A PUF-based Lightweight and Secure Mutual Authentication Mechanism for Remote Keyless Entry Systems

Rohini Poolat Parameswarath
Department of Electrical and Computer Engineering
College of Design and Engineering
National University of Singapore
Singapore
rohini.p@nus.edu.sg

Biplab Sikdar
Department of Electrical and Computer Engineering
College of Design and Engineering
National University of Singapore
Singapore
bsikdar@nus.edu.sg

***Abstract*—Keyless entry systems in cars give users the flexibility of unlocking the car door without using physical keys. Remote Keyless Entry (RKE) system is one of the common types of keyless entry systems. RKE systems that use a fixed code to unlock the car are susceptible to replay attacks. RKE systems that use rolling codes instead of fixed codes are still vulnerable to RollJam attacks. To protect RKE systems from such attacks, we propose a lightweight and secure mutual authentication mechanism based on Physical Unclonable Functions (PUFs). We provide formal security analysis to show that the proposed mechanism is resilient against several common attacks. We also provide a comparison with other existing schemes which shows that the proposed mechanism is very efficient in terms of computation cost.**

***Index Terms*—Mutual authentication, Remote Keyless Entry (RKE) system, Replay attack, RollJam attack, Physical Unclonable Function (PUF)**

I. Introduction

Over the years, various automotive electronics components have been added to cars to make them more user-friendly. While the added systems improve users' convenience, they have also opened up new attack surfaces [1]. To name a few, Electronic Control Units (ECUs), radio channels, keyless entry, Bluetooth, and On-Board Diagnostic (OBD) ports are some of the attack surfaces in modern cars through which an attacker can mount attacks. The automotive attacks can be physical or remote [1]. The attacks on keyless entry systems do not require physical access and can be carried out using relatively inexpensive hardware and software. As a result, keyless entry systems have been always targeted by attackers, and threats against keyless entry systems are real. This paper investigates security issues with keyless entry systems and presents a solution to make them secure from attacks.

Keyless entry systems enable users to lock and unlock their cars without requiring physical keys [2]. There are mainly two types of keyless entry systems: Remote Keyless Entry (RKE) systems and Passive Keyless Entry and Start (PKES) systems. To unlock or lock the car door, the user presses a button on the key fob in the RKE systems. The PKES systems enable unlocking the car door when the key fob is in close proximity to the car. Several attacks have been reported against the RKE and the PKES systems [2]–[5]. In this paper, we focus on the RKE systems.

978-1-6654-3540-6/22

In the RKE systems, when the user presses a button on the key fob, radio frequency (RF) signals are generated from the key fob and sent to a receiver in the car. The RF signals generated are in the 315 MHz, 433 MHz, or 868 MHz band, depending on the geographic location of the car. The frequency band used in North America is 315 MHz. In Europe, 433 MHz or 868 MHz bands are used [4]. The first generations of the RKE systems employed a static code in the RF signals. An adversary may carry out replay attacks by capturing and replaying such signals to gain access to the car. Such replay attacks do not require much effort from an adversary. To prevent replay attacks, instead of using static codes, rolling codes were introduced. Each time the unlock button on the key fob is pressed, a new rolling code is produced. The generated rolling code can be used to unlock the car only once [4]. Microchip Technology's KeeLoq [6] and NXP's Hitag-2 [7] rolling code schemes are popular rolling code schemes. However, there have been attempts to break the security of even the RKE systems based on rolling codes. The RollJam attack [8] has shown that even the RKE systems based on rolling codes can be compromised using special types of equipment. In this paper, we propose an authentication mechanism to protect RKE systems from such attacks. The proposed authentication mechanism is based on Physical Unclonable Functions (PUFs).

PUF is a hardware security primitive that makes use of the randomness introduced during the manufacturing process to derive secrets [9]. PUFs map input challenges to responses and do not store the responses in memory. Hence, PUFs have been used in security applications in different domains.

A. Related Work

Several studies have highlighted the security issues that exist in keyless systems. Different attacks against keyless entry systems are presented in [3]. The authors of [4] presented attacks that can be used against rolling code-based RKE systems. The authors of [5] analyzed the Hitag-2 algorithm-based RKE and showed that keys can be extracted from Hitag-2. Different attack surfaces in cars are discussed in [10]. The authors of [11] surveyed remote automotive attack surfaces including RKE for different car models. The authors of [12] conducted attacks against the RKE systems on six different car models. All these studies point to the need for enhanced security in RKE systems.

