Secure Wireless Smart Car Door Unlocking System

*Navneet Vinod Melarkode[1], Varun Niraj Agarwal[2], Avaneesh Kanshi[3], Dr. Anisha M. Lal[4]
[1 2 3 4] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu – 632014
[1] navneetmelarkode@gmail.com
[2] varunswaika@gmail.com
[3] avaneeshkanshi5@gmail.com
[4] anishamlal@vit.ac.in

* Corresponding Author

Abstract. Security is the most essential factor in software as well as internet environment. In the current era, software and hardware security is the most important thing to secure our digital and physical assets. In many cases, tangible entities like smart homes, smart car etc. having some IoT base support enhance the supervision. This paper proposes a secure alternative for the car unlocking system by dynamically generating disposable encryption keys that cater to the future demands of smart cars powered by Internet of Things while employing ultra-modern communications techniques like MAC-then-encrypt. Encryption keys are used to facilitate locking and unlocking functionality of car via the internet to ensure security. The paper compares the proposed methodology with existing algorithms like Digital Signature Authentication to verify its effectiveness in real world scenarios.

Keywords: Connected Cars, Internet of Vehicles, Pseudorandom Number, Hashing, Rolling Code.

1 INTRODUCTION

The recent growth of smart cars in Internet of Vehicles (IoV) is expected to increase from 40% in 2020 to 70% in 2025. Rapid development of infrastructure, systems, artificial intelligence learning models would cause a major shift of the consumers from the conventional type of vehicles to a more modern type. This is may also bring increased security issues, one of the major concerning issues being car door unlocking system. IoV vehicles are equipped with certain secret key that is transmitted over the internet to the owner's car that is also connected to the internet. There are two types of security aspects when it comes to a car key, i.e., software and hardware security. Software security is the area which deals with the securing the data that is being transmitted over the IoT network. Hardware security deals with the inbuilt security system of hardware to prevent any intruder from entering and making the hardware vulnerable. The paper focuses on the software security of the car unlocking system. When the secret key is matched with the one stored inside the car, the car is unlocked or locked as per the needs. This method may seem to be very time inexpensive and easy for the manufacturers to implement into the system. But it has a drawback, if an attacker is able to get inside the network, they can sense any piece of information that is being sent over the internet. If their intentions are for committing a car theft, they can easily read the key and replicate it later on for unlocking the car. The current methodology of car door unlocking system compromises its security aspect. Several approaches like symmetric and asymmetric encryption and decryption have been implemented but later on found out that it can easily be cracked with hardware-induced attacks. This paper proposes a novel approach to avoid the aforementioned problem by implementing an end-to-end system that generates disposable keys to verify the authenticity of the client. To send the keys over the internet via an unsecure public channel compromises the security. The proposed model ensures that any message sent through unverified channels are encrypted and hashed to avoid breach of trust and privacy.

1.1 Organization of the Paper

- Section 2 explains about the existing methodologies that have been proposed by other authors.
- Section 3 explains about the proposed methodology and how a unique framework was created by implementing ideas from existing security algorithms.
- Section 4 explains about the metrics that are considered while comparing the proposed model with the industry standards. It also talks about the comparison of different sub processes that are used in the algorithm.

2 RELATED WORK

Pranab Ray et al. [1] incorporate the rolling key algorithm to overcome the flaws in its predecessor methodologies. Previous studies employed RF security which would have to pass through unsecure channels. The signals that pass through these channels are essential for locking and unlocking the system which brings about a blaring disadvantage in the security standpoint. To prevent and overcome the relay threats caused by unsecure channels, the rolling key algorithm is implemented which enables a 2-way handshake. The rolling key algorithm provides an unused key at every instance of data transfer between the sender and receiver.

R. Valanarasu [2] present a secure architecture for hospital environments with the help of an Internet of Things backend. They recognized limitations in the current framework such as inflexible modes of networking, data security etc. and strive to alleviate these threats with the help of the proposed method. The method presents a major upgrade from the existing methods by using a regulation and policy layer to overlook all the trust components such as safety, privacy and dependability.

Colin Urquhart et al. [3] analyze the underlying cyber security of a renowned car brand. It has been noted that as the technology of automobiles advance, many instances of cars being connected via 3G and 4G mobile networks are reported. While these services increase the ease of use and aid the consumer, they come with vulnerable system defects that lead to increase in the attacking area of the vehicle. The paper also carries out experiments to assist the study in the field of in-car attacks as well as standard benchmark attacks such as key fob rolling code and various other privacy attacks.

