4. We denote the packet overlap as l' (see Fig. 4(b)). Then, the level of hiding is given by the ratio of l' to the packet length, $\rho = l'/l$, $0 \le \rho \le 1$. $\rho = 0$ represents the case of zero interference. J1 perfectly receives both $Rreq(A)$ and $Rreq(B)$. However, in this case, A and B repeat the neighbor detection procedure. In fact, since $Rreq(\cdot)$ is periodically broadcasted, they can now perfectly synchronize at the next rendezvous and send their $Rreq$s with total overlap. $\rho = 1$ denotes total interference. Because two signals interfere completely, J1 decodes neither of them. When $0.5 \le \rho < 1$, the first $l-l'$ part of $Seq(B)$ and the last $l-l'$ part of $Seq(A)$ are exposed to J1. However, their counterparts to recover $\overline{Seq}$ are hidden by interference; thus, J1 cannot recover $\overline{Seq}$. When $\rho_{th} \le \rho < 0.5$, where ρ_{th} is a threshold value, the last l' part of $Seq(B)$ and the first l' part of $Seq(A)$ are hidden in interference. Thus, J1 can decode the $(l-2l')/l$ part in the middle of $\overline{Seq}$. Note that the first l part of $\overline{Seq}$ is still protected.

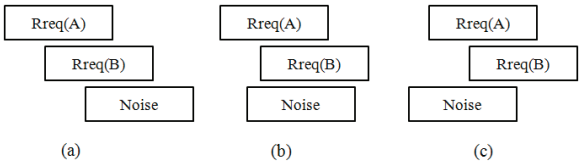

Fig. 7. B' RX antenna receives two $Rreq$ packets, and J3 disturbs the reception with its noise signal.

So, A and B are able to update the sequence to $\overline{NSeq}$ while J1 cannot. In the case of $0 < \rho < \rho_{th}$, J1 receives both $Rreq(A)$ and $Rreq(B)$ having few bit errors so that it corrects the errors successfully using error correction code [6, 14] and/or error estimation schemes [4, 10]. However, when A and B realize that $\rho < \rho_{th}$, they do not proceed authentication; instead, they repeat the rendezvous procedure. From the above discussion it is clear that, SBR achieves protected neighbor detection and rendezvous with low latency. The worst case latency, which occurs when $0 \leq \rho < \rho_{th}$, is λ (one period of $Rreq$) + l.

In the second attack scenario, J emits noise signals. In Fig. 6, J3 attacks B so that B may have hard time to receive $Rreq(A)$, but A can receive $Rreq(B)$ successfully. SBR is robust against the noising attack as long as one node decodes the first frequency index of $Seq(\cdot)$ of the communicating partner. Fig. 7, extended from Fig. 4(c), illustrates three scenarios where B receives two $Rreq$ packets which are collided with the noise signal of J3. In all cases, A receives $Rreq(B)$ correctly. In the first two cases, B detects the noise signal while receiving $Rreq(A)$. B cannot recover $Seq(A)$ and construct $\overline{Seq}$ fully. Instead, it decodes the first frequency index f_1 of $Seq(A)$ and uses it to begin FHSS for authentication in which it requests to update the sequence to $\overline{NSeq}$. Because A is aware of $Seq(B)$, they can keep authenticating each other with the new sequence. In the case Fig. 7(c), B cannot decode $Seq(A)$ at all, so it cannot proceed with authentication. On the other hand, A starts authentication with $\overline{Seq}$, but does not hear B's response in the first slot. A, then, returns back to neighbor detection. In this scenario, the wasted time is $\lambda + \delta$ (one slot duration). However, these slot times can be set quite small, thus the time overhead of SBR against the noising attack is negligible.

Lastly, SBR is robust against the inserting attack. The jammer J impersonates a legitimate node by broadcasting $Rreq(J)$. However, J cannot forge the certificate of a legitimate node. Suppose J1 in Fig. 6 tries to connect to A. Since A does not distinguish the jammer from legitimate B in neighbor detection, they proceed to authentication after sharing $\overline{Seq}$. During authentication, however, A detects the jamming attack and terminates the connection with J1. The time overhead is computed as $2l$ for neighbor detection + ($\mid M_A \mid + \mid M_B \mid$) for authentication. Even though the time penalty is higher than in previous attacks, even the insertion attack is foiled by SBR.

V. Evaluation

We evaluate the proposed SBR with respect to three jamming models: eavesdropping, noising, and inserting. SBR and other algorithms are implemented and evaluated on MATLAB.