Attack models and defence techniques for autonomous vehicles are discussed in [13]. Similarly, the authors of [14] presented the current defence methods available in connected and autonomous vehicles. The authors of [15] proposed a taxonomy of attacks on autonomous vehicles and their defences. A radio frequency fingerprinting method to detect malicious requests is presented in [2]. An authentication protocol for RKE systems based on a symmetric encryption algorithm is presented in [16]. The Ultimate KeeLoq technology [6] uses a running timer to prevent replay attacks against RKE systems. An authentication method based on timestamps is proposed in [17] to prevent replay attacks against RKE systems. The authors of [18] proposed an authentication method using time stamping and XOR encoding. The authors of [19] presented an authentication scheme for RKE systems based on asymmetric cryptographic techniques.

However, most of the authentication mechanisms discussed above are computationally expensive. The authentication mechanism for RKE systems should be secure and lightweight. Hence, we propose an authentication mechanism based on PUFs for RKE systems.

B. Our Contributions

To protect RKE systems from replay and RollJam attacks, we propose a lightweight mutual authentication mechanism based on PUFs. Our contributions in this work can be summarised as follows:

1. Design of a novel mutual authentication mechanism for RKE systems: We propose a mutual authentication mechanism for RKE systems based on PUFs that provides all the required security features. The proposed authentication mechanism is computationally very efficient.

2. Protection from several attacks: The proposed mechanism protects RKE systems from replay, RollJam, and impersonation attacks.

3. Security proof: We provide formal security proof to show that the proposed mechanism is resilient against various attacks.

The rest of the paper is organized as follows. In Section II, we discuss the system model and the adversary model. In Section III, we present the proposed PUF-based authentication mechanism. Section IV presents the security analysis of the proposed mechanism with formal proof. Then, we present the performance analysis and provide a performance comparison with other existing RKE authentication mechanisms in Section V. Finally, conclusions are given in Section VI.

II. System Model and Adversary Model

A. System Model

The car is locked or unlocked using a key fob. The key fob has at least two buttons that activate the car's 'lock' and 'unlock' functions, respectively. A PUF is included with each key fob.

B. Adversary Model

We assume that the adversary has the capability to eavesdrop, capture, or jam the signals transmitted from the key fob to the car. Then, the adversary may replay the previously captured messages. If the adversary captures the signals exchanged in the RKE systems that employ static code, he/she can replay it later to unlock the car. The RKE systems based on rolling codes are protected from such replay attacks. However, the RollJam attack can be executed to compromise the RKE systems based on rolling codes.

In the RollJam attack, the adversary captures an unlock signal sent from the key fob to the car. At the same time, the adversary jams the same signal so that it will not reach the car. As the first attempt to unlock the car failed, the user presses the key fob button again. The adversary captures the second signal, jams it, and sends the first captured signal to the car. As the replayed signal sent from the adversary's device unlocks the car, the user does not notice this attack easily. However, the adversary has captured a valid signal (the second signal) that he/she can use later.

The attacks where an attacker A can eavesdrop, capture, jam, or replay the messages sent between the key fob and the car can be modelled using the following queries:

CaptureK(key fob, car, m) models the attacker's ability to eavesdrop and capture a message m sent from the key fob to the car.

CaptureC(car, key fob, m) models the attacker's ability to eavesdrop and capture a message m sent from the car to the key fob.

SendC(car, m) models the the attacker's ability to impersonate the key fob and sends a message m to the car.

SendK(key fob, m) models the attacker's ability to impersonate the car and sends a message m to the key fob.

Jam(key fob, car, m) models the attacker's ability to jam a message m between the key fob and the car so that the message does not reach the destination.

An attacker may call the queries *CaptureK*, *CaptureC*, *SendC*, *SendK*, and *Jam* a polynomial number of times.

III. Proposed Mechanism

In this section, we present the proposed lightweight, mutual authentication mechanism. The proposed mechanism consists of two phases: registration and authentication.

A. PUFs

Since the proposed authentication mechanism is based on PUFs, we start with a quick overview of PUFs. A PUF maps a challenge C_i to a response $R_i = PUF(C_i)$. The challenge-response pair is denoted as (C_i, R_i). An ideal PUF always outputs the same response R_i for a challenge C_i. If the challenges are different, the responses from the PUF also will be different. Two PUF devices produce different responses for the same challenge.

