Diffie-Hellman in the Air: A Link Layer Approach for In-Band Wireless Pairing

Wenlong Shen, *Student Member, IEEE*, Yu Cheng, *Senior Member, IEEE*, Bo Yin, *Student Member, IEEE*, Jin Du, *Graduate Student Member, IEEE*, and Xianghui Cao, *Senior Member, IEEE*

***Abstract*—Key establishment is one fundamental issue in wireless security. The widely used Diffie-Hellman key exchange is vulnerable to the man-in-the-middle (MITM) attack due to its lack of mutual authentication. This paper presents a novel in-band solution for defending the MITM attack during the key establishment process for wireless devices. Our solution is based on the insight that an attacker inevitably affects the link layer behavior of the wireless channel, and this behavior change introduced by the attacker can be detected by legitimate users. Specifically, we propose a key exchange protocol and its corresponding channel access mechanism for the protocol message transmission, in which the Diffie-Hellman parameter is transmitted multiple times in a row without being interrupted by other data transmissions on the same channel. The proposed key exchange protocol forces the MITM attacker to cause multiple packet collisions consecutively at the receiver side, which can then be monitored by the proposed detection algorithm. The performance of the proposed solution is validated through both analysis and simulations and the results show that the proposed solution is secure against the MITM attack and can achieve a low false positive ratio. The proposed solution is in-band, and can be implemented on off-the-shelf wireless devices.**

***Index Terms*—Diffie-Hellman, device pairing, in-band, MITM attack, link layer defense.**

I. Introduction

With the explosive growth of mobile devices, smart home appliances, smart cars, unmanned aerial vehicles, and many other Internet of things, people are living in a world of wirelessly connected devices [1]–[5]. Securing the data communication between these devices is of critical importance, especially when sensitive personal data are involved [3], [6]–[9]. Cryptographic solutions can be implemented to protect data communication, however, how to distribute the cryptographic key in the first place is a nontrivial task [10]. It is known that the Diffie-Hellman (DH) key agreement allows two parties with no pre-shared knowledge to jointly establish a shared secret key over the public channel [11]. Although the DH protocol is secure against eavesdroppers, its lack of mutual authentication makes it vulnerable to the man-in-the-middle (MITM) attack. The typical approach to address MITM attack is to execute a device pairing protocol which usually utilizes an out-of-band (OoB) channel which inevitably involves human interactions [12]. Besides the inconvenience brought by human efforts, the utilization of the OoB channels is also restricted by certain user interfaces or device hardware, such as a keyboard or screen. With all these limitations of the OoB channel, it is of significant importance to seek in-band solutions for the initial trust establishment between wireless devices.

The necessity of the OoB channel in a device pairing protocol is related to the Dolev-Yao attack model [13], in which an attacker has full control of the channel and can overhear, intercept, and synthesize any message. This adversary model offers the opportunity to design an in-band solution to deal with the MITM attack during the key exchange process [14]–[16]. The ideas of these works are based on the I-code technique [17], which can protect the integrity of a message payload by modulating the message signal with Manchester coded ON/OFF keying. There are two major limitations of this type of solutions. First, although the I-code technique protects the integrity of the protocol messages from being modified by the MITM attacker, it cannot prevent an impersonation attacker. PHY-UIR (physical-layer key agreement with user introduced randomness) is similar to DH key agreement with more resistance to key manipulation attacks, but still vulnerable to the MITM attack [18]. To provide authentication, OoB channels are inevitably involved, such as the WiFi push button configuration that requires the user to physically click the button on the device [14]. Second, the I-code technique requires modification to the physical layer signal modulation method, which is not easy to be implemented on off-the-shelf wireless devices.

In this paper, our solution is based on the link layer behavior monitorability of a wireless channel, i.e., the attacker's behavior inevitably impacts the wireless channel behavior at the link layer, and this link layer behavior change can be observed by legitimate users. Specifically, to launch the MITM attack, an attacker has to replace the original message with his own by intercepting the original messages and forge new ones. This may cause packet collisions which are perceptible to the legitimate receiver. We

Manuscript received January 13, 2021; revised June 29, 2021; accepted September 24, 2021. Date of publication September 30, 2021; date of current version November 18, 2021. The work of W. Shen, Y. Cheng, and B. Yin was supported in part by NSF under Grants CNS-1320736 and ECCS-1610874. The work of J. Du, and X. Cao was supported in part by the National Natural Science Foundation of China under Grant 92067111, in part by the Outstanding Young Scholarship of Jiangsu Province under Grant BK20180012, and in part by Defense Industrial Technology Development Program under Grant JCKY2020206B068. The review of this article was coordinated by Prof. Xiaojiang Du. *(Corresponding author: Xianghui Cao.)*

Wenlong Shen and Bo Yin were and Yu Cheng is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: wshen7@hawk.iit.edu; cheng@iit.edu; byin@hawk.iit.edu).

J. Du and Xianghui Cao are with the School of Automation, Southeast University, Nanjing 210096, China (e-mail: dujin@seu.edu.cn; xhcao@seu.edu.cn).

Digital Object Identifier 10.1109/TVT.2021.3116619

exploit this link layer behavior monitorability property of the wireless channel to design a novel in-band solution to deal with the MITM attack. To achieve this, our protocol design has the legitimate user transmit the DH key exchange message multiple times in a row, such that a MITM attacker has to jam all these messages, which will lead to a burst sequence of packet collisions at the receiver's antenna.[1] This abnormal link layer behavior can be then detected by the detection algorithm at the receiver's side. The details of our proposed scheme are designed specifically for IEEE 802.11 based wireless networks, one of the most widely used wireless communication standards. However, the methodology of our solution can be used for designing key establish protocols for other distributed coordinated contention based wireless networks, such as IEEE 802.15.4 networks. The main contributions of this paper can be summarized as follows.

1) We study the MITM attack over wireless networks where we model the attacker's behavior on the link layer.
2) We propose a DH-based key establishment protocol along with a channel access mechanism. The proposed protocol forces a successful MITM attacker to cause consecutive packet collisions at the link layer.
3) We design an attacker detection algorithm, which can distinguish the consecutive packet collision introduced by the MITM attacker from normal packet collisions.
4) We evaluate the performance of our proposed solution through both theoretical analysis and simulation. The proposed solution has zero missed detection ratio and can achieve a low false positive ratio.

The remainder of this paper is organized as follows. More related works are briefly reviewed in Section II. Section III presents the adversary model and the attacks behavior model. The proposed key establishment protocol and the attacker detection mechanism is introduced in Section IV. Section V and VI show the performance of the proposed solution through theoretical analysis as well as numerical and simulation results. We conclude this paper in Section VII.

II. Related Work

Establishing a shared secret key over a public channel can be achieved using a cryptographic method such as the DH key exchange [1], [11]. However, running the standard DH key exchange protocol over a public channel is vulnerable to the MITM attack. As a typical attack method, resistance against MITM attack is usually of particular concern in application scenarios where communication and mutual authentication take place between two entities in a local area [8], [19]–[22]. Many research efforts have been devoted to developing device pairing protocols, which usually leverage OoB channels to provide the mutual authentication required to prevent the MITM attack. The OoB channel is assumed to possess certain security properties, for example, it is only accessible by the legitimate users, which helps verify the message source.

OoB channels usually require non-trivial human effort and advance user interfaces. Typical OoB channels used in device pairing include mechanical vibration [23], visual channels such as device screen [24] and LED lighting [25], [26], audio channels such as speaker and microphone [27], [28]. There are new OoB channels emerging with the sensing modalities in advanced smart devices, like synchronized drawing [29]. There are also methods exploiting the shared environmental context, such as lightening and sound, to verify the proximity of the involved devices [30]. [31] utilized a public ledger to prevent MITM attack, but user interaction was requisite. Again, these solutions require additional sensing modalities that are not available to all devices, and cannot prevent a nearby MITM attacker that sharing the same physical context.

There are a few attempts to develop in-band solutions for initial trust establishment [14]–[16]. The insight of these solutions is similar to the I-code technique [17], which protects the integrity of the message payload [32] by modulating the message signal with Manchester coded ON/OFF keying. Specifically, in [14], the author designed a tamper-evident announcement (TEA) message format which improves the performance and fixed the security vulnerability of the I-codes by introducing an exceptional long synchronization packet to guarantee an adversary cannot hide the fact that a TEA message is being transmitted. However, this solution relies on the push button configuration to provide authentication, which in fact is an OoB channel. [15] proposed an in-band solution for trust establishment among multiple users, which can compare multiple authentication strings at the same time. [16] developed a group key establishment protocol for IEEE 802.15.4 based network, in which message self-authentication is achieved by combining the I-code integrity guarantee property and the transmission prescheduling function of the IEEE 802.15.4 superframe structure. However, it relies on the assumption that there exists a trustable coordinator. Besides, these I-code solutions require modification to the physical layer signal modulation method. [33] took advantage of the existence of multiple devices to perform message integrity verification. Some studies utilized RSS (Received Signal Strength) variations to generate and extract secret key, but it was not suitable for static environment because the lack of variation makes it highly predictable [34]. [35] used RSSI (Received Signal Strength Indicator) to detect MITM attack and rogue AP, but its accuracy was not perfect.

III. MITM Attack Modeling

In this section, we introduce the models of the MITM attacker and its behavior on the message level. We present the best strategy for the attacker in an IEEE 802.11 network, and describe our system model and problem statement.