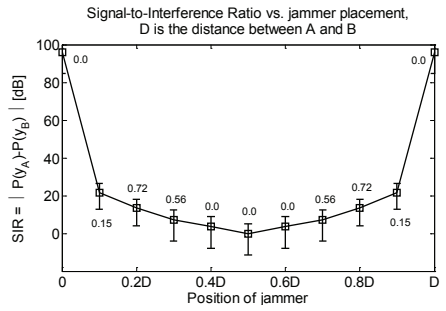

Fig. 8. J1 receives two signals and is able to decode one message if SIR $> \tau$. Power difference of two signals (SIR), $\mid P(\mathrm{y}_A) - P(\mathrm{y}_B) \mid$ [dB], is measured along varying positions of J1.

A. Eavesdropping capability

To evaluate robustness of SBR against the eavesdropping attack, we use the network configuration with J1 in Fig. 6, where the distance between node A and node B is defined as *D*. The position of J1 is represented in terms of *D*. For instance, 0.3*D* indicates that J1 places closer to A whereas the position is 0.5*D* if it is in the middle. We set *D* so that the power of an incoming signal at a receiver is -85 dB, the receive sensitivity threshold. Thus, if J1 is on *D*, it receives $\mathrm{y}_A[n]$ of -85 dB and $\mathrm{y}_B[n]$ of 10 dB. During message transmission, J1 receives two signals, $\mathrm{y}_A[n]$ and $\mathrm{y}_B[n]$, and its message reception is determined by the signal-to-interference ratio (SIR) among those signals. As J1 spans between A and B, we redefine SIR as $\mid P(\mathrm{y}_A) - P(\mathrm{y}_B) \mid$ [dB]. We measure the SIR values as a function of J1's position, and Fig. 8 demonstrates the results. The left half of the graph denotes that J1 is closer to A, i.e., $P(\mathrm{y}_A) > P(\mathrm{y}_B)$, which implies that J1 is more likely to decode $\mathrm{y}_A[n]$ after handling $\mathrm{y}_B[n]$ as an interference. On the right half, J1 might decode $\mathrm{y}_B[n]$ in the same way. However, J1 only decodes one message at a time, which does not allow it to recover a shared sequence $\overline{Seq}$. Thus, a simple eavesdropping jammer cannot disrupt SBR communication.

We also consider a highly advanced jammer that can leverage the state of the art technology of successive interference cancellation (also called successive decoding, SD). SD exploits *capture effect*: When two signals collide, a stronger signal is first decoded during a collision, and then the recovered signal is subtracted from the collided one to obtain the other signal. According to the experimental results in [7], two signals can be completely recovered when 9 dB $\leq$ SIR $\leq$ 11 dB, and recovery probability decreases as SIR deviates from the range. With SD, the eavesdropping attack can succeed with probability according to the position. The probability is denoted in Fig. 8. In the worst case, when J1 places on either 0.2*D* or 0.8*D*, it could decode both $\mathrm{y}_A[n]$ and $\mathrm{y}_B[n]$ and intrude SBR communication with probability of 72%. The following experiments take into account such probabilities and evaluate the latency performance of SBR communication.

B. Compare SBR with existing method

To estimate the effect of jamming, we measure communication latency for neighbor detection and authentication.

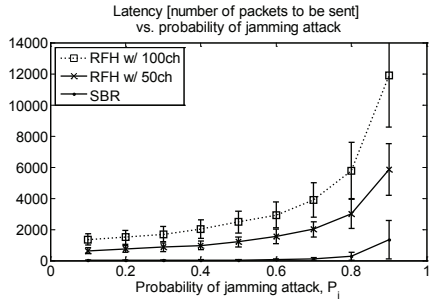

Fig. 9. Comparison of latency performance in the presence of the noising jammer: SBR vs. RFH. The jammer for SBR is assumed to perform successive decoding so that it listens to signals and then launch an attack on the agreed sequence.

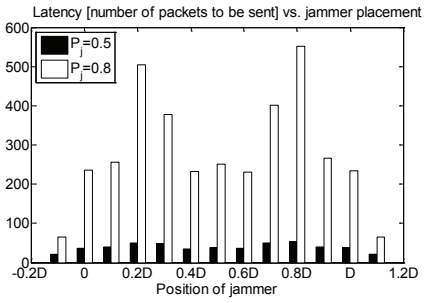

Fig. 10. Latency performance of SBR in terms of jammer placement. The jammer is assumed to be powerful so that it can launch three attacks, noising, eavesdropping, and inserting, simultaneously.

To this end, we define P_j as probability of jamming attack; for instance, P_j=0.6 represents that a packet (signal) from a legitimate node is jammed and thus fails to be delivered with probability of 60%.