B. Assumptions

The assumptions made in this paper are given below:

• During the registration phase, the messages between the key fob and the car are transmitted through a secure channel.

• A PUF is included with each key fob. The PUF response for a given challenge cannot be predicted.

• Attempts to tamper with the PUF render it unusable [20].

C. Registration Phase

The steps in the registration phase are given below:

Step 1: The key fob composes a message M_{R1} with its ID ID_x and a registration request. Then, the key fob sends M_{R1} to the car.

Step 2: Upon receiving M_{R1} from the key fob, the car receiver generates a challenge C_i, composes a message M_{R2} with C_i, and sends it to the key fob.

Step 3: The key fob generates $R_i = PUF(C_i)$. Then, the key fob sends R_i to the car through a message M_{R3}.

Step 4: After receiving M_{R3} from the key fob, the car receiver stores the key fob's ID ID_x, and the challenge-response pair (C_i, R_i). Then, the receiver generates a key K_s and sends it to the key fob through a message M_{R4}.

Step 5: Upon receiving M_{R4} from the car receiver, the key fob stores the key K_s. This key will be used later during the authentication phase.

Before using the key fob for the first time, the registration phase must be completed. The registration phase can be enabled by adding one more button to the key fob or by pressing one of the existing buttons on the key fob a predetermined number of times. The registration phase is depicted in Fig. 1.

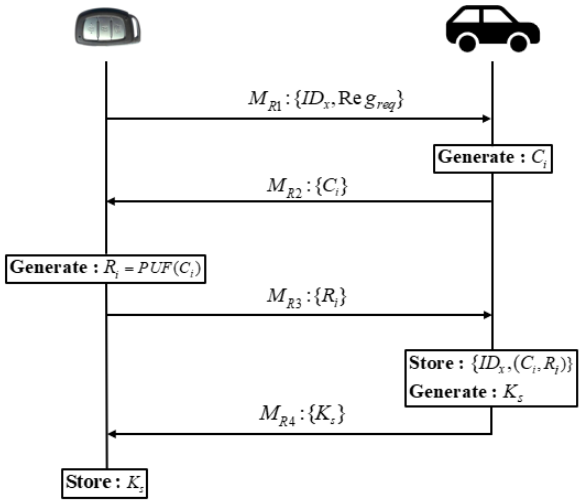

Fig. 1: Registration phase.

D. Authentication Phase

Step 1: When the user presses the key fob button, the key fob generates a nonce N_i. Then, the key fob computes $N_i^* = N_i \oplus K_s$. After that, the key fob composes a message $M_{A1} : \{ID_x, N_i^*\}$ and sends it to the car receiver.

Step 2: Upon receiving M_{A1}, the car receiver decodes $N_i = N_i^* \oplus K_s$. The receiver generates a nonce N_c and computes $N_c^* = N_c \oplus K_s$. Then, the receiver calculates an authentication parameter which is the secure hash of a concatenated message. To compute the authentication parameter, the receiver concatenates the parameters ID_x, N_i, K_s, and N_c and calculates the hash of the resulting concatenated message as $A_0 = H(ID_x \parallel N_i \parallel K_s \parallel N_c)$. A_0 serves as the authentication parameter from the car receiver. Then, the car receiver finds the challenge-response pair (C_i, R_i) stored together with ID_x. Then, it computes $C_i^* = C_i \oplus K_s$. After that, the receiver generates a new challenge C_i^{new} for the next round of authentication and computes $C_i^{*new} = C_i^{new} \oplus K_s$. Finally, the receiver composes $M_{A2} : \{A_0, N_c^*, C_i^*, C_i^{*new}\}$ and sends it to the key fob.

