Secure Authentication in Cross-Technology Communication for Heterogeneous IoT

Sihan Yu, Xiaonan Zhang, Pei Huang, and Linke Guo
Department of Electrical and Computer Engineering, Clemson University
{sihany, xiaonaz, peih, linkeg}@g.clemson.edu

***Abstract*—The emerging Cross-Technology Communication (CTC) has enabled the direct communication among different protocols, which will greatly enhance the spectrum efficiency. However, it will also bring security challenges to end IoT devices since the attacks can be from heterogeneous devices. Current deployed security mechanisms cannot be applied among heterogeneous devices. This work proposes a new mechanism to verify the legitimacy of signal source so that only the signals from legal CTC devices can be further processed. We verify the legitimacy of devices by embedding authorization codes into the packets at the sender side and verify them at the receiver side. Theoretical analysis and experiments show that this mechanism can provide effective protection on heterogeneous communication pairs.**

***Index Terms*—Cross-Technology Communication, physical layer security, device authentication**

I. Introduction

The wide deployment of the Internet of Things (IoT) has caused serious problems of wireless spectrum scarcity [1]. To solve this problem, Cross-Technology Communication (CTC) was proposed to support direct communication among devices with different wireless protocols (e.g., WiFi, Bluetooth, and ZigBee) [2]. Different from the existing indirect methods such as deploying a multi-protocol gateway, CTC can save the deployment cost and reduce the number of wireless transmission. However, the use of CTC also brings some potential security risks. For example, a ZigBee smart lock may receive commands (`LOCKING/UNLOCKING`) from various kinds of devices, including the authorized ZigBee gateway, some legal smartphones or other illegal WiFi devices. As a result, this new paradigm provides opportunities for malicious WiFi devices to manipulate the ZigBee smart lock. Since both of the legal and illegal devices use the same command, how to differentiate the legitimacy of received signals becomes a challenging problem.

Most existing security mechanisms (such as [3], [4]) cannot differentiate the source of received packets when they have the same content. They can only use the timestamp to prevent the replay attack. In this poster, we propose a physical layer security mechanism to provide device authentication between WiFi and ZigBee devices under the condition that allowing replay. Our idea is to embed an authorization code (AC) into the packet at the sender side and verify it at the receiver side. The embedded AC will change over time, making attackers unable to predict or reuse the overheard AC for attacking purposes.

II. System Design

A. System Overview

Our designed scheme uses a hash function to generate a chain of ACs [5], which will be known only by both the legal CTC device and the ZigBee receiver. Each time, an AC is embedded in the preamble (i.e., "`00000000A7`", as shown in Fig. 1) of a ZigBee packet and sent by a legal CTC device. If the receiver finds that the received AC is correct, the packet will be regarded as from a legal CTC device. According to the Direct Sequence Spread Spectrum (DSSS) technique adopted by ZigBee, a symbol is further represented by 32 chips. If we pick out some positions (e.g., the yellow chips in Fig. 1) to embed our AC, these 32 chips can still be correctly decoded because of the fault tolerance of DSSS.

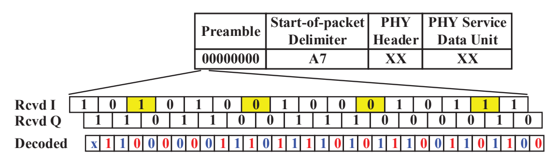


Fig. 1. Format of ZigBee Packet

B. Authorization Code Generation

To generate a chain of ACs, we select a random number n_r and deliver it to the legal CTC device and ZigBee receiver in a secure way (e.g., input it manually by the user). Then, they recursively computes $n_i = h(n_{i+1})$ to get the AC chain $\{n_1, ..., n_{r-1}\}$, in which $i \in [1, r-1]$ and $h(\cdot)$ denotes the cryptographic one-way hash function such as SHA-1. Finally, the legal CTC device uses n_i as the AC of the i-th transmission. Because the order of generation and usage of the ACs are different, even if the attacker can overhear the current AC, it cannot derive the next available value.

C. Authorization Code Encoding

Our AC encoding mechanism is inspired by the ZigBee decoding mechanism. A received symbol consists of in-phase and quadrature parts, which have an offset of half chip. The even and odd chips will be decoded as $I \oplus Q$ and $\overline{I \oplus Q}$, respectively, shown as the blue chips and red chips in Fig. 1. According to this characteristic, we can find that if we flip (turn 1 to 0 or turn 0 to 1) I and Q simultaneously, the decoding result does not change. Therefore, we decide to pick out some positions from the signal I to embed the AC and flip some adjacent chips to ensure the decoding result does not have too many errors. Fig. 2 shows an example of how to embed AC and reduce chip error. By flipping some adjacent chips, we can push the wrong chips to any position. If two wrong chips have been pushed to the same position, the number of wrong chips is reduced.