We implement SBR and Random Frequency Hopping (RFH) to compare latency performance in the presence of the eavesdropping and noising jammer. Without neighbor detection stage, two communication nodes in RFH hop over random frequencies individually and try to exchange their authentication messages, M_A and M_B, whenever they rendezvous by chance. Note that increasing number of sub-frequencies in RFH could mitigate the effect of jamming. Thus, this experiment considers 50 and 100 sub-frequencies. On the other hand, nodes in SBR perform neighbor detection first, and then exchange their authentication messages over an agreed hopping sequence. We assume that each authentication message is fragmented into m (=6) packets [13]. Thus, RFH exchanges $2m$ packets to complete authentication, whereas SBR exchanges $2m$+2 packets including two $Rreq(\cdot)$s. The jammer emits noise signals randomly so that it destructs a legitimate packet with P_j. We assume here that the jammer is not persistent (i.e. $P_j \simeq 1$) and not colluded. In such severe jamming, we may consider a quorum-based FH scheme for neighbor detection procedure to minimize latency [12]. The jammer in SBR is also able to hear neighbors, recovers $\overline{Seq}$ with SD, and then attack on the agreed sequences of frequencies. When a SBR node detects failure of first $m/2$ consecutive packets on authentication, it stops authentication immediately and repeats neighbor detection again.

Fig. 9 shows the results. As P_j increases, latency in RFH increases with exponential tail. In particular, when there are 100 sub-frequencies, nodes are required to send 12K packets to finish authentication. 6K packets are sent with 50 sub-frequencies. The jammer hardly expects nodes' hopping sequences with more sub-frequencies, but this pays for slow latency. Moreover, nodes rely on accidental rendezvous to exchange messages, which aggravate the latency performance. On the other hand, SBR demonstrates almost consistent latency outcomes over all ranges of P_j. It's better results are attributed to the fact that authentication messages are exchanged over agreed hopping sequences that are less affected by the jamming attack. Even with 0.9 of P_j, the latency of SBR does not go beyond 2K packets.

C. Performance of SBR

The previous results illustrate average latency of SBR for comparison. This experiment takes into account details of SBR to show fine-grained performance. For instance, as shown in Fig. 8, the jammer using SD has different capability of eavesdropping according to its position.

1) Effect of noising attack: Because J1 places between A and B, its noise signal affects both of them. We take P_j to represent the probability of J1's noising attack. On the other hand, J2 or J3 influences either A or B. As analyzed in Section IV, regardless of noising attack from J3, if B can decode the first frequency index f_1 of $Seq(A)$, it can successfully proceed authentication. In other words, if B receives the first $x\%$ of A's signal, it can decode f_1. For analysis, we assume that the length of the noise signal is two times longer than that of a legitimate signal, i.e., $2l$. Given that B's reception is interfered by J3 with P_j, the probability that B receives the first $x\%$ of A's signal is computed as $(1-P_j)+\frac{l}{(l+2l)}\frac{(100-x)}{100}P_j$. The same calculation applies to A attacked by J2. For experiments, we set $x = 10$.

2) Effect of eavesdropping attack: J1 can eavesdrop signals using SD in the middle. The jammer is also able to launch an attack based on the eavesdropping outcome. Once the sequence $\overline{Seq}$ is recovered, it emits noise signals on the corresponding hopping frequencies and blocks the legitimate communication. Both A and B detect such attack by monitoring failures of packet transmission in authentication. When A detects that the first $m/2$ consecutive packets are dropped, it stops authentication immediately. Unlike J1, J2 and J3 cannot listen to both nodes at the same time. Thus, A and B are not affected by eavesdropping attack from J2 and J3.

3) Effect of inserting attack: The node B is vulnerable to J3's impersonating attack which is represented by P_j. If attacked, B wastes $m/2$ time slots in authentication. J1 affects A in the same way. Note that the attack message is also l long. J1 influences both A and B, so the latency penalty doubles. But, when they collide together, J1 also fails to attack. Thus, given P_j, the probability that J1 succeeds is computed as $2P_j - P_j^2$.

The latency performance is illustrated in Fig. 10. In the simulation scenarios, the jammer is assumed to be powerful so that it can launch three attacks simultaneously. In other words, two legitimate nodes are influenced by all of noising, eavesdropping, and impersonating attacks. However, as analyzed, the impact of P_j on nodes is different according to the position of the jammer. As shown, when the jammer places on $0.2D$ or $0.8D$ with P_j=0.8, SBR shows the worst latency performance. The jammer in the position is more likely to eavesdrop legitimate communication and to recover the common hopping sequence for authentication. Moreover, its impersonating attack affects both nodes at the same time, which also worsens the latency outcomes. Unlike the J1 case, J2 and J3 (-$0.1D$ and $1.1D$ in the figure) disturb neighbor detection and authentication less severely. The black bars in Fig. 10 depicts the latency performance when jamming becomes less hostile, i.e., P_j=0.5. On every positions, the number of packet to be sent is less than 50. Recall that RFH with 100 sub-frequencies in Fig. 9 requires to transmit around 2K packets under the same P_j. Comparison obviously shows that SBR performs efficiently with moderate jamming attacks. Results in Fig. 10 also discovers that SBR works quiet adaptively to the severity of the jamming attack.

VI. Conclusion

We have presented a novel, jam-protected rendezvous protocol, SBR. To reduce latency, SBR performs neighbor detection before authentication. During detection, nodes exchange a common secret key in plaintext, achieving a quick rendezvous. To protect the plaintext, the peer nodes transmit packets simultaneously, and leverage SIC to recover the packets while preventing the attacker from decoding. The paper investigated the feasibility of self interference cancellation in an adversarial MANET and analyzed the effect of three jamming attack models based on different jammer position. The simulation results show that the proposed SBR could withstand various jamming attacks in a time-efficient and secure way.

Deploying and testing SBR on a vehicular testbed is in our future plans. We also plan to evaluate the impact of mobility on SBR performance. The SIC strategy also allows to perform full-duplex communications. We will carry out experiments to determine its feasibility in VANET and its impact on throughput performance.

References

[1] Quellan inc. qhx220 narrowband noise canceller ic. http://www.quellan.com/products/qhx220_ic.php.

[2] C. Capar, M. Zafer, D. Goeckel, D. Towsley, and D. Agrawal. Physical-layer-enhanced wireless secret key exchange. Technical Report UM-CS-2010-032, University of Massachusetts, Amherst, 2010.

[3] P. Castoldi. *Multiuser Detection in CDMA Mobile Terminals*. Artech House Publishers, 1 edition, 2002.

[4] B. Chen, Z. Zhou, Y. Zhao, and H. Yu. Efficient error estimating coding: Feasibility and applications. In *ACM SIGCOMM*, Aug. 2010.

[5] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *ACM MobiCom*, 2010.

[6] M. Elaoud and P. Ramanathan. Adaptive use of error-correcting codes for real-time communication in wireless networks. In *IEEE Infocom*, March 1998.

[7] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *ACM SIGCOMM*, 2008.

[8] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and mitigating the impact of rf interference on 802.11. In *ACM Sigcomm*, 2007.

[9] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *ACM MobiCom*, 2008.

[10] K. Jamieson and H. Balakrishnan. Ppr: Partial packet recovery for wireless networks. In *ACM SIGCOMM*, Aug. 2007.

[11] T. Jin, G. Noubir, and B. Thapa. Zero pre-shared secret key establishment in the presence of jammers. In *ACM Mobicom*, 2009.

[12] E.-K. Lee, S. Y. Oh, and M. Gerla. Frequency quorum rendezvous for fast and resilient key establishment under jamming attack. In *ACM Mobicom poster*, 2010.

[13] E.-K. Lee, S. Y. Oh, and M. Gerla. Randomized channel hopping scheme for anti-jamming communication. In *Wireless Days Conference*, Oct. 2010.

[14] K. C.-J. Lin, N. Kushman, and D. Katabi. Ziptx: Harnessing partial packets in 802.11 networks. In *ACM Mobicom*, Sep. 2008.

[15] B. Radunovic, D. Gunawardena, P. Key, A. Proutiere, N. Singhy, V. Balan, and G. Dejean. Rethinking indoor wireless: Low power, low frequency, full-duplex. Technical Report MSR-TR-2009-148, Microsoft Research, 2009.

[16] A. Raghavan, E. Gebara, E. Tentzeris, and J. Laskar. Analysis and design of an interference canceller for collocated radios. *IEEE Transactions on Microwave Theory and Techniques*, 53(11):3498–3508, Nov. 2005.

[17] S. Sen, R. R. Choudhury, and S. Nelakuditi. Csma/cn: Carrier sense multiple access with collision notification. In *ACM Mobicom*, 2010.

[18] D. Slater, P. Tague, R. Poovendran, and B. Matt. A coding-theoretic approach for efficient message verification over insecure channels. In *ACM workshop on Wireless Security (WiSe)*, 2009.

[19] M. Strasser, C. Popper, and S. Capkun. Efficient uncoordinated fhss anti-jamming communication. In *ACM MobiHoc*, 2009.

[20] M. Strasser, C. Popper, S. Capkun, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE Symposium on Security and Privacy*, 2008.

[21] A. X9.63-2001. Key agreement and key transport using elliptical curve cryptography. Technical report, American National Standards Institute, 2001.

[22] W. Xu, T. Wood, W. Trappe, and Y. Zhang. Channel surfing and spatial retreats: defenses against wireless denial of service. In *ACM WiSec*, 2004.