Postprint: Improved Wireless RFID Key Generation Algorithm Based on Activation Flag Bit

Authors: Yang Jun, Zou Zhige

Date: 2018-10-11T00:00:00+00:00

Abstract

To address the security issues present in corresponding key generation algorithms designed for three common Radio Frequency Identification (RFID) over-the-air key generation scenarios—including lack of algorithmic theoretical proof, replay attacks, key forgery attacks, and RFID tag identity ID leakage—this work proposes a more secure improved RFID system over-the-air key generation algorithm based on activation flag bits. The improved algorithm constructs a secure algorithmic framework solely through a combination of multiple ultra-lightweight bitwise operations, thereby reducing cost and improving efficiency; it utilizes an activation flag bit AckBit mechanism along with a freshness mechanism to resist replay and key forgery attacks; and it demonstrates the security feasibility of the target algorithm through a complete GNY logic proof process and security comparative analysis. Finally, a tag cost comparison between the original algorithm and the improved algorithm is presented, showing that the improved algorithm achieves higher security while satisfying low-cost constraints.

Full Text

Preamble

Improved RFID Key Wireless Generation Algorithm Based on Activation Flag

Yang Jun[1], Zou Zhige[2]
(1. Wuhan Polytechnic, Wuhan 430074, China; 2. Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract: In the context of three common wireless key generation scenarios for radio frequency identification (RFID) systems, existing key generation algorithms suffer from several security vulnerabilities, including lack of formal theoretical proof, susceptibility to replay attacks, key forgery attacks, and leakage of RFID tag identity IDs. This paper proposes a more secure improved

RFID system key wireless generation algorithm based on an activation flag mechanism. The improved algorithm constructs a secure framework using only ultra-lightweight bitwise operations, thereby reducing costs and improving efficiency. It employs an activation flag AckBit mechanism combined with freshness mechanisms to resist replay and key forgery attacks. Through a complete GNY logic proof process and comparative security analysis, we demonstrate the security and feasibility of the proposed algorithm. Finally, a cost comparison between the original and improved algorithms shows that the improved algorithm achieves higher security while maintaining low-cost requirements.

Keywords: RFID; wireless key generation; more secure; activation flag; GNY logic proof

0 Introduction

With the continuous development of the Internet of Things, radio frequency identification (RFID) technology, as a crucial component of its perception layer, has attracted increasing attention. In RFID systems, tags and readers typically need to perform mutual authentication, identification, and localization operations, all of which require the participation of shared keys. Key distribution generally follows two approaches: either the keys are pre-configured at the factory with fixed values, or they are dynamically generated wirelessly at both the reader and tag ends using cryptographic algorithms. The latter approach offers dynamic and convenient advantages and has become a research hotspot in recent years. However, precisely because of this dynamic openness in RFID systems, the rapid, secure, and accurate generation of shared keys has emerged as a critical challenge for RFID key wireless generation algorithms. In recent years, ultra-lightweight bitwise operation-based key wireless generation algorithms have been widely studied due to their dynamic unpredictability, low cost, and reliable security.

1 Related Research

Literature [5] first proposed the concept of RFID key wireless generation, breaking away from the limitations of pre-established keys. Based on the assumption that "backward channels cannot be eavesdropped," it introduced the "WiKey" RFID key wireless generation algorithm. However, subsequent works have pointed out the practical limitations of WiKey' s assumptions, primarily noting its vulnerability to key forgery attacks due to all communication data being transmitted in plaintext, and have proposed various improved algorithms [6-9].

Literature [6] proposed an RFID key generation algorithm based on multiple bitwise operations (word-composite bitwise operations, crossover bitwise operations, XOR, and AND operations). While this approach improved authentication efficiency by utilizing tag ID information in communication, it risked

exposing tag identity IDs in single-tag key generation scenarios. Additionally, since each tag must generate random numbers for authentication calculations, system cost and complexity increase, making it unsuitable for low-cost applications.

Literature [7] also proposed a new key generation algorithm that addressed WiKey' s security flaws, but its authentication process required two rounds of communication, resulting in low efficiency.

Literature [8] presented an improved algorithm based on partial tag ID. This algorithm, which only introduces tag ID information and random numbers, has low cost and is suitable for low-cost application environments. However, our research reveals that in group tag generation scenarios, it cannot resist eavesdropping or replay attacks, leading to tag identity leakage and key forgery. Specific vulnerability analyses are as follows:

Tag Identity Leakage Vulnerability: During communication, an attacker A can eavesdrop on the second step where group tags Ti obtain all tags' Mi values. Continuing to eavesdrop on the third step where the reader sends plaintext to all tags, attacker A can decrypt to obtain . Since the attacker has already obtained the left half of the plaintext tag ID through eavesdropping, the complete tag identity ID information is compromised, resulting in tag identity leakage.