L. Jamjoom et al. [4] focus on developing a wireless car lock controller built on a mobile device. The research implements the said system incorporating Internet of Things concepts. The proposed methodology revolves around granting requests through a server-based utility over the internet. To accommodate the authorization, a code is sent to the mobile device. The mobile device must have the companion application pre-installed to complete the communication of the code. The code is transmitted via Bluetooth. It is then sent to a front-end controller which employs recognition techniques and relays a flag back to the mobile device. Future research in this field can aim to improve the monitoring of other parameters that include but are not limited to fuel consumption, Carbon Dioxide emission rate etc.

Sophia Auer et al. [5] use the growing popularity of in-car sharing and the rising number of applications of Blockchain technology as motivation for devising a new methodology for shared mobility that involves key

aspects from both its constituents, Blockchain and Internet of Things technologies. The paper presents an architecture for encapsulating these technologies to assist car-leasing and car-sharing. The proposed method goes a step ahead by eliminating the requirement for keys to gain vehicle access. The authors identify that while sustainable, blockchain technology alone cannot expand this field in the future. Future researches can aim to evaluate the authenticity of the Internet of Things devices and the scalability of the proposed model. At present, the model fares well for simulated data.

Jeffrey Cashion et al. [6] presents a unique way of authentication of the client to the server using rolling code technology. It prevents the intruders from making copies of the code and using it to perform sidejacking. With the help of this technique, the client easily proves the server of its legitimacy. This in return prevents hijacking of the system to its best. The efficiency of the proposed code puts out optional payload integrity and confidentiality via a multi-level security model.

Kyle Greene et al. [7] performs set of infiltration experiments which revealed the vulnerabilities in the radio frequency communication of cars and garages in remote keyless systems. A timestamp-based solution was presented to enhance existing rolling code algorithm. The output of the rolling code algorithm is appended with certain command and the timestamp which is further encrypted using AES algorithm. The algorithm proved to be highly secure with low complexity and able to be power efficient.

L. Jamjoom et al. [8] develops a smartphone based wireless controlled car lock. The idea is to allow a group of authorised people to share a lot of cars. Whenever an authorised person requires a car, a request is firstly submitted via internet to a server-based utility. On the availability of a car the request is granted and vice versa.

Sophia Auer [9] presents a high-level architecture for a blockchain-IoT-based platform for promoting shared mobility combining car-sharing and car-leasing. The proposed platform requires secure information sharing among multiple stakeholders (such as user, lessee, and service provider), leading to the decision to choose blockchain for its facilitation. IoT data generated by vehicles is of significant relevance to all involved stakeholders helping to streamline processes and features. This proved to provide security, privacy, authenticity, reliability and scalability without making the system more complex.

Hamdan Y. B. and Sathesh [17] highlight about 2 kinds of threats, namely information privacy attack and context aware privacy attack. Privacy is a major concern while communicating over the internet and hence needs proper security measures to be taken to ensure privacy of client-server channel.

It can be observed that the recurring pitfalls in previous keyless entry methods are data leakage, replay attacks and session hijacking. Data leakage occurs due to plain text being sent over unsecure communication channels which can be easily intercepted. Replay attacks are extremely common in keyless entry systems. They work on the assumption that the same signal can be used repeatedly to achieve the same result. Session hijacking is similar to a man-in-the-middle attack which interrupts the communication channel and tries to exploit the vulnerabilities of the system. Previous methods highlight the need for scalable systems as the Industry of Vehicles continue to grow at an exponential rate.

2.1 Pseudorandom Numbers

[10] A pseudorandom number sequence is a number sequence which appears to be statistically random, but are completely generated with the help of a deterministic function. These sequences are not truly random as they are heavily determined by the initial value set by the user. This initial value is also called as the seed. These generators are key in practice owing to their speed in generating these sequences.[11] While pseudorandom number generators are calculated using a deterministic function, it is necessary for the sequence to show approximate characteristics of a true random distribution. To do so, Python's NumPy library has been employed. By pre-generating the seed, the pseudorandom sequence can be generated in the desired way.

2.2 Rolling Code

[12] Rolling codes essentially transmit a counter that is incremented by each button press in a cryptographically authenticated way. [13] Rolling codes have been incorporated into many keyless entry doors unlocking systems due to its extreme versatility and lightweight nature. From [14], it can be seen that introducing a pseudorandom number generator into the rolling code can help extend the protection to prevent template attacks as well as alleviate the risks posed by brute force attacks. This is possible because the two random number generators on either side of the pipeline generate the same sequence of numbers which otherwise seem random for any intruder trying to intercept this stream.

2.3 Hashing

Hashing functions were proposed as an alternative to fulfil aspects of information security due to various reasons. [15] Firstly, it is computationally easy to calculate the hash of any given message when compared to asymmetric and symmetric key algorithms. Furthermore, two different messages are highly unlikely to be associated with the same hash because of the sheer number of possible hash values. Hash functions being employed in the proposed model are collision resistant to a high degree. Subsequently, messages cannot be altered with without changing the hash. Ultimately, generating a message from a given hash is very unfeasible thus nearly eliminating brute force attacks [16].

The proposed method employs SHA hashing function over the MD5 hashing function due to very less and infrequent potential collisions. MD5 generates a 128-bit output while SHA256 generates a 256-bit output. While the SHA256 hashing may be slower than MD5, the speed is not slow enough for us to overlook the security advantages it provides.

3 METHODOLOGY

3.1 Objective

The paper presents a novel approach by combining the power of the tried and tested state-of-the-art algorithms which are designed to work in a simplex communication environment and ultra-modern hashing technologies.

This enables us to encrypt any data in an irreversible manner, making the communication over the internet extremely secure. Our proposed method employs a rolling filter as a key generation method for a modified MAC-then-encrypt authenticated encryption algorithm. Instead of using a bi-directional encryption algorithm in the final stage of MAC-then-encrypt, the proposed model uses another (different from key hash) hashing algorithm which

makes the encrypted message doubly hashed and impossible to crack. Double hashing and a unidirectional approach can be integrated due to the fact that the message is predefined in both of the systems and only the source of the messages needs to be authorised. A unidirectional double hashing methodology can be implemented by chaining the outputs of subprocesses to form the resultant message string.

The proposed methodology brings elements of security and robustness together. By sending uncrackable messages over the internet, the proposed system ensures authentication in both the parties. This has not been implemented in predecessor systems which adds to the novelty of the entire concept.

3.2 Modules

The proposed method can be divided into 2 modules. The first one being client/key side and the second one, the car side. The client will be interchangeably used as key since the client is the key in the proposed framework. The client will try to lock or unlock the car from their remote device connected through the internet. Connected cars are always connected to the internet. The car will be listening for a hash digest over the internet. As soon as it receives the digest, the car will then validate and verify the authenticity to proceed to take a decision on the car controls.

Certain assumptions made in the following 2-way handshake system include the initialization of a seed before the client and car are actually made public. This seed will be completely random, and it is necessary for the same seed to be set in both the car as well as the client. This is essential because the entire architecture of Pseudorandom number generated rolling codes is dependent on this. Furthermore, in case the client side rolling code queue does go out of sync with the car rolling code queue, a manual reset must be performed with the initialization of a new random seed.

3.2.1 Key Side

The key will generate a rolling code queue with the seed that is initialized with. The rolling code queue uses the pseudorandom number generator with the deterministic function to generate a stream of numbers. These numbers are then enqueued sequentially. Upon pressing of the car control button such as lock or unlock; the client then polls the queue and hashes the polled pseudorandom number. Depending on the car control button pressed, the corresponding code will then be appended to the newly generated hash digest. After appending the code, the entire message is now hashed again in order to protect the status of the car from being compromised in the case of a man in the middle attack. The architecture followed here is an indirect implementation of the MAC-then-Encrypt scheme where a MAC is produced based on the plaintext, and the plaintext and MAC are encrypted again to produce a ciphertext based on both. While the ciphertext is sent, the result hash digest is sent to the car in the proposed system.

The key side has to perform hashing twice for every button click, and given the computational prowess of modern-day machines, the given model feels viable. In the off chance where the key isn't connected to the car side, the counter of the rolling code will keep getting incremented until a point where the car queue won't be able to look ahead and get back in sync. In the case where a hacker impersonates a client and sends repeated signals to the car, a Denial-of-Service prevention mechanism is in place depending on the number of false signals being sent in a given amount of time.

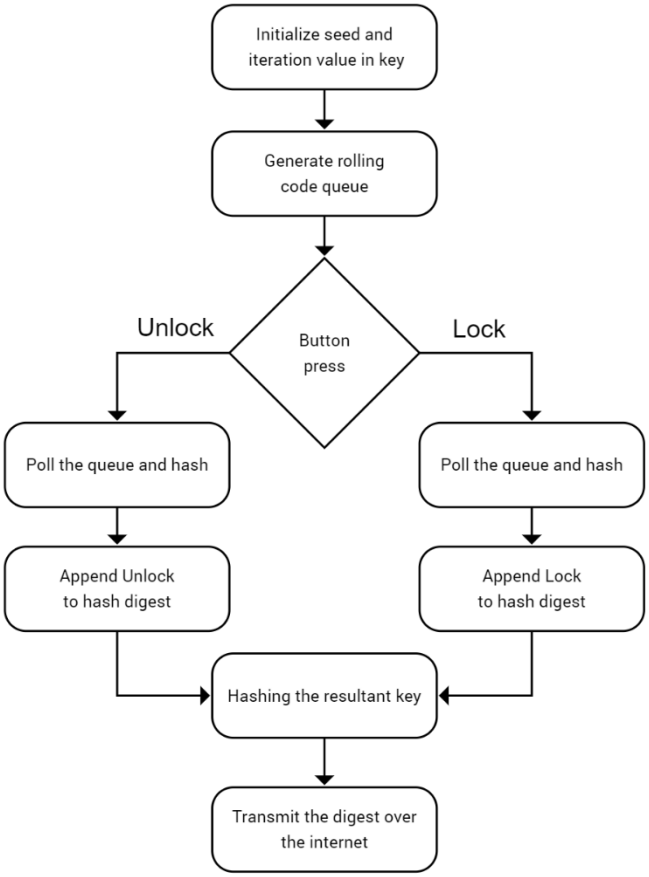

Figure 1. Flowchart for client side

From Figure 1, It can be seen that the doubly hashed message sent over the internet is secure. It is to be noted that the client side does not perform any validation or verification of the person pressing the button. This is analogous to a traditional key, where nobody can actually verify if it is actually the key owner opening the lock. Apart from the 2 hashes and the appending of car control signal, the client side does not perform any other computations. The same flowchart is given in the form of words in Algorithm 1.

1. Initialize queue rq
2. Populate rq
3. if button_press is true:
    a) $rq.top \rightarrow temp$
    b) $rq.pop$
    c) hash($temp$) $\rightarrow temp$
    d) $temp$ + “LOCK”/ “UNLOCK” $\rightarrow temp$
    e) hash($temp$) $\rightarrow temp$
    f) Transmit

Algorithm 1. Client-side algorithm

3.2.2 Car Side

The car side algorithm is fairly trivial as compared to the key side. Just like the key, the car too will be fitted with the same randomly generated seed. The car generates the same set of pseudorandom numbers and will listen for any message that is sent over the internet.

Upon receiving the message, the car checks whether the queue is empty. Queue being empty is an edge case for when the car and key go out of sync. If the queue is empty, the car and key would then have to be manually reset with a brand-new seed. Until then, the internet functionalities would be restricted in the car to prevent future attacks as the car is now vulnerable. If the queue is not empty, the car then polls the queue and subjects it to the exact same hashing system to generate a hash digest.

As mentioned earlier, hashes cannot be decrypted, but it can only be compared with other hash digests to check their validity. Consequently, each number from the queue would have to be hashed and then appended with each control signal/code of the car (Lock or unlock) and then hashed again.

The resulting hash digest will be compared with the message sent over the internet. If they are a match, the car will execute the corresponding control, and if they are not, the car moves ahead until the entire queue is tested. The ideal rolling code queue size is 256, which would mean it would take exactly 256 out of sync clicks from the key side to restrict the car's internet activity and make it available only for physical locking/unlocking.

From Figure 2 below, it can be seen that the car actively listens for a message and upon receiving of the message, it validates the hash digest by subjecting the rolling code queue to the same hash functions in the same order. For each number, the car will have to perform 1 common hash with the addition of 2 hashes after appending the respective car control signal (Lock or unlock) to the intermediate hash digest. The same can be seen with the following algorithm.