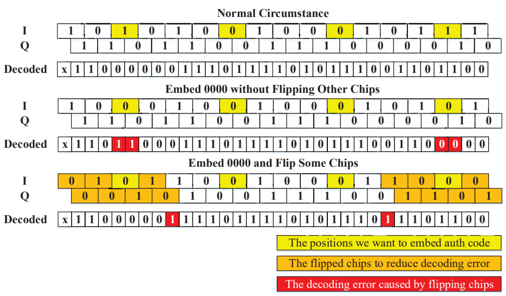


Fig. 2. Auth Code Embedding

D. Authorization Code Decoding

At the receiver side, the ZigBee device only needs to extract the chip value from predefined positions. After collecting a complete AC, it can verify whether this value is correct. Because of the channel condition may be not good, we do not require the value to be completely correct. We can set a threshold, e.g., as long as 80% of the bits are correct, we regard it as a correct AC.

III. Performance Evaluation

In this section, we mainly evaluate the following two performance. One is whether the AC can be extracted accurately at the receiver side, especially when the channel condition is not good. The other is whether embedding AC will bring a bad effect on data accuracy. We conduct both simulation and field experiments to evaluate the performance, in which simulation is based on GNU Radio and field experiments are based on USRP and TI CC26X2R1 launchpad. Each time we send 100 WiFi-emulated ZigBee packets, each packet includes 20 symbols. Then we test various kinds of error rate at the receiver side.

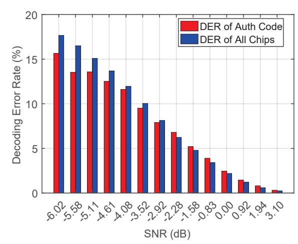
A. Decoding Accuracy of Authorization Code

Fig. 3(a) shows the decoding error rate (DER) of the AC with different SNRs. It can be seen that the DER of AC has no significant difference with that of general chips. In other words, as long as the packet can be received accurately, the AC can be extracted accurately.

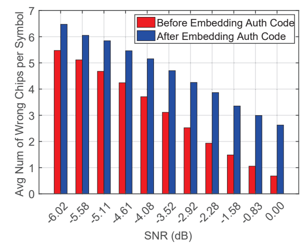
B. Bad Effects of Authorization Code

Fig. 3(b) to Fig. 3(d) shows whether embedding AC will bring some bad effects to data accuracy. From Fig. 3(b), we can find that it does increase the chip error rate (CER), but the increase is steady and bounded, which mainly depends on how many chips have been modified. In this example, in order to embed AC, there are averagely two unavoidable wrong chips in each symbol. Therefore, after embedding AC, the average number of wrong chips in each symbol is approximately 2 chips larger than before embedding.

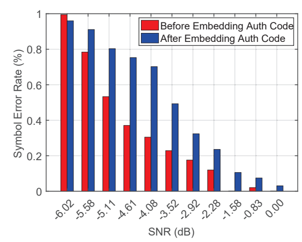
Fig. 3(c) and Fig. 3(d) show the variation of symbol error rate (SER) and packet error rate (PER) caused by embedding AC. It can be seen that it does increase the error rate, especially when SNRs are relatively low. However, when the SNRs are greater than 0 (we did not show them in these figure because their values are very close to 0), they do not have a significant difference, indicating the error is acceptable.



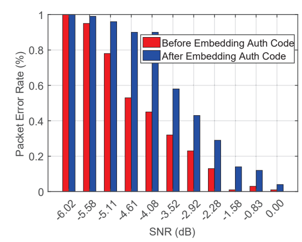
(a) Decoding Error Rate



(b) Chip Error Rate



(c) Symbol Error Rate



(d) Packet Error Rate

Fig. 3. Decoding Accuracy Evaluation

IV. Conclusion

In this poster, we propose a physical layer security mechanism to verify the legitimacy of the signal source for heterogeneous IoT. It verifies the legitimacy by checking the AC that embedded in the packet. Experiment results demonstrate that the receiver can recognize the embedded AC accurately while maintaining the normal communication.

V. Acknowledgment

The work of L. Guo is partially supported by National Science Foundation (NSF) under grant CNS-1947065.

References

[1] statista. statista report. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 2–14.

[3] N. F. Pub, "197: Advanced encryption standard (aes)," *Federal information processing standards publication*, vol. 197, no. 441, p. 0311, 2001.

[4] X. Zhang, P. Huang, L. Guo, and Y. Fang, "Hide and seek: Waveform emulation attack and defense in cross-technology communication," in *Proceedings of the 39th Annual International Conference on Distributed Computing Systems*. IEEE, 2019, pp. 1–10.

[5] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Safedsa: Safeguard dynamic spectrum access against fake secondary users," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 304–315.