Step 3: The key fob receives M_{A2}. Then, it decodes $N_c = N_c^* \oplus K_s$ and verifies the authentication parameter A_0. If the authentication parameter is successfully verified, the key fob decodes $C_i = C_i^* \oplus K_s$ and $C_i^{new} = C_i^{*new} \oplus K_s$. Then, the key fob uses its PUF and generates the responses corresponding to the received challenges as $R_i = PUF(C_i)$ and $R_i^{new} = PUF(C_i^{new})$. After that, the key fob computes $R_i^* = R_i \oplus K_s$ and $R_i^{*new} = R_i^{new} \oplus K_s$. Then, the key fob concatenates the parameters ID_x, N_c, K_s, R_i, and R_i^{new}. Then, it calculates the hash of the resultant concatenated message as $A_1 = H(ID_x \parallel N_c \parallel K_s \parallel R_i \parallel R_i^{new})$. A_1 serves as the authentication parameter from the key fob. Finally, the key fob composes $M_{A3} : \{A_1, R_i^*, R_i^{*new}, cmd\}$ where cmd indicates the 'lock' or 'unlock' operation to be carried out. Then, the key fob sends M_{A3} to the car.

Step 4: Upon receiving M_{A3}, the receiver decodes $R_i = R_i^* \oplus K_s$ and $R_i^{new} = R_i^{*new} \oplus K_s$. Then, it verifies the response R_i and the authentication parameter A_1. If the verification is not successful, the authentication process is terminated. If the verification is successful, the 'lock' or 'unlock' operation is executed based on cmd. Then, the receiver updates the stored challenge and responses to C_i^{new} and R_i^{new}, respectively, to use in the next round of authentication. The details of the authentication phase are depicted in Fig. 2.

IV. SECURITY ANALYSIS

This section presents the formal security proof for the proposed authentication mechanism.

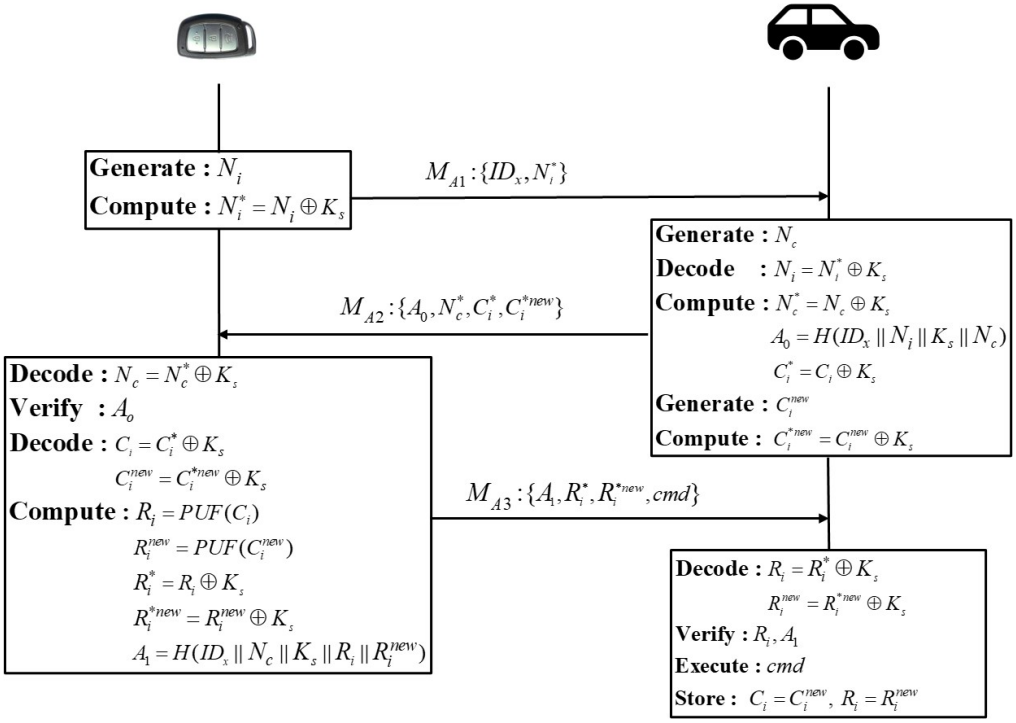

Fig. 2: Authentication phase.

Lemma 1. Proposed mechanism is resilient against cloning attacks.

Proof. PUFs cannot be cloned [21]. Hence, an adversary cannot clone them to generate the responses required during authentication.

Lemma 2. The probability of correctly predicting a PUF's response is negligible.

Proof. If a PUF returns a response R_i of length l_2 for a challenge C_i of length l_1, the PUF's challenge-response pair (C_i, R_i) can be represented as $(\{0,1\}^{l_1}, \{0,1\}^{l_2})$. A challenge-response game can be used to define the security of the PUF. If an adversary A can guess the PUF response correctly for a challenge C_i, A wins the game.