Key Forgery Vulnerability: Because the algorithm design lacks freshness verification for random numbers, attacker A can impersonate a tag by replaying Mi or recalculating Mi based on the decrypted . This allows the attacker to pass the database' s legitimacy verification and obtain the key calculation factor ki, then decrypt to get the key k, or directly calculate the key k using all obtained tag ID information. Therefore, this algorithm cannot resist replay attacks and suffers from key forgery and tag identity leakage risks.

Literature [9] proposed an encrypted RFID key generation algorithm based on pseudonym flags. While introducing pseudonym flags can prevent tag identity leakage to some extent, our research shows that this algorithm still suffers from key forgery attack vulnerabilities in single-tag key generation scenarios. When the reader sends encrypted messages to the tag, an attacker can directly eavesdrop to obtain messages A and B, then use these two data values themselves to crack and calculate the algorithm' s key information k. Thus, the attacker can quickly brute-force the system key information k based on the algorithm' s design vulnerabilities, making the protocol vulnerable to key forgery attacks.

In summary, existing RFID system key wireless generation algorithms have the following drawbacks, as shown in Table 1 .

Table 1. Vulnerability Classification of Existing Protocols

Existing Protocol Defects	Literature
Lack of formal proof	[5, 6, 7, 8, 10]
Tag identity ID leakage	[5, 8, 9]

Existing Protocol Defects	Literature
Key forgery attack	[5, 6, 8, 9, 10]

2 Improved More Secure RFID System Key Wireless Generation Algorithm

2.1 Prerequisites and Symbol Definitions

Similar to general RFID system key wireless generation algorithms, the back-end database and reader are considered a unified secure entity (hereinafter collectively referred to as the reader), while the communication channel between reader and tag is considered insecure [11-12]. Therefore, encrypted transmission must be designed into the protocol to ensure security. The basic principle of the self-composite crossover bitwise operation Sac(X,Y) can be found in literature [13]. The symbols used in this algorithm are defined in Table 2 .

Table 2. Symbol Definitions

Symbol	Definition
TID_L, TID_R	Left and right parts of tag pseudonym identity flag
k	Shared key between tag and reader
AckBit	Activation flag, AckBit=1 means tag activated; AckBit=0 means tag not activated
r1, r2	Random numbers between tag and reader
rl, rr	Left and right parts of random numbers
A, B, C, D	Intermediate calculation values
Sac(X,Y)	Self-composite crossover bitwise operation
Rot(X,Y)	Cyclic shift operation

2.2 Algorithm Details

2.2.1 Multi-Tag Batch Key Wireless Generation Scenario The detailed process for the reader generating individual keys for multiple tags is as follows, as shown in Figure 2 [Figure 2: see original paper]. The multi-key generation process is similar to single-key generation, with the following differences:

a) Multiple tags Tag1,⋯,Tagi send key generation request commands along with their index i: <Rq,1,4,i,⋯>, then wait for reader response.

b) The reader responds sequentially within a given time frame and queues tags in order of index i. All tags Ti complete the key generation process through steps c)~f). If timeout or authentication failure occurs, the protocol terminates. If the database successfully generates keys synchronously, proceed to step f).

c) The reader determines whether to continue based on the tag' s index i and activation flag AckBit. If index i is found and AckBit=0, the algorithm continues. The reader retrieves the stored data (i, IDi, TIDi_L, TIDi_R, AckBit), computes values, and generates two random numbers r1 and r2. Finally, it sends A and B to tag Tag. If index i is found but AckBit=1, indicating the tag key has already been generated, the algorithm terminates due to duplicate request. If index i is not found, the algorithm terminates.

d) After receiving data A and B, tag Tag uses its stored pseudonym information to decrypt A and B to obtain random numbers r1 and r2. The decryption process is as follows: . After successful decryption, the tag generates verification data C and sends it to the reader.

e) After receiving message C, the reader computes C' using the previously generated r1, r2, and TIDi information, then verifies whether C equals C' . If they match, the reader successfully authenticates the tag as legitimate and sends the key generation command Create to the tag. Simultaneously, the database synchronously generates key Keyi for the tag and updates the stored fields to (i, ID, TIDi_L, TIDi_R, Keyi, AckBit=1).

f) After receiving the key generation command Create, tag Tag generates key Keyi: . The tag' s stored fields are updated to .

2.2.2 Single-Tag Key Wireless Generation Scenario The algorithm for a reader generating an individual key for a single tag is a special case of the multi-tag process and will not be repeated here. The single-tag process diagram is shown in Figure 1 [Figure 1: see original paper].

2.2.3 Group Tag Group Key Wireless Generation Scenario The detailed process for the reader generating a unified group key for group tags is as follows, as shown in Figure 3 [Figure 3: see original paper].

a) The reader first broadcasts to the group tags, issuing the key generation command.

b) After receiving command Rq, all tags in the group use their pseudonym information to compute verification data A1, A2,⋯,Ai and send all Ai values to the reader: .

c) After receiving all Ai information, the reader checks within a given time frame whether all tags in its group records have corresponding Ai messages that satisfy . If all records match and are equal, indicating that all tags in the group can be found in the reader' s database, all group tags are successfully activated. The reader updates its stored field information to (i, ID, TIDi_L, TIDi_R, AckBit=1). If any tag record in the reader' s group records does not correspond to the Ai message sent by tags, group tag activation fails. The reader initiates a new round of activation within the given time until all group tag records match. If timeout occurs, the

algorithm terminates. When all group tags are successfully activated, the reader generates random number r and begins generating a shared group key Key for the group tags. It computes an encrypted group key calculation factor Ki for each tag in the group: , and sends message <Ki, Bi, Ci, Di> to the tags.

d) After receiving message <Ki, Bi, Ci, Di>, each tag in the group first decrypts to obtain random number r based on its stored fields: . The tag then computes D' and verifies whether it equals the received D. If they match, the tag successfully authenticates the reader. Each tag in the group uses the decrypted r and its own identity ID information to compute the group key Keyi: . Tags then send confirmation Si to the reader sequentially: .

e) The reader must receive all Si confirmation commands from the group within the specified time and verify whether all group keys computed by tags are equal. The reader decrypts Si based on stored tag fields (i, ID, TIDi_L, TIDi_R, AckBit=1). If all decrypted keys are equal, indicating that all tags in the group have successfully obtained the group key, the reader updates its stored field information to (i, ID, TIDi_L, TIDi_R, Key, AckBit=1) and broadcasts an update reply command to the tag group. If any Keyi is not equal, indicating a group key factor error, the algorithm terminates.

f) After receiving the update command, each tag in the group updates its stored fields to , completing the algorithm.

In summary, this section designs improved, more secure RFID system key wireless generation algorithms applicable to three different scenarios: single-tag key generation, multi-tag key generation, and group tag key generation. The algorithm implements tag activation and encrypted authentication based on cyclic shift, XOR, and self-composite crossover bitwise operations. In group key generation scenarios, random numbers are introduced to ensure key freshness and prevent replay attacks. Throughout the communication process, the tag' s real identity ID remains anonymously hidden, using only the pseudonym TID for calculations to prevent identity leakage. The correspondence between tag index i and the AckBit activation flag marks tag status information, improving algorithm efficiency.

3 Algorithm Formal Proof

GNY logic is a knowledge and belief-based logical reasoning method widely applied in RFID algorithm proofs. GNY logic consists of several axioms and rules; its basic definitions can be found in literature [14,15]. The following rules are primarily used in our proof process:

- **Recognizability Rules:** R1=, R2=
- **Freshness Rule:** F1=

- **Message Parsing Rule:** I1=

3.1 Initialization Assumptions

First, we establish initialization assumptions for the single-tag key generation algorithm. H1 and H2 represent information already possessed by the tag and reader; H3 and H4 represent what the tag and reader believe they can recognize; H5 and H6 represent the tag and reader' s beliefs about the freshness of certain data.

3.2 Formal Model

We construct a GNY logic formal model for the single-tag key generation algorithm to completely simulate the algorithm' s interaction process. M represents the communication process, Tag represents the tag entity, and Reader represents the reader entity.

3.3 Security Objectives

We define the security objectives that the single-tag key generation algorithm must achieve using GNY logic language. D1 represents the reader' s initial identification of the tag identity; D2 and D3 represent the tag' s verification of the reader and belief in the freshness of sent information; D4 represents the reader' s verification of the tag and re-identification of the tag identity.

3.4 Proof Process

When the reader receives message M1, i.e., , the reader searches the backend database records (i, ID, TIDi_L, TIDi_R, AckBit) for matching tag index information i. If the search is successful, the reader achieves initial tag identification, and security objective D1 is proven.

When the tag receives message M2, i.e., , from initialization assumption H5: , and from H1: , and according to freshness rule F1, we obtain . Continuing with freshness rule F1, we get . Then according to message parsing rule I1 (where), security objective D2 is proven.

Security objective D3: From initialization assumption H5: , and from H1: , and using recognizability rules R1 and R2, we obtain . From freshness rule F1, we get . Continuing with freshness rule F1, we obtain . Then according to message parsing rule I1 (where), security objective D3 is proven.

When the reader receives message M3 from the tag, i.e., , the reader again searches the backend database records (i, ID, TIDi_L, TIDi_R, AckBit) and random numbers r1, r2 to verify the tag, obtaining . Finally, according to recognizability rule R1: , we get , achieving expected objective D4.

Because the group key generation scenario for batch tags and group tags cannot be compared due to different numbers of tags, this analysis focuses on the single-

tag scenario, comparing the improved algorithm with related literature in terms of tag computational cost, storage cost, and communication cost. As shown in Table 4 , Xt represents XOR operations, PRNG represents pseudo-random number generators, R represents shift operations, Rt represents cyclic shift operations, Sa represents self-composite crossover bitwise operations, pseudonym TID, tag unique identity ID, key Key, and random numbers all have length I, X represents temporary storage space during authentication, and communication cost is measured in L.

Table 4. Performance Comparison of Single-Tag Key Generation Algorithms

Metric	Literature [5]	Literature [8]	Literature [9]	Our Algorithm
Computational Cost	3Xt+1PRNG	5Xt+R	4Xt+2Rt+2Sa	4Xt+2Rt+2Sa
Storage Cost	4I+X	4I+X	5I+X	5I+X
Communication Cost	3L	3L	3L	3L

As shown in Tables 3 and 4 and Figure 4 [Figure 4: see original paper], for single-tag key generation, compared with literature [5, 8, 9], our algorithm achieves ultra-lightweight standards similar to the comparison algorithms in terms of tag computational cost, storage space, and communication cost. However, our improved algorithm provides higher security while maintaining comparable tag-side computational costs that remain below the average, satisfying low-cost requirements.

6 Conclusion

This paper proposes more secure improved algorithms for three RFID key wireless generation scenarios, providing a complete GNY logic proof process that demonstrates algorithm security and feasibility. While satisfying low-cost RFID system applications, the improved algorithm addresses the original algorithms' security deficiencies in three aspects: tag anonymity, replay attacks, and key forgery attacks. Compared with literature [5, 8, 9], the improved algorithm achieves higher security without increasing tag-side computational costs, making it suitable for low-cost RFID applications.

References

[1] Shen J, Tan H, Zhang Y, et al. A new lightweight RFID grouping authentication protocol for multiple tags in mobile environment [J]. Multimedia Tools & Applications, 2017: 1-23.

[2] Wang Jinru. Wireless key generation algorithm for RFID system based on partial pseudonym ID [J]. Computer Engineering and Applications, 2018, 54(1): 128-132.

[3] Chen H, Wang Z, Xia F, et al. Efficiently and completely identifying missing key tags for anonymous RFID systems [J]. IEEE Internet of Things Journal, 2017, PP(99): 1-1.

[4] Zhang Bing, Qin Zhiguang, Wan Guogen. Study on hybrid key management mechanisms of RFID system based on PKI and CPK [J]. Journal of University of Electronic Science and Technology of China, 2015, 44(3): 415-421.

[5] Lu Li. Wireless key generation for RFID systems [J]. Chinese Journal of Computers, 2015, 38(4): 822-832.

[6] Jian Biyuan, Liu Daowei. Wireless key generation algorithm for RFID system based on bit operation [J]. Computer Engineering and Applications, 2017, 53(16): 98-103.

[7] Si Jin, Jian Biyuan, Liu Daowei. Wireless key generation algorithm for RFID system [J]. Computer Engineering and Design, 2017, 38(10): 2686-2690.

[8] Huang Qi, Ling Jie, He Xiaotao. Improved RFID key wireless generation algorithm based on tag part ID [J]. Computer Science, 2017, 44(1): 172-175.

[9] Su Qing, Li Qian, Peng Jiajin, et al. Wireless key generation protocol for encrypted RFID system based on pseudonym logo [J]. Computer Engineering, 2017, 43(8): 173-177.

[10] Zhang Zhaohui, Liu Yue, Liu Daowei. Based on tag' s ID wireless key generation for RFID system [J]. Application Research of Computers, 2017, 34(1): 261-263.

[11] Labbi Z, Maarof A, Senhadji M, et al. Hybrid encryption approach using dynamic key generation and symmetric key algorithm for RFID systems [C]// Proc of International Conference on Networked Systems. Springer International Publishing, 2016: 244-249.

[12] Dimitriou T. Key evolving RFID systems: Forward/backward privacy and ownership transfer of RFID tags [J]. Ad Hoc Networks, 2016, 37: 195-208.

[13] Wang Xiaowei, Lu Zhixiang, Lu Tao. Ultra-lightweight mobile authentication protocol based on self-assembly crossover [J]. Computer Engineering and Design, 2017(12): 3252-3257.

[14] Garcia R, Modesti P. An IDE for the design, verification and implementation of security protocols [C]// Proc of IEEE International Symposium on Software Reliability Engineering. Washington DC: IEEE Computer Society, 2017: 157-163.

[15] Zhu Hongfeng, Liu Tianhua. Research on privacy protection security protocol [M]. Beijing: Science Press, 2015.