A. MITM Attack in Diffie-Hellman Key Exchange

As shown in Fig. 1(a), the DH key exchange protocol works as follows: Assume Alice and Bob agree on a large prime p and a finite cyclic group $\mathcal{G}$ of order o with a generator g. Alice randomly picks a secret value a $(0 \leq a \leq p-1)$ and calculates $g^a \mod p$, and Bob randomly picks a secret random value b

[1] In this paper, we use "packet" to generally mean a chunk of information delivered over the network. We use "packet," "message," and "link-layer frame" interchangeably, for the convenience of presentation. In our protocol, each message is carried by a single link-layer frame.

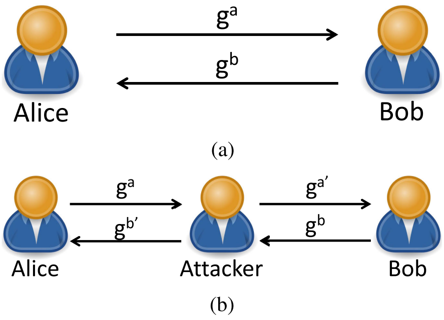


Fig. 1. The DH key exchange protocol and the MITM attack scenario.

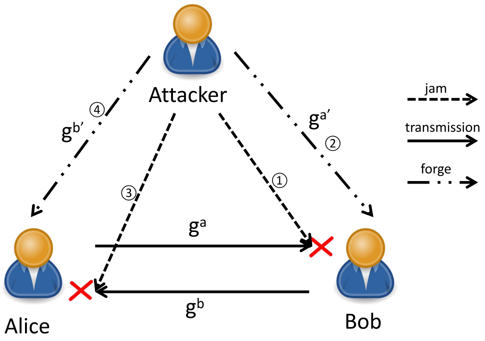


Fig. 2. Wireless MITM attack scenario.

$(0 \leq b \leq p-1)$and calculates $g^b \mod p$. Then Alice and Bob exchange their value of g^a and g^b (all values are $\mod p$ unless otherwise specified). At the last stage, Alice calculates $K_A = (g^a)^b$ and Bob calculates $K_B = (g^b)^a$. Both Alice and Bob will arrive at the same secret value K since $(g^a)^b = (g^b)^a$.

Although the DH key exchange is secure against passive attackers (eavesdroppers), as is well known, it is vulnerable to the MITM attack. Since the message transmission is conducted over a public channel, and there is no pre-shared secret between Alice and Bob, the received g^a and g^b cannot be authenticated. In the MITM attack scenario, the attacker intercepts the legitimate messages and forges fake ones. For example, as shown in Fig. 1(b), the attacker intercepts the message g^a, and sends $g^{a'}$ to Bob pretending himself to be Alice. The attacker will also intercept g^b and forge a $g^{b'}$. As a result, the attacker establishes two secret keys with Alice and Bob respectively, while Alice and Bob think they have established a shared secret key with each other.

B. Adversary Model in Wireless Communications

The MITM attack is easy to be implemented in wired networks as the attacker may get physical access to the cables and then intercept or manipulate the message signal without being noticed by the receiver. However, in wireless scenarios, message transmissions take place over the open shared wireless medium, where it is not so easy for the attacker to take full control of the wireless channel. As a result, some common attacker vectors such as message interception and message modification are difficult, if not impossible, to be practically implemented in wireless communications.

The most commonly used, which is also the strongest attacker model, is the Dolev-Yao model [13], [36], in which the attacker has the capability to eavesdrop, modify, compose, and replay any messages transmitted and received by legitimate devices. In this model, in addition to eavesdropping and insertion, the attacker can fully modify and annihilate signals at the receiver's antenna. However, this may require the attacker be able to measure distances and estimate the channel with high precision to any target node, and be able to achieve perfect carrier phase synchronization and precisely control the signal amplitude levels at the receiver. In practical applications with time-varying fading channels (e.g., mobile networks), the state information of the sender-receiver channel is almost unavailable to the attacker, and carrier phase synchronization and amplitude control at the receiver is difficult. Therefore, we assume that deterministic message modification and message annihilation is not achievable for the attacker under study.

In this paper, we consider a realistic adversary model which can eavesdrop and replay legitimate wireless messages, and can insert messages at arbitrary time. Such an attacker is practical for devices with the state-of-art RF techniques such as beamforming and ability to perform spatial selectivity transmit or jam signals to one particular target. As discussed above, the only limitation of the attacker is that he cannot arbitrarily manipulate nor annihilate an ongoing wireless message. Here we do not consider that an attacker utilizes the capture effect to manipulate a message, since a significantly overpowered signal transmitted by the attacker can be detected by applying a received signal strength threshold at the legitimate users. We shall focus on cyber-space attack, thus assume that the attacker do not physically obstruct legitimate communication nodes by means such as moving, powering off or damaging.

C. Wireless MITM Attacker Behavior Modeling

Based on the above attacker model, in order to launch the MITM attack during the DH key exchange process, the best the attacker can do is to send a jamming signal at the same time when Alice transmits g^a, resulting in a packet collision at Bob's receiver, such that Bob cannot decode the message g^a from Alice. This can be achieved by either jamming the complete message or jamming only the message preamble [36]. After jamming the original g^a, the attacker then has to forge a $g^{a'}$ using Alice's identity criteria (usually Alice's IP address and MAC address) and sends it to Bob. At Bob's point of view, the failure of decoding the original message seems to be caused by normal packet collision, which is quite often in random access based wireless networks, such as IEEE 802.11 and IEEE 802.15.4 based wireless networks. Bob will accept the forged $g^{a'}$ as the legitimate message from Alice since there is no authentication mechanism. The attacker then performs the same strategy to g^b and successfully launched the MITM attack. The attack scenario is illustrated in Fig. 2.

Based on the behavior order of the attacker, here we can further categorize the MITM attack into two categories. If the attacker follows the order of 1-2-3-4 as illustrated in Fig. 2, we

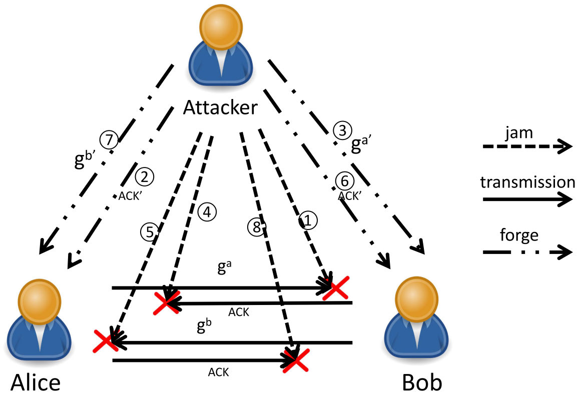


Fig. 3. MITM attack in IEEE 802.11 based network.

define this type of attack as **Type I** attack. However, the attacker can also conduct the MITM attack by performing steps 1-4-2-3. In this case, the attacker first impersonates Bob to establish a shared key with Alice, then impersonates Alice to establish a shared key with Bob. In this paper, we term the second type of attack as **Type II** attack.

It is worth noting that for the transmission of the inserted messages $g^{a'}$ and $g^{b'}$, the attacker has to use directional antenna such that the party being impersonated cannot receive the message signal, otherwise the attack will be detected with minor efforts: Alice and Bob can keep monitoring the channel, and raise an alarm if they see any packet being transmitted is using their own IP and MAC addresses.

Consider a practical scenario where Alice and Bob communicate via an IEEE 802.11 based WiFi network. Since, by the standard, the receiver replies the sender an ACK upon successfully receiving a packet, if g^a is jammed by the attacker, Bob will not correctly decode this message, thus Alice will not receive the corresponding ACK. Then, after a certain period of time, Alice will try to retransmit the message, until a maximum retransmission counter is reached. If the attacker keeps jamming all the retransmission from Alice, Alice will notice the abnormal behavior of the wireless channel and may stop trying to establish the secret key with Bob. If the attacker ignores Alice's retransmission of g^a and still forges his $g^{a'}$, Bob will receive both key exchange messages, which indicates the existence of the attacker. In both cases, the MITM attack will not succeed. The best strategy for the attacker is after jamming g^a, he forges an ACK to make Alice believe that the message g^a has been received correctly by Bob, then the attacker can forge a $g^{a'}$ and send it to Bob. However, after receiving the forged $g^{a'}$, Bob will send an ACK, and the reply address of this ACK will be Alice's MAC address. The attacker has to jam this ACK to prevent Alice from receiving double ACKs for only one message transmission.

Fig. 3 presents the attacker's best strategy. In summary, to successfully launch the MITM attack in an IEEE 802.11 based wireless network with ACK mechanism, the attacker has to follow the 8 steps illustrated in Fig. 3. A **Type I** attacker will follow the orders of 1-8 and a **Type II** attacker will follow the steps of 1-2-7-8-3-4-5-6. In the attack process, although it is possible to jam the transmitter during the backoff window time, the result will be that the transmitter defers its transmission, which does not benefit the attacker. This is because the backoff counter of the transmitter will be frozen until the channel is detected idle for a period of time according to IEEE 802.11. In this sense, an efficient way for the attacker is to jam the channel when the two key pairing parties are transmitting messages.

D. System Model and Problem Statement

We shall focus on IEEE 802.11 based networks to present the system model and our design while it should be noticed that the proposed scheme can be extended to other wireless networks with contention based MAC. Consider an IEEE 802.11 based wireless network consisting of some stations (among which is Alice) and an access point (Bob). We assume that Alice has already performed the scanning process and decided to associate and establish a secrete key with Bob. In this scenario, Alice and Bob first establish a wireless link between them by going through an association handshake, in which Alice sends an association request and Bob replies with an association reply. Immediately after the association handshake, Alice and Bob try to establish a shared secret key by a DH key exchange. Suppose Alice initializes the key exchange protocol. During this process, an attacker with capabilities defined above may try to launch the MITM attack. The case of impersonation attack during the association process will be discussed in Section IV-E. Besides Alice, Bob and the attacker, there are n $(n \geq 0)$ other wireless stations sharing the same wireless channel. For ease of presenting the secure pairing process, we assume that Alice and Bob have no data packets to transmit other than the messages related to the key exchange protocol, while other wireless stations have data traffic which can be treated as background traffic to the key exchange process. All the background wireless stations access the wireless channel according to the distributed coordination function (DCF) specified in IEEE 802.11 standards. We assume that the n background stations are within the transmission and receiving range of Alice and Bob (no hidden terminals), and their network traffic density is stable. This assumption leads to a stable one-hop network which is practically possible if the stations (including Alice and background ones) and the access point locate within a small area (e.g., an office) and that the background stations do not have traffic bursts.

Based on the above system model and assumptions, in this paper, we aim at addressing the following problem: How can Alice and Bob prevent or detect the MITM attack during their key exchange process while using only in-band channel? We consider the attacker as being detected if both Alice and Bob detect its presence. We are seeking practical solutions that are easy to be implemented without modification to existing wireless hardware.

IV. In-Band Secure Key Establishment Protocol

Above we notice that in order to prevent Alice or Bob from receiving legitimate key exchange messages, the attacker has to transmit jamming signals to intentionally collide the legitimate messages at the receiver's antenna. To distinguish the packet collision introduced by the MITM attacker from normal packet collisions due to simultaneous transmission, our protocol requires Alice to transmit g^a multiple times consecutively, such that to successfully launch the MITM attack, the attacker has to jam all the g^a from Alice, thus resulting in a burst sequence of packet collisions at Bob's receiver. Then Bob can notice this abnormal channel behavior and detect the existence of the attacker. We achieve this by modifying the DH key exchange protocol and the channel access scheme for the protocol message transmissions, as well as carefully designing an attacker detection mechanism.

A. Preliminary: IEEE 802.11 DCF

Below we briefly describe the main procedures in the DCF of IEEE 802.11 MAC protocol. A station with a packet to transmit first monitors the channel before transmitting. If the channel is sensed idle for a distributed interframe space (DIFS) time period, then it selects a random backoff counter uniformly selected from $[0, CW]$, where CW is its current contention window size. The station decreases its backoff counter by 1 for each time slot when the channel is sensed idle after the DIFS. If the channel is sensed busy during the backoff procedure, the station freezes its backoff counter until the channel is sensed idle again for more than a DIFS time period. The station transmits when its backoff counter reaches 0. A packet collision happens when more than one stations have their backoff counter reaching zero at the same time slot and simultaneously transmit subsequently. After a short interframe space (SIFS) time period following successful reception of a packet, the receiver replies an ACK to the sender. Since SIFS is shorter than DIFS, no other stations are able to transmit until the transmission of the current ACK finishes. If a station does not receive an ACK within a timeout period after transmitting a packet, it doubles its contention window size CW until reaching an upper limit $CW_{max} = 2^{\beta} CW_{min}$. Once reaching the limit, CW remains in CW_{max} until a maximum retransmission limit is reached (then, CW is reset to CW_{min} and the packet failed to be transmitted is discarded).

B. The Proposed Key Exchange Protocol

As illustrated in Fig. 4, in the modified DH key exchange protocol, each key exchange message is transmitted m times. In this situation, to successfully perform the MITM attack, the attacker has to block all these $2m$ messages by introducing $2m$ extra packet collisions to the channel, and Alice and Bob can each observe m packet collisions. The value of m shall be determined based on the current channel condition, which will be discussed in later sections of this paper. We shall use M_i^a and M_i^b to denote the ith message from Alice and Bob, respectively; the superscript is dropped when we generally indicate a message from either Alice or Bob.

We set the message format for M_i as follows:

$$M_i = \{\ i \mid\ m \mid g^a \,(\text{or } g^b\,) \mid \text{dummy data}\},$$

where i denotes the current message number. g^a or g^b is the same as the original DH protocol. We use dummy data to fill the message payload to be the maximum size allowed in IEEE 802.11 standard in order to prevent the attacker from jamming more than one packets by sending one exceptionally long jamming signal.

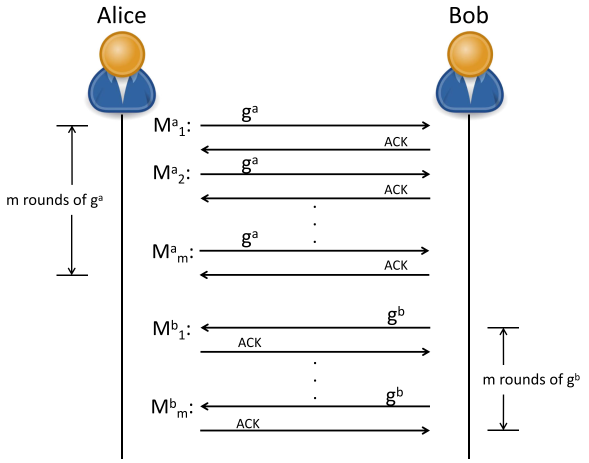


Fig. 4. The proposed key exchange protocol.

Fig. 5 illustrates the reason why we use dummy data to fill up the message payload. In Fig. 5(a), Alice transmits g^a two times without adding dummy data, and the attacker jams the message with one jamming signal. In this case Bob will detect one collision event. If the two packet transmission duration plus their inter-frame space is less than the transmission time of one maximum-sized packet, this will seem to be one normal packet collision for Bob. In Fig. 5(b), Alice transmits g^a with dummy data added, and the attacker tries to jam the message transmission with only one jamming signal. In this case, Bob will be able to observe a collision longer than normal packet transmission, thus detect the existence of the attacker. If the attacker wants to jam the maximum-sized key exchange message without being detected by the collision duration criteria, he has to jam each message separately, as shown in Fig. 5(c). In summary, with dummy data added to the key exchange messages, if the attacker attempts to prevent Bob from receiving Alice's key exchange messages, Bob is guaranteed to observe m packet collisions. The same is true for the key exchange messages from Bob to Alice. We term a packet collision with a duration longer than the time required to transmit a maximum-sized packet as an exceptionally long packet collision.

Our proposed key exchange protocol guarantees that if the attacker wants to successfully launch the MITM attack, both Alice and Bob will be able to observe m extra packet collisions. The remaining problem is how to distinguish these extra packet collisions introduced by the attacker from normal ones caused by simultaneous transmissions, since packet collision frequently happens due to the distributed nature of the channel access mechanism: if the backoff counters of two or more stations happen to reach 0 at the same time slot, a packet collision will occur. Since every station within the wireless network shares equal chance to access the wireless channel, on average Alice has to wait for n packet transmission before it gets a chance to transmit a single protocol message. This implies that in presence of the attacker, on average Bob can only observe

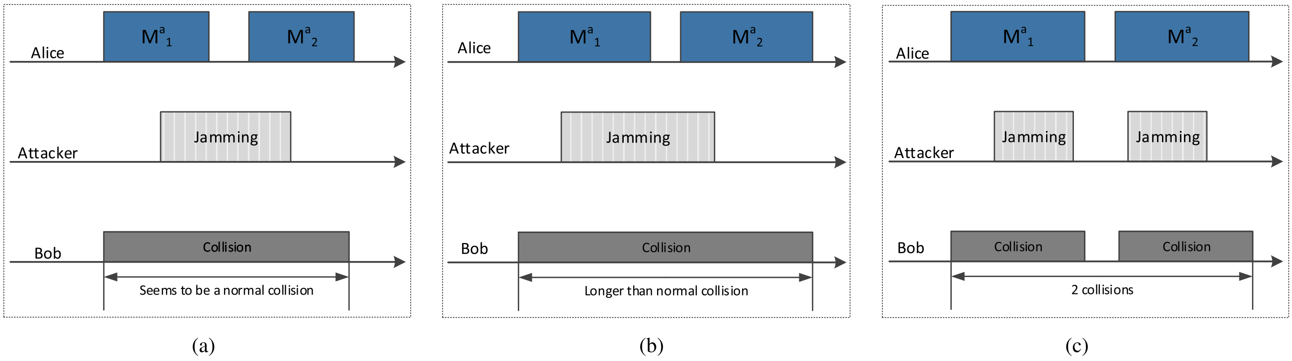


Fig. 5. Use maximum packet size for protocol message transmission. (a) Small packet size. (b) One jamming signal to jam two packets. (c) Two jamming signals.

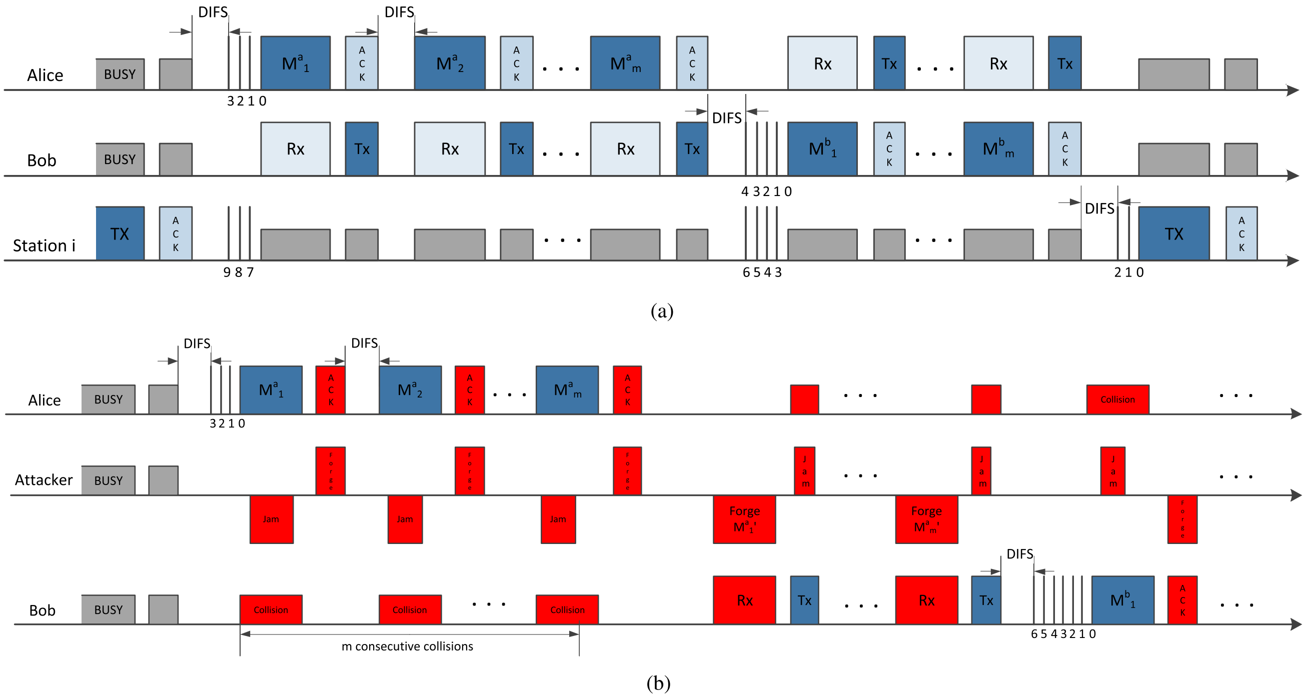


Fig. 6. Channel behavior of the proposed scheme. (a) Channel behavior in absence of the attacker. (b) Channel behavior in presence of the attacker.

one extra collision for every $n + 1$ packet transmissions, which cannot serve as a good detection criterion. In this sense, we can grant the key exchange protocol packets a higher priority to access the wireless channel over background packets, such that the extra packet collisions introduced by the attacker will not be diluted by normal collisions. If we let the key exchange messages access the channel without going through the backoff process by fixing the backoff counter to be 0, messages M_i^a and M_i^b will have priority over the background data packets.

The channel access scheme for the proposed key exchange protocol is as follows. Upon receiving the ACK of $M_{(i-1)}^a$, Alice transmits M_i^a $(2 \leq i \leq m)$ immediately after a DIFS, and upon receiving the ACK of $M_{(i-1)}^b$, Bob transmits M_i^b $(2 \leq i \leq m)$ immediately after a DIFS. The channel access scheme for M_1^a and M_1^b follow the normal backoff mechanism defined in the IEEE 802.11. This channel access scheme for the proposed key exchange protocol guarantees the channel access priority for message M_i^a and M_i^b $(2 \leq i \leq m)$. Only M_1^a and M_1^b compete for the channel with the other stations. There is a probability that the transmission of M_1^a and M_1^b encounters a collision, but the other messages in the key exchange protocol are guaranteed to be collision-free. More importantly, this channel access scheme forces the MITM attacker to consecutively collide multiple packets, such that the receiver can distinguish these m consecutive packet collisions from normal ones, thus detecting the presence of the attacker.

Fig. 6 illustrates the channel behavior of the proposed key exchange protocol with the priority channel access scheme. Fig. 6(a) shows the channel behavior of the proposed key exchange protocol when the MITM attacker absents. Alice starts the key exchange protocol by transmitting M_1^a after winning the channel competition. Other stations (station i for example) can only get access to compete for the channel after the transmission

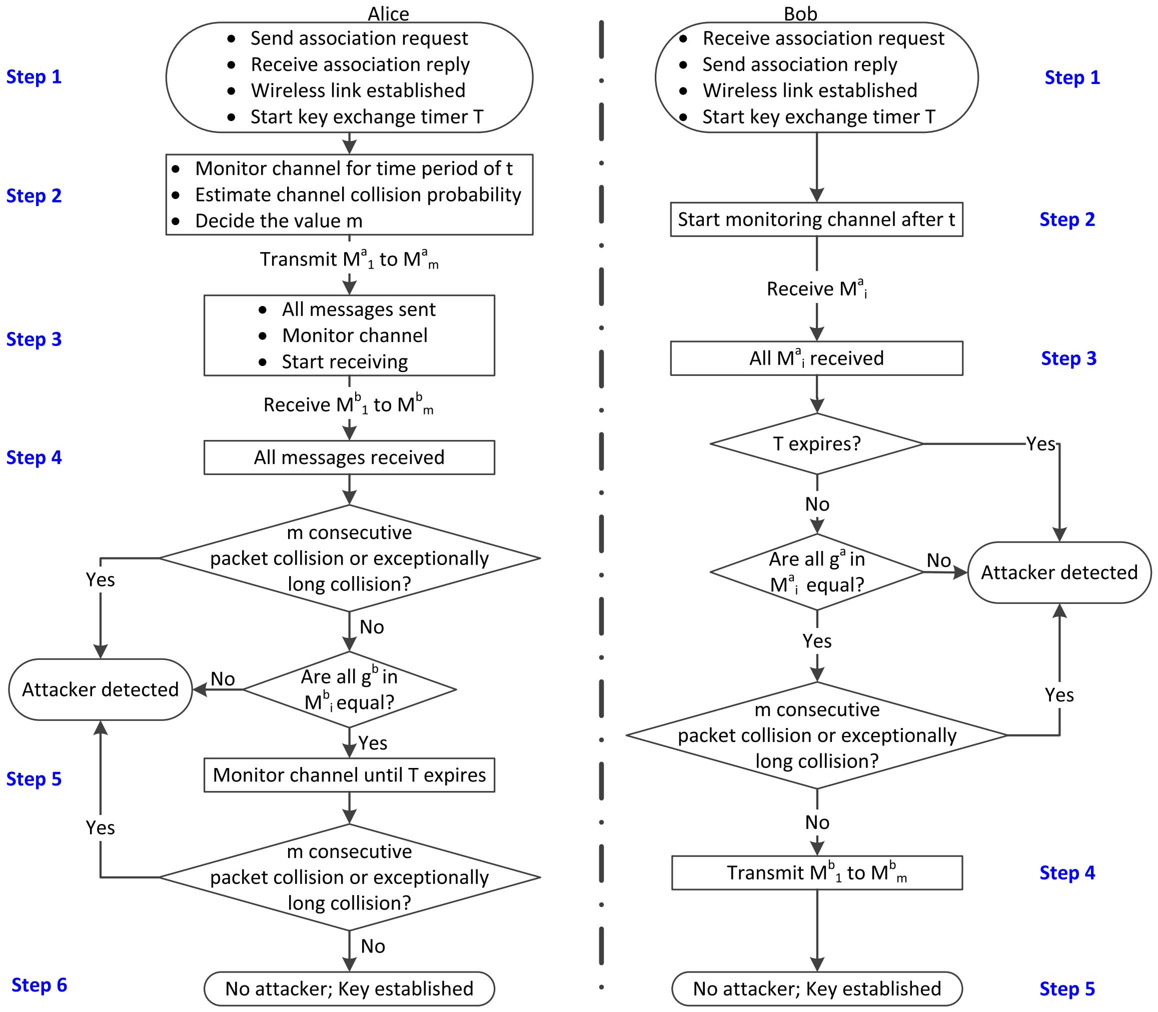

Alice
Send association request
Receive association reply
Wireless link established
Start key exchange timer T
Step 1
Monitor channel for time period of t
Estimate channel collision probability
Decide the value m
Step 2
Transmit M^a_1 to M^a_m
All messages sent
Monitor channel
Start receiving
Step 3
Receive M^b_1 to M^b_m
All messages received
Step 4
m consecutive packet collision or exceptionally long collision?
Yes
No
Are all g^b in M^b_i equal?
No
Attacker detected
Yes
Monitor channel until T expires
Step 5
Yes
m consecutive packet collision or exceptionally long collision?
No
No attacker; Key established
Step 6
Bob
Receive association request
Send association reply
Wireless link established
Start key exchange timer T
Step 1
Start monitoring channel after t
Step 2
Receive M^a_i
All M^a_i received
Step 3
T expires?
Yes
No
Are all g^a in M^a_i equal?
No
Attacker detected
Yes
m consecutive packet collision or exceptionally long collision?
Yes
No
Transmit M^b_1 to M^b_m
Step 4
No attacker; Key established
Step 5

Fig. 7. System flowchart.

of M^a_m finishes, because M^a_i $(2 \leq i \leq m)$ is transmitted immediately after a DIFS, and they have no chance to decrease their backoff counters. Fig. 6(b) illustrates the channel behavior of the proposed key exchange protocol when a **Type II** attacker presents. The figure only shows the first half of the attack, where the attacker jams all the M^a_i from Alice and then forges $M^{a'}_i$ to Bob. The attacker's behavior results in m consecutive packet collisions at Bob's receiver, which only happens with low probability under normal conditions. The attacker can be then detected by our detection mechanisms (to be introduced later). The second half of the attack (omitted in Fig. 6(b) due to space limitations) will be the attacker jamming the M^b_i and forging $M^{b'}_i$ to Alice, which will result in m consecutive packet collisions at Alice's receiver.

C. Detection System Design

The key insights for the detection algorithm design are: 1) if the attacker jams all the key exchange messages from Alice (or Bob), then Bob (or Alice) will observe m consecutive packet collisions. 2) If the attacker fails to jam all the key exchange messages and still try to forge an $M^{a'}_i$ (or $M^{b'}_i$), then Bob (or Alice) will receive key exchange messages containing different parameters g^a and $g^{a'}$ (or g^b and $g^{b'}$). On this basis, we design our system as in Fig. 7. Before we get into the details, let's first introduce in practice how does a station detect a collision.

1) Detecting a Packet Collision: In an IEEE 802.11 wireless network, a station keeps monitoring the channel. Whenever the station's antenna detects an ongoing packet transmission, it will first decode the physical and MAC headers, and obtain the Destination Address (DA) contained in the MAC header. If the DA does not match its own MAC address, the station will drop the packet without decoding the data payload; otherwise, it will continue decoding the payload. If the packet can be decoded and passes the FCS check (error-detecting code), the station will send an ACK to indicate the correct reception of the packet. If the packet header or the packet payload cannot be decoded or the FCS check fails, the station will consider this packet has collided and no ACK will be transmitted.

Theoretically, we can have Alice and Bob detect the packet collisions within their antenna's receiving range by allowing them decode all the received packets, no matter whether the DA matches or not. However, this method will impose huge energy burden and shorten the lifetime of energy-restricted wireless devices. Besides, if the attacker jams the preamble of a message, the receiver will not even notice that there is an ongoing message (as synchronization will fail). In this paper, we adopt a channel collision detection method which considers the preamble jamming and is energy-efficient. We notice that from an observer's point of view, in case of a successful packet transmission, the channel occupancy follows the pattern

$$\begin{aligned}\text{Busy (duration} > \text{ACK)} \quad &\rightarrow \quad \text{Idle (duration} = \text{SIFS)}\\ &\rightarrow \quad \text{Busy (duration} = \text{ACK)},\end{aligned}$$

and in case of a collision, the channel occupancy follows:

$$\text{Busy (duration} > \text{ACK)} \quad \rightarrow \quad \text{Idle (duration} > \text{SIFS).}$$

In our proposed scheme, Alice and Bob will use these channel occupancy patterns to determine whether an ongoing packet transmission is a successful one or a collision.

2) Flowchart of the Proposed System: Now let's go through the details of the proposed system. The very first step will be Alice and Bob establish their wireless link through an association handshake. Assume Alice is always the one who initiates the association handshake and the key exchange process. The system flowchart is shown in Fig. 7.

After sending an associate request and receiving the association reply from Bob, Alice starts a key exchange timer T (step 1). Alice will only install the established key after T expires and no attacker detected. Then Alice monitors the channel for a time period of t before starting to transmit M_1^a. This time period of t is termed as the monitoring window, during which Alice estimates the channel condition (channel collision probability and channel traffic density) and decides the number of rounds m for the proposed key exchange protocol (step 2). The details of how to chose a proper m will be discussed in the next section. With m being decided, Alice can start the key exchange protocol by transmitting M_1^a to M_m^a according to the proposed channel access scheme. After finishing the transmission, Alice starts to monitor the channel and receiving M_i^b (step 3). The attacker detection decision making will begin after Alice received all the M_i^b (step 4). She will check if there are m consecutive packet collisions between step 3 and step 4, if there are exceptionally long packet collisions, and whether the received g^b are all equal. A **Type I** attacker will be detected at this point. If no attacker detected at this point, Alice still needs to monitor the channel until the key exchange timer T expires (step 5) to decide whether a **Type II** attacker exists. After T expires, if no m consecutive packet collision and no exceptionally long packet collision has been detected, Alice can confirm that there is no MITM attack during the key exchange process and install the established key.

For Bob, upon receiving the association request from Alice, it transmits the association reply and starts his key exchange timer T (step 1). Bob will start to monitor the channel after a time period of t, since during this time Alice is estimating the channel condition and has not started the key exchange protocol yet. Thus, providing that Bob starts monitoring the channel before Alice transmitting the key messages, they do not have to be exactly synchronized. After receiving all M_i^a from Alice (step 3), Bob checks if the timer T expires, if all the received g^a are equal, and if there are m consecutive packet collisions or any exceptionally long packet collision. Both the **Type I** attacker and the **Type II** attacker will be detected by these detection criteria. If there is no attacker being detected, Bob can start transmitting his M_i^b. After Bob finishes the transmission of M_m^b, he can install the established key.

3) Alternative Detector Design: Given the channel access mechanism, the channel occupancy of the protocol messages has a unique pattern: the ACK of the previous message is followed by an idle period of DIFS and then a maximum-sized packet transmission. Correspondingly, if the attacker jams all the key exchange messages, the receiver will observe m consecutive collisions with a fixed idle interval which equals to SIFS+ACK+DIFS. Based on this collision pattern, we can design a more advanced detector to distinguish this attacker behavior from normal packet collisions. Specifically, a timer can be added on top of the current detector design to record the timestamp for recent collisions, and if consecutive collisions are detected, the users can compare the timestamps to check if the idle intervals match the unique pattern. Compared to the current design, this more advanced detector can achieve 0 false positive ratio with the extra cost of memory space for recording the collision timestamps and extra codes to check the collision pattern. We suggest the current solution in this paper for better performance and cost trade-off.

D. Discussions on Practical Implementation Aspects

The proposed link-layer solution for wireless device pairing works completely in-band and does not require any OoB or human interaction or special hardware. Besides, our solution can be implemented on off-the-shelf wireless devices without modifying their physical layer transmission mechanism, and a user only needs to configure its backoff counter during the key establishment process.

1) Channel Access Mechanism: In the proposed solution, the protocol messages are transmitted immediately after a DIFS, without going through the back off process. This channel access mechanism aims at granting Alice or Bob continuously channel access during the protocol message transmission. In fact, this can be achieved by setting the inter-message space to be any value between SIFS and DIFS. We choose DIFS in our design for the purpose of easy implementation: Alice or Bob only needs to change the backoff counter configuration in the key establishment stage.

2) Imperfect Channel Condition: In practice, during the key exchange process, Alice may retransmit a message if the previous transmission was collided by other background stations or if the previous collision-free transmission arrived at Bob with some bit error. With carrier sensing and collision detection in IEEE 802.11 networks, collided transmissions will be detected and counted in the proposed detector. Therefore, even if the channel is imperfect, the proposed protocol still works along with the proposed detector as long as there are no hidden terminals.

3) RTS/CTS Mode: The details of our solution are designed for the basic access mode in IEEE 802.11 wireless network.

Besides the basic access mode, there is an RTS/CTS mode aiming at addressing the hidden terminal problem, in which each station goes through a request-to-send (RTS) and clear-to-send (CTS) handshake before the data packet transmission [37]. The proposed solution can be further extended to cover the RTS/CTS mode. In the RTS/CTS mode, only the RTS packet has a chance to collide with other RTS packets, while the data packet is collision-free. If the attacker only collides the data packet, the detection becomes trivial, so a successful attacker has to jam the CTS request. Base on this insight, we can have Alice transmit the RTS request m times consecutively using the proposed channel access mechanism, such that the attacker has to jam all the m RTS to cause m consecutive collisions, which can then be detected by Bob.

4) Other Related Issues: Although redundant transmissions of the key exchange messages make the proposed scheme less time-efficient, no extra information is leaked since the messages being exchanged are the same as those in the traditional DH protocol. Therefore, the proposed protocol is secure against not only passive eavesdropper but also MITM attack. In addition, consider the denial-of-service (DoS) attack which can be launched by deliberately jamming the channel to cause multiple packet dropouts. Although the attacker does not have to jam all the key exchange messages due to packet collisions from other normal stations, the difficulty of causing a DoS is no less than that in the traditional DH process.

E. Discussions on Impersonation Attack

The proposed scheme is based on the assumption that Alice and Bob have already gone through an association handshake and have a wireless link available between them. However, the attacker may impersonate Alice or Bob during the association phase. Specifically, when Alice sends the association request, the attacker can jam this message and then impersonate Bob to send a reply to Alice. The proposed scheme covers the key exchange process after the association, but not the association phase itself. The essential difference between the impersonation attack and the MITM attack is that in the latter case, both Alice and Bob know that a key exchange process is expected within a certain time window (corresponding to the key exchange timeout window T in the proposed solution). However, in the former case, being the passive party in the association handshake, Bob is not aware of that Alice is trying to establish a shared key with him.

The proposed solution can work on top of existing identity authentication techniques, such as the push button configuration (PBC) mechanism defined in IEEE 802.11 standards, to address the impersonation attack. However, an identity authentication protocol requires either pre-shared knowledge or OoB channels. In fact, we can address the impersonation attack by a similar link layer approach. Due to the space limitation, here we only introduce the general methodology for a secure association protocol without going deep into the details. In the secure association protocol, Alice transmits the association request m times, with random delays between each request, and Bob replies with n association replies once he receives an association request. The n replies will be transmitted using the channel access mechanism proposed in Section IV-B. To successfully impersonate Bob, the attacker has to jam all the m requests from Alice, otherwise if Bob receives a single request, the attacker has to jam the corresponding n replies, which results in n consecutive collisions at Alice's side and can be detected by Alice. With the random delay between the association requests, even if the attacker knows the exact transmission starting time of the first request, he does not know the transmission starting time of the following requests. The best he can do is to examine the packet header of every transmission on the channel, and send a jamming signal when he sees the current packet is indeed from Alice. If the packet header containing Alice's address is perceptible to the attacker, it is also perceptible to Bob. Based on this insight, Bob can keep monitoring the channel and keep a record on the source addresses of collided transmissions. If the attacker jams all the m requests from Alice, Bob will notice that m transmissions from the same address collided consecutively. At this point, Bob sends n alarm messages using the channel access scheme as in Section IV-B. If Alice receives an alarm, she will be aware of the attacker; if all the alarms are jammed by the attacker, Alice will observe n consecutive collisions and be aware of the attacker.

V. Performance Analysis

A. Missed Detection Ratio

The missed detection ratio is defined as the probability that an attacker successfully launched the MITM attack without being detected by Alice or Bob. With the proposed scheme, a MITM attacker will always be detected. The missed detection ratio of the proposed scheme is 0.

The MITM attack is considered to be successful, only if the attacker can establish two separate shared keys with Alice and Bob, while passing all the detection criteria in the proposed scheme. The proposed scheme has three detection rules: (1) all received g^a (g^b) are equal; (2) no m consecutive packet collision is detected; (3) no exceptionally long packet collision is detected. In the proposed scheme, Alice will transmit g^a m times. If the attacker does not intercept all these m messages, Bob will at least receive one g^a from Alice. In this case, if the attacker transmits his own $g^{a'} \neq g^a$ to Bob, according to detection rule (1), Bob will detect his presence. The only successful chance for the attacker is that he manages to intercept all the g^a from Alice. Under the adversary model, the attacker can only achieve this by colliding these messages at Bob's antenna. If the attacker uses one jamming signal to collide multiple M_i^a, an exceptionally long packet collision will be observed by Bob, as illustrated in Fig. 5, which violates detection rule (3). The attacker has to individually collide these m packets. According to the proposed channel access mechanism, these m packets will be transmitted consecutively without being interrupted by normal background data packets, so colliding them will result in m consecutive packet collisions at Bob's receiver, which violates detection rule (2). In summary, under the adversary model defined in Section III-B, a MITM attacker can always be detected by our detection rules. The missed detection ratio of the proposed scheme is 0.

B. False Positive Ratio

A false positive occurs when the proposed system detects an attacker but in fact no threat exists. Our system will raise an alarm if any of the following situations occurs: (1) the received g^a (g^b) does not match; (2) Alice or Bob detects m consecutive packet collisions within the detection window (the time period starts after the monitoring window until the key exchange timer T expires); (3) an exceptionally long packet collision is detected. Given the assumption that there are no hidden terminals, if there is no attacker existing, situations (1) and (3) will not happen. A false positive will only occur under the circumstance that within the detection window, there exist m consecutive packet collisions due to simultaneous transmissions. Based on the proposed scheme, athough Alice (or Bob) has a higher priority than background stations to access the wireless channel once it gains the access, it has to contend with those stations to access the channel before transmitting the first key message. Therefore, during this contention period, Bob (or Alice) may witness m consecutive collisions which are in fact due to collisions among background stations, and consequently a false alarm will be triggered.

First, we present a mathematical model for the consecutive collision detector. Let I_q be the indicator of the qth observed packet transmission, i.e.,

$$I_q = \begin{cases} 1, & \text{if the } q\text{th packet is a collision,} \\ 0, & \text{if the } q\text{th packet is a successful transmission.} \end{cases}$$

We use X_q to denote the state of the detector, then the behavior of the detector can be mathematically described as

$$\begin{cases} X_{q+1} = I_n \times (X_q + I_q) \\ X_0 = 0. \end{cases} \tag{1}$$

Based on the above discussions, we use m to be the detection threshold of the following detector:

$$\delta_q = \begin{cases} 1, & \text{if } X_q \geq m, \\ 0, & \text{if } X_q < m, \end{cases}$$

i.e., whether m consecutive collisions happen or not. The detector value X_q will be reset to 0 as soon as it exceeds the threshold m and the detection procedure starts over again.

Consider the sequence $\{X_q\}$ as a discrete random process, which takes values from a finite set $A = \{0, 1, 2, \ldots, m\}$. The detector is said to be in state $i \in A$ at step n if $X_q = i$. The state transition happens when a packet transmission over the wireless channel is observed. According to (1), the next state X_{q+1} is independent of previous states except X_q, where the transition probability is

$$P_{ij} = P\{X_{q+1} = j \mid X_q = i\}, \quad \forall i, j \in A.$$

Thus, $\{X_q\}$ can be modeled as a discrete-time Markov chain with the transition probability matrix:

$$P = \begin{bmatrix} P_{00} & P_{01} & \ldots & P_{0m} \\ P_{10} & P_{11} & \ldots & P_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m0} & P_{m1} & \ldots & P_{mm} \end{bmatrix}$$

Let p_{ch} denote the channel collision probability, which is the probability that an observed packet transmission is a collision. Assuming the collision probability of each observed transmission is independent, then we have

$$P_{ij} = \begin{cases} p_{\text{ch}}, & \text{if } j = i + 1 \text{ and } 0 \leq i \leq m - 1 \\ 1 - p_{\text{ch}}, & \text{if } j = 0 \text{ and } 0 \leq i \leq m - 1 \\ 1, & \text{if } j = 0 \text{ and } i = m \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let $(\pi_0, \pi_1, \ldots, \pi_m)$ denote the steady-state probabilities of the Markov chain. It can be solved by the following:

$$\begin{cases} \pi_j = \sum_{i=0}^{m} \pi_i P_{ij}, \quad \forall j \in \{0, 1, \ldots, m\} \\ \sum_{j=0}^{m} \pi_j = 1. \end{cases} \tag{3}$$

In particular, we can derive the close form of π_m as

$$\pi_m = \frac{p_{\text{ch}}^m - p_{\text{ch}}^{m+1}}{1 - p_{\text{ch}}^{m+1}}. \tag{4}$$

The detector will raise an alarm when it reaches state m. Under normal situations where the attacker is absent, if the detector reaches state m during the detection window, a false positive occurs. The false positive ratio (denoted as P_{fp}) of the proposed system is determined by the channel collision probability p_{ch}, the number of messages m in the proposed key exchange protocol, as well as the number of packet transmissions observed in the detection window. Assuming there are totally k transmissions observed in the detection window, then P_{fp} can be derived as

$$P_{\text{fp}} = k\pi_m = k \cdot \frac{p_{\text{ch}}^m - p_{\text{ch}}^{m+1}}{1 - p_{\text{ch}}^{m+1}}. \tag{5}$$

From (5), we can see that given k and p_{ch}, the false positive ratio of the proposed system monotonically decreases with m increases. So we can always choose a large enough m to reach a low target false positive ratio. It is worth noting that in implementation, p_{ch} and k will be estimated based on the transmission observed during the monitoring window t (step 2 of Fig. 7) and may not be the exact packet collision probability for the transmissions observed during the detection window. Besides, (5) is derived under the assumption that the collision probability for each observed transmission is independent. So when implementing our system, the value of m should be selected conservatively.

C. Cost Analysis

The cost introduced by the proposed solution has two aspects: the repeat transmissions of the DH protocol message introduce extra communication overhead; the shared key is considered to

TABLE I
SIMULATION SETUP

Slot time	$9\mu s$
DIFS/SIFS/ACK durations	$34\mu s$ / $18\mu s$ / $28\mu s$
Initial backoff window size	32
Maximum backoff stages	6
Maximum retry limit	7

be valid at the end of the detection window, which results in a delay to the key establishment process.

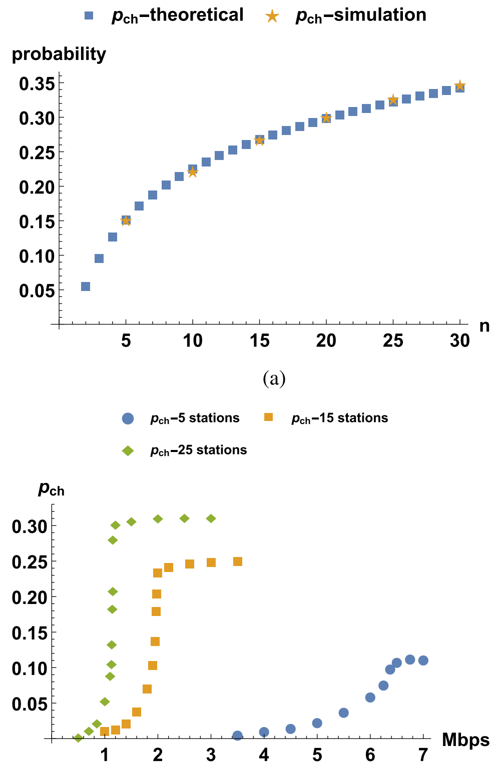
The proposed solution repeats the DH protocol m times, which has $2(m-1)$ more message transmission compared to the original DH protocol. Typically, the key agreement protocol is only used for initial trust establishment, and the subsequent key updates can be performed based on the existing shared secret. So the communication overhead of the proposed solution is in fact a one-time cost. From the numerical results in the next section, we can see that even at an extremely busy channel condition, the required value of m to achieve a 1% false positive ratio is no larger than 10.

The delay introduced by the proposed solution equals to the key exchange timer T consisting of the channel monitoring window t and the detection window $T-t$. The channel monitoring window is for Alice to estimate the channel collision probability and determine a value m for the key agreement protocol, and the detection window should be large enough to cover the $2\,m$ protocol message transmission. The value of T and t is preset by Alice. In practice, we recommend using $T=1.5\ s$ and $t=1\ s$, since $1\ s$ monitoring window gives Alice plenty of samples for channel condition estimation, and a detection window of $0.5s$ is enough for transmitting 26 maximum-sized protocol message with the proposed channel access mechanism under the lowest possible WiFi data rate (1 Mbps). The recommended setting introduces a $1.5s$ delay in the key establishment process, which is satisfactory considering that the required human interaction in existing OoB device pairing methods usually takes a couple of seconds.

VI. NUMERICAL AND SIMULATION RESULTS

In this section, we present numerical and simulation results of the proposed system. We first present the results of our collision detection algorithm, demonstrating that it can precisely detect whether an observed transmission is a collision or not. We then present the results of the false positive ratio, and show that the proposed system can achieve a low false positive ratio. In addition, we demonstrate how to configure the system parameters through a case study.

We use OMNeT++ to conduct the simulations, where Alice, Bob, and multiple background stations share an IEEE 802.11a wireless channel. The packet size of the background traffic is uniformly distributed between 500 Bytes and 2000 Bytes. The link layer parameters are set to be the default values of IEEE 802.11a standard. We summarize the key parameters used in the simulations in Table I.



(a)

(b)

Fig. 8. Channel collision probability. (a) Under saturated traffic. (b) Under unsaturated traffic.

A. Channel Collision Probability

The channel collision probability under saturated traffic condition can be explicitly analyzed using the Markov chain based models [37]–[39]. We can use equations (14) and (17) in [38] to calculate the channel access probability τ and the conditionally collision probability p. Then the channel collision probability p_{ch} can be derived as

$$p_{\text{ch}} = \frac{1-(1-\tau)^n - n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}. \tag{6}$$

As shown in Fig. 8(a), we numerically calculate p_{ch} with the number of stations varying from 2 to 30. We then implement the collision detection algorithm on a silent node, which only monitors the channel without transmitting or receiving any packets. This silent node makes decisions purely based on the channel occupancy pattern it observed, and count the number of collisions and successful transmissions within the simulation run time. We plot the channel collision probability obtained from the silent node in Fig. 8(a), where we can see that the p_{ch} obtained by our collision detection algorithm well matches the theoretical analysis.

We also implement the collision detection algorithm to obtain the channel collision probability p_{ch} under unsaturated traffic conditions. To simulate the unsaturated traffic, we implement a Poisson traffic generator on each background station. Fig. 8(b) shows the results p_{ch} under different traffic densities. As long as

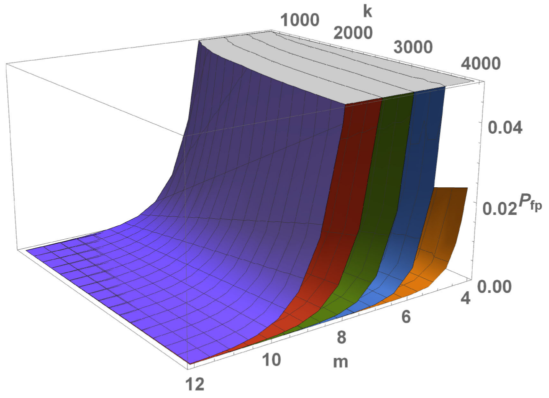


Fig. 9. False positive ratio.

the data rate of each station drops below a certain threshold, we can observe that p_{ch} drastically decreases to a low level.

B. False Positive Ratio

As we analyzed in Section V-B, the false positive ratio of the proposed system is affected by a number of variables, including the channel collision probability p_{ch}, the number of transmissions being monitored within the detection window, as well as the number of messages transmitted in the proposed key exchange protocol m. By (5), P_{fp} will increase with p_{ch} and k increasing, and with m decreasing, which matches the intuition that these trends will give the normal packet transmission a better chance to hit m collisions in a row. The numerical results of P_{fp} is presented in Fig. 9. The five surfaces from bottom to top represent the P_{fp} with p_{ch} being 5%, 10,% 15%, 20%, and 25%, respectively, while k varying from 1000 to 4000 and m ranging from 4 to 12. When p_{ch} equals 25%, which means the channel is operating in an extremely busy condition, the proposed system can achieve a false positive ratio of 1% if m is larger than 10.

We also conduct simulations to demonstrate the behavior of the proposed detector in terms of the false positive ratio under different monitoring window duration and different m. We simulate two scenarios with 5 and 12 background stations, under unsaturated and Poisson traffic arrivals with data rates equal to 1.54 Mbps and 1.17 Mbps, respectively. In the simulations, we also have another silent node to monitor the channel, record the consecutive collisions it observed and implement the proposed detector. First, if the detection window size increases, more collisions are expected to be observed and hence, with a higher probability, false alarms will be triggered during the window. Therefore, for a stable network when the attacker absents, the false positive rate increases as the detection windows increases, just as predicted by (5). These resutls are omitted. Second, simulation results on the false positive ratio P_{fp} under different m are plotted in Fig. 10, where the detection window is fixed at 1 s. We can see that the achieved P_{fp} decreases as m increases. In particular, for a bad selection of m, e.g., 1, P_{fp} becomes very large, due to that frequently happened collisions with background transmissions is mistakenly identified as MITM attacks. This suggests to choose m conservatively. In addition, the simulation results show that the actual false positive ratio is higher than the theoretical value calculated using (5). This is due to the fact that, under the unsaturated condition, most of the times only a few stations simultaneously have packets ready to transmit, resulting in a low average channel collision probability. However, when a collision happens, the involved stations will attempt to retransmit their packets, as a result, the number of stations competing for the next transmission is higher than average, which leads to a higher than average channel collision probability. While in our analysis we assume a time-independent packet collision probability which results in a lower false positive ratio.

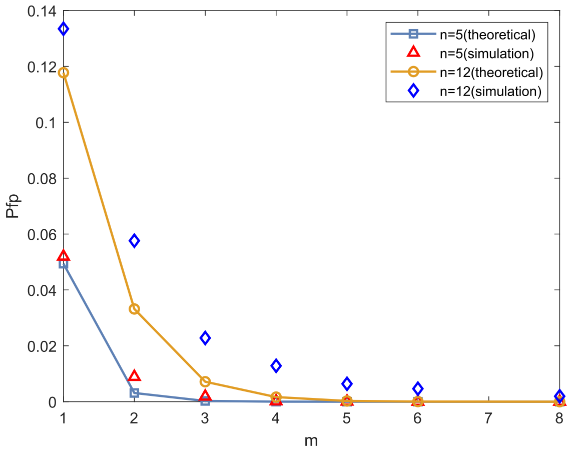


Fig. 10. Impact of differnt m on false positive ratio.

C. A Case Study

In this subsection, we present a simulation case to demonstrate how does Alice select the system parameter m based on the information she observed during the monitoring window, in order to achieve a target false positive ratio. In this simulation case, Alice, Bob, and other 10 background stations are sharing an IEEE 802.11a wireless channel. Each of the 10 background stations is generating packets with 2.0 Mbps data rate. The key exchange timeout T is set to be 1.5 seconds, and the duration of the monitoring window t is set to be 1 s. Based on the information observed during t, Alice will decide m, which is the number of messages to be transmitted in the proposed key exchange protocol, to reach a target P_{fp} of 0.5%.

After receiving the association reply from Bob, Alice starts to monitor the channel for $t = 1$ second. During this time, Alice observed 2065 transmission events, among them there are 1994 successful transmissions and 71 collisions (these numbers are obtained from a simulation case). Based on these values, Alice estimates the channel collision probability p_{ch} to be 3.44%, and the number of transmissions in the detection window ($T - t =$ 0.5 seconds) to be 1033. According to (5), if $m = 4$, P_{fp} is 1.36%, and if $m = 5$, P_{fp} is estimated to be 0.08%, which reaches the target false positive ratio. As we mentioned earlier, m should be selected conservatively. So Alice will set m to be 7 and starts the key exchange protocol.

We also conduct simulations for both normal and the MITM attack scenarios. We let Alice transmit 7 maximum-sized packets at the beginning of the detection window with her backoff counter fixed to 0. The attacker is set to transmit a jamming packet without channel sensing and backoff process, and the transmission of the attacker is triggered when Alice starts to transmit. In the normal case, during the 0.5 seconds detection window, Bob observes 41 collisions and 996 successful transmissions on the channel, and the maximum length of consecutive collisions observed is 2. In the attack case, Bob successfully detects 7 consecutive collisions at the very beginning of the detection window, which indicates the presence of the attacker.

VII. Conclusion

We systematically studied the wireless MITM attack, and modeled the attacker's behavior on message level. We then presented a novel in-band solution to detect the attacker during the key exchange process. The proposed scheme forces the attacker to generate a burst sequence of consecutive channel collisions, which can be detected by legitimate parties. We further presented analysis as well as simulation results to validate the performance of the proposed solution. Our solution achieves a missed detection ratio of 0, and can achieve a low false positive ratio by proper parameter design. A case study is also presented to demonstrate how to configure the system to achieve a guaranteed performance.

References

[1] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10626–10636, Dec. 2017.

[2] A. Hanyu *et al.*, "Adaptive frequency band and channel selection for simultaneous receiving and sending in multiband communication," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 460–463, Apr. 2019.

[3] Y. Qu *et al.*, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5171–5183, Jun. 2020.

[4] A. Hanyu, Y. Kawamoto, and N. Kato, "Adaptive channel selection and transmission timing control for simultaneous receiving and sending in relay-based UAV network," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2840–2849, Oct.–Dec. 2020.

[5] G. Chen, X. Cao, and J. Jin, "Joint scheduling and channel allocation for Kalman filtering over multihop WirelessHART networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3555–3565, May 2021.

[6] W. Shen, L. Liu, X. Cao, Y. Hao, and Y. Cheng, "Cooperative message authentication in vehicular cyber-physical systems," *IEEE Trans. Emerg. Top. Comput.*, vol. 1, no. 1, pp. 84–97, Jun. 2013.

[7] H. Tan, M. Ma, H. Labiod, A. Boudguiga, J. Zhang, and P. H. J. Chong, "A secure and authenticated key management protocol (SA-KMP) for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9570–9584, Dec. 2016.

[8] R. Ma, J. Cao, D. Feng, H. Li, and S. He, "FTGPHA: Fixed-trajectory group pre-handover authentication mechanism for mobile relays in 5G high-speed rail networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2126–2140, Feb. 2020.

[9] B. Mao, Y. Kawamoto, J. Liu, and N. Kato, "Harvesting and threat aware security configuration strategy for IEEE 802.15.4 based IoT networks," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 2130–2134, Nov. 2019.

[10] H. Li, Y. Yang, Y. Dai, S. Yu, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 484–494, Apr.–Jun. 2020.

[11] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[12] S. Mirzadeh, H. S. Cruickshank, and R. Tafazolli, "Secure device pairing: A survey," *IEEE Commun. Surv. Tut.*, vol. 16, no. 1, pp. 17–40, Jan.–Mar. 2014.

[13] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[14] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi, "Secure in-band wireless pairing," in *Proc. USENIX Secur. Symp.*, 2011, pp. 1–16.

[15] Y. Hou, M. Li, and J. D. Guttman, "Chorus: Scalable in-band trust establishment for multiple constrained devices over the insecure wireless channel," in *Proc. ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2013, pp. 167–178.

[16] W. Shen *et al.*, "Secure in-band bootstrapping for wireless personal area networks," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1385–1394, Dec. 2016.

[17] S. Čapkun, M. Čagalj, R. Rengaswamy, I. Tsigkogiannis, J.-P. Hubaux, and M. Srivastava, "Integrity codes: Message integrity protection and authentication over insecure channels," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 4, pp. 208–223, Oct.–Dec. 2008.

[18] Q. Hu, B. Du, K. Markantonakis, and G. P. Hancke, "A session hijacking attack against a device-assisted physical-layer key agreement," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 691–702, Jan. 2020.

[19] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for V2G using physical unclonable function," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7234–7246, Jul. 2020.

[20] J. Cao, M. Ma, and H. Li, "G2RHA: Group-to-route handover authentication scheme for mobile relays in LTE-A high-speed rail networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9689–9701, Nov. 2017.

[21] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, and P. Lorenz, "On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5535–5548, May 2020.

[22] P. Bagga, A. K. Das, M. Wazid, J. Rodrigues, K.-K. R. Choo, and Y. Park, "On the design of mutual authentication and key agreement protocol in internet of vehicles-enabled intelligent transportation system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1736–1751, Feb. 2021.

[23] K. Lee, V. Raghunathan, A. Raghunathan, and Y. Kim, "SYNCVIBE: Fast and secure device pairing through physical vibration on commodity smartphones," in *Proc. IEEE 36th Int. Conf. Comput. Des.*, 2018, pp. 234–241.

[24] D. Balfanz, G. Durfee, R. E. Grinter, D. K. Smetters, and P. Stewart, "Network-in-a-box: How to set up a secure wireless network in under a minute," in *Proc. USENIX Secur. Symp.*, 2004, pp. 207–222.

[25] V. Roth, W. Polak, E. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *Proc. ACM Conf. Wireless Netw. Secur.*, 2008, pp. 220–235.

[26] L. Wu, X. Du, W. Wang, and B. Lin, "An out-of-band authentication scheme for Internet of Things using blockchain technology," in *Proc. Int. Conf. Comput., Netw. Commun.*, 2018, pp. 769–773.

[27] C. Soriente, G. Tsudik, and E. Uzun, "HAPADEP: Human-assisted pure audio device pairing," in *Proc. Int. Conf. Inf. Secur.*, 2008, pp. 385–400.

[28] Q. Hu, Y. Liu, A. Yang, and G. Hancke, "Preventing overshadowing attacks in self-jamming audio channels," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 45–57, Jan./Feb. 2021.

[29] M. Sethi, M. Antikainen, and T. Aura, "Commitment-based device pairing with synchronized drawing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2014, pp. 181–189.

[30] D. Schürmann and S. Sigg, "Secure communication based on ambient audio," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 358–370, Feb. 2013.

[31] T. Bui and T. Aura, "Key exchange with the help of a public ledger," in *Cambridge International Workshop on Security Protocols*. Berlin, Germany: Springer, 2017, pp. 123–136.

[32] N. O. Tippenhauer, K. B. Rasmussen, and S. Capkun, "Physical-layer integrity for wireless messages," *Comput. Netw.*, vol. 109, pp. 31–38, 2016.

[33] N. Ghose, L. Lazos, and M. Li, "Secure device bootstrapping without secrets resistant to signal manipulation attacks," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 819–835.

[34] S. N. Premnath *et al.*, "Secret key extraction from wireless signal strength in real environments," *IEEE Trans. Mobile Comput.*, vol. 12, no. 5, pp. 917–930, May 2013.

[35] L. Wang and A. M. Wyglinski, "Detection of man-in-the-middle attacks using physical layer wireless security techniques," *Wireless Commun. Mobile Comput.*, vol. 16, no. 4, pp. 408–426, 2016.

[36] C. Pöpper, N. O. Tippenhauer, B. Danev, and S. Capkun, "Investigation of signal and message manipulations on the wireless channel," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2011, pp. 40–59.

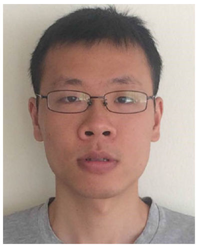
[37] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[38] H. Zhai, Y. Kwon, and Y. Fang, "Performance analysis of IEEE 802.11 MAC protocols in wireless LANs," *Wireless Commun. Mobile Comput.*, vol. 4, no. 8, pp. 917–931, 2004.

[39] X. Cao, L. Liu, W. Shen, A. Laha, J. Tang, and Y. Cheng, "Real-time misbehavior detection and mitigation in cyber-physical systems over WLANs," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 186–197, Feb. 2017.

Wenlong Shen (Student Member, IEEE) received the Ph.D. degree in computer engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2018. He is currently a Data Scientist with Comcast Corporation. His research interests include network security, wireless networks, Big Data, and machine learning based network automation and monitoring.

Yu Cheng (Senior Member, IEEE) received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2003. He is currently a Full Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. His research interests include wireless network performance analysis, information freshness, machine learning, and network security. He was the recipient of the Best Paper Award at QShine 2007, IEEE ICC 2011, and a Runner-Up Best Paper Award at ACM MobiHoc 2014. He was the recipient of the the National Science Foundation (NSF) CAREER Award in 2011 and IIT Sigma Xi Research Award in the junior faculty division in 2013. He has served as Symposium Co-Chairs for IEEE ICC and IEEE GLOBECOM, and Technical Program Committee (TPC) Co-Chair for IEEE/CIC ICCC 2015, ICNC 2015, and WASA 2011. He was the Founding Vice Chair of the IEEE ComSoc Technical Subcommittee on Green Communications and Computing. He was an IEEE ComSoc distinguished Lecturer in 2016–2017. He is an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, and IEEE WIRELESS COMMUNICATIONS.



Bo Yin (Student Member, IEEE) received the B.E. degree in electronic information engineering and the M.E. degree in electronic science and technology from Beihang University, Beijing, China, in 2010 and 2013, respectively, and the Ph.D. degree in computer engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2020. His research interests include network resource allocation, Age-of-Information, and ML-based network optimization.

Jin Du (Student Member, IEEE) received the B.E. degree from the School of Automation, China University of Geosciences, Wuhan, China, in 2019. He is currently working toward the master's degree with the School of Automation, Southeast University, Nanjing, China. His research interest focuses on industrial wireless networking.

Xianghui Cao (Senior Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively. From 2012 to 2015, he was a Senior Research Associate with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. He is currently a Professor with the School of Automation, Southeast University, Nanjing, China. His current research interests include cyber-physical systems, wireless network performance analysis, wireless networked control, and network security. He was the recipient of the Best Paper Runner-Up Award from ACM MobiHoc in 2014 and the First Prize of Natural Science Award of Ministry of Education of China in 2017. He is also an Associate Editor for *ACTA Automatica Sinica and IEEE/CAA Journal of Automatica Sinica*.