1. A challenger C sends a challenge C_i to A. The PUF response corresponding to C_i is R_i.

2. A guesses the PUF output as R_i^* and wins the game if the PUF response is predicted correctly, i.e., if $R_i = R_i^*$.

A's advantage in this security game is the probability of predicting R_i correctly. It can be modelled as $\alpha_{PUF} = Pr[R_i = R_i^*]$. From Lemma 1, a PUF cannot be cloned to generate responses. Hence, the only option for A to predict R_i is to make a random guess. As a result, $\alpha_{PUF} = Pr[R_i = R_i^*] = \frac{1}{2^{l_2}}$.

Theorem 1. The proposed protocol is resilient against replay attacks.

Proof. The game where an adversary A tries to execute a replay attack is given below:

1. A eavesdrops and captures a message M_{A1} by running the *CaptureK(key fob, car, M_{A1})* query.

2. A replays the message M_{A1} using the *SendC(car, M_{A1})* query at a later point in time to get authenticated as the key fob.

3. If one round of the authentication process is completed successfully, A wins the game.

When A replays the message M_{A1}, the receiver sends the corresponding message M_{A2} to A as mentioned in the authentication protocol. A should compose M_{A3} to continue the authentication process. A should know the correct PUF response to compose M_{A3}. From Lemmas 1 and 2, the only possibility for A to know the PUF response is to guess it. If R_i has l_2 bits, the probability of A correctly guessing R_i is $Pr[R_i = R_i^*] = \frac{1}{2^{l_2}}$ which is negligible. As a result, A cannot continue the authentication process. A's advantage in this game is $\alpha_{replay} = Pr[R_i = R_i^*] = \frac{1}{2^{l_2}}$ which is negligible. Thus, A cannot execute the replay attack to get authenticated. Thus, the proposed protocol prevents replay attacks.

Theorem 2. An adversary A cannot execute the RollJam attack on the proposed mechanism.

Proof. The RollJam attack on the key fob can be modelled using a security game between *A* and the car as:

1. During the key fob's i^{th} authentication round, A eavesdrops the message M_{A1}^i and captures it by running the *CaptureK(key fob, car, M_{A1}^i)* query.

2. A jams the message M_{A1}^i by running the *Jam(key fob, car, M_{A1}^i)* query.

3. During the key fob's next authentication attempt, A eavesdrops the message M_{A1}^{i+1} and captures it by running

the *CaptureK(key fob, car,* M_{A1}^{i+1}*)* query.

4. A jams the message M_{A1}^{i+1} by running the *Jam(key fob, car,* M_{A1}^{i+1}*)* query. A replays the message M_{A1}^{i} using the *SendC(car,* M_{A1}^{i}*)* query.

5. A replays the message M_{A1}^{i+1} using the *SendC(car,* M_{A1}^{i+1}*)* query at a later point in time to get authenticated as the key fob.

6. If one round of the authentication process is completed successfully, A wins the game.

When A replays the message M_{A1}^{i+1}, the receiver sends the corresponding message M_{A2}^{i+1} to A as mentioned in the authentication protocol. A should compose M_{A3}^{i+1} to continue the authentication process. A should know the correct PUF response to compose M_{A3}^{i+1}. From Lemmas 1 and 2, the only possibility for A to know the PUF response is to guess it. If R_i has l_2 bits, the probability of A correctly guessing R_i is $Pr[R_i = R_i^*] = \frac{1}{2^{l_2}}$ which is negligible. As a result, A cannot continue the authentication process. A's advantage in this game is $\alpha_{RollJam} = Pr[R_i = R_i^*] = \frac{1}{2^{l_2}}$ which is negligible. Thus, A cannot execute the RollJam attack to get authenticated.

Theorem 3. The proposed protocol provides mutual authentication.

Proof. The game where an attacker A tries to get authenticated as a legitimate key fob is given below:

1. A initiates the authentication process with the car.

2. A calls *SendC(car, m)* query as a legitimate key fob.

3. If one round of the authentication process is completed successfully, A wins the game.

A needs to send valid messages M_{A1} and M_{A3} to the car receiver for successful authentication as a key fob. A has two options to send the correct messages: capture valid messages and replay it later or generate valid messages as a legitimate key fob. From Theorem 1, A cannot capture valid messages and replay it later to get authenticated. From Theorem 2, A cannot execute the RollJam attack as well, i.e., α_{replay} and $\alpha_{RollJam}$ are negligible. Hence, the only option for A to get authenticated is to generate valid messages as a legitimate key fob. To compose M_{A3}, A needs to know the correct PUF response R_i. From Lemmas 1 and 2, the only possibility for A to know R_i is to guess it. If R_i has l_2 bits, the probability of A correctly guessing R_i is $Pr[R_i = R_i^*] = \frac{1}{2^{l_2}}$. Hence, the adversary's advantage in successfully authenticating as a valid key fob α_{Auth} is negligible. Similarly, the probability of generating valid messages as a car receiver is negligible.

Hence, if one round of the authentication process is completed successfully, the messages received at the car receiver must have originated from a valid key fob. Similarly, we can show that if one round of the authentication process is completed successfully, the messages received at the key fob must have originated from a valid car receiver. Thus, the proposed authentication mechanism provides mutual authentication.

V. Performance Analysis and Comparison

In this section, we evaluate and compare the performance of the proposed mechanism with other similar schemes.

A. Security Properties

For the performance comparison, we consider security properties such as mutual authentication and protection against various attacks. The comparison of the security features is summarised in Table I. Though [16], [17], and [18] cover some of the key security features, mutual authentication is an additional property provided by the proposed protocol. Clock synchronization between the key fob and the receiver is required for the protocols mentioned in [17] and [18]. On the contrary, the proposed protocol does not require clock synchronization.

B. Computation Cost

Next, we consider the computation cost. The registration phase occurs only once. Hence, we evaluate the computation cost during the authentication phase.

The key fob requires 6 XOR, 2 PUF, 4 concatenation, and 1 hash operations. The car receiver requires 6 XOR, 3 concatenation, and 1 hash operations for one iteration of the authentication process. We use a Raspberry Pi 3B to simulate the RKE system and to run operations such as hash (SHA-1), XOR, and concatenation. The operations have been simulated in the Python programming language. Let T_H, T_{XOR}, and T_{co} denote the time taken by hash, XOR, and concatenation operations, respectively. From the simulations, T_{XOR} is 10.9 µs, T_H is 16.9 µs and T_{co} is 4.9 µs. We consider a PUF proposed in a recent paper [22] to be deployed in the key fob for the proposed mechanism. The PUF generates a response of 320 bits with an operation time of 0.4 µs. The number of operations performed by the key fob and the receiver and the computation time during the authentication phase are shown in Table II.

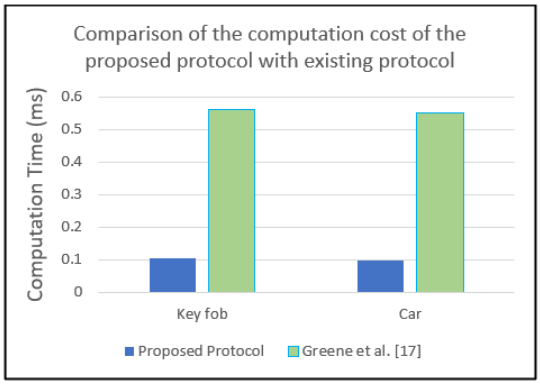

Fig. 3: Comparison of computation cost.

In contrast to the PUF-based protocol proposed in this paper, the protocol in [17] uses AES encryption and decryption. The proposed protocol does not use AES encryption

TABLE I: Comparison of the proposed mechanism with existing authentications mechanisms: security features

Features	Glocker et al. [16]	Greene et al. [17]	Greene et al. [18]	Proposed Mechanism
Key fob Authentication	Yes	Yes	Yes	Yes
Mutual Authentication	No	No	No	Yes
Resilience Against Replay Attack	Yes	Yes	Yes	Yes
Resilience Against RollJam Attack	No	Yes	Yes	Yes
Requirement for Clock Synchronization	No	Yes	Yes	No
Security Proof	No	No	No	Yes

TABLE II: Number of operations and computation time

Operation	**Key fob**	**Receiver**
XOR	6	6
PUF	2	0
$Concatenation$	4	3
$Hash$	1	1
Computation Time (ms)	0.103	0.097

and decryption. Let N_{ENC} denote the number of AES encryption/decryption operations. From the simulations, T_{ENC} is 0.55 ms. The comparison of the computational cost of the proposed mechanism and the protocol in [17] is plotted in Fig. 3. Hence, we can conclude that the proposed mechanism provides more security features than other existing schemes while having a lower computation cost.

VI. Conclusion

In this paper, we explored the attacks on RKE systems. Then, we proposed a mutual authentication mechanism based on PUFs for RKE systems. We also showed that the proposed mechanism is lightweight while still providing all the necessary security features. The security and performance analysis of the proposed mechanism shows that PUFs can be used to achieve an efficient authentication mechanism for RKE systems.

VII. Acknowledgement

This work was supported in part by grants R-263-000-E78-114 and R-263-001-E78-114 funded by the Ministry of Education, Singapore.

References

[1] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors*, vol. 20, no. 8, p. 2364, 2020.

[2] K. Joo, W. Choi, and D. H. Lee, "Hold the door! fingerprinting your car key to prevent keyless entry car theft," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.

[3] A. I. Alrabady and S. M. Mahmud, "Analysis of attacks against the security of keyless-entry systems for vehicles and suggestions for improved designs," *IEEE transactions on vehicular technology*, vol. 54, no. 1, pp. 41–50, 2005.

[4] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlides, "Lock it and still lose it - on the (in) security of automotive remote keyless entry systems," in *Proceedings of the 25th USENIX Security Symposium*, 2016, pp. 929 – 944.

[5] R. Benadjila, M. Renard, J. Lopes-Esteves, and C. Kasmi, "One car, two frames: attacks on hitag-2 remote keyless entry systems revisited," in *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

[6] Ultimate keeloq technology. [Online]. Available: https://www.microchip.com/en-us/solutions/wireless-connectivity/rf-remotes/ultimate-keeloq-technology

[7] Hitag 2. [Online]. Available: https://www.nxp.com/products/rfid-nfc/hitag-lf/hitag-2-transponder-ic:HT2X

[8] Defcon 23 rolljam attack. [Online]. Available: https://samy.pl/defcon2015/

[9] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *2007 44th ACM/IEEE Design Automation Conference*, 2007, pp. 9–14.

[10] A. Chattopadhyay, K.-Y. Lam, and Y. Tavva, "Autonomous vehicle: Security by design," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[11] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *black hat USA*, vol. 2014, p. 94, 2014.

[12] O. A. Ibrahim, A. M. Hussain, G. Oligeri, and R. Di Pietro, "Key is in the air: Hacking remote keyless entry systems," in *Security and Safety Interplay of Intelligent Software Systems*. Springer, 2018, pp. 125–132.

[13] M. Pham and K. Xiong, "A survey on security attacks and defense techniques for connected and autonomous vehicles," *Computers & Security*, p. 102269, 2021.

[14] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (cavs)," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[15] V. L. Thing and J. Wu, "Autonomous vehicle security: A taxonomy of attacks and defences," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2016, pp. 164–170.

[16] T. Glocker, T. Mantere, and M. Elmusrati, "A protocol for a secure remote keyless entry system applicable in vehicles using symmetric-key cryptography," in *2017 8th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2017, pp. 310–315.

[17] K. Greene, D. Rodgers, H. Dykhuizen, K. McNeil, Q. Niyaz, and K. A. Shamaileh, "Timestamp-based defense mechanism against replay attack in remote keyless entry systems," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1–4.

[18] K. Greene, D. Rodgers, H. Dykhuizen, Q. Niyaz, K. Al Shamaileh, and V. Devabhaktuni, "A defense mechanism against replay attack in remote keyless entry systems using timestamping and xor logic," *IEEE Consumer Electronics Magazine*, vol. 10, no. 1, pp. 101–108, 2020.

[19] R. P. Parameswarath and B. Sikdar, "An authentication mechanism for remote keyless entry systems in cars to prevent replay and rolljam attacks," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1725–1730.

[20] M. S. Kirkpatrick, S. Kerr, and E. Bertino, "System on chip and method for cryptography using a physically unclonable function," U.S. Patent 8,750,502, Jun 10, 2014.

[21] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[22] G. Bansal and B. Sikdar, "S-maps: Scalable mutual authentication protocol for dynamic uav swarms," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 088–12 100, 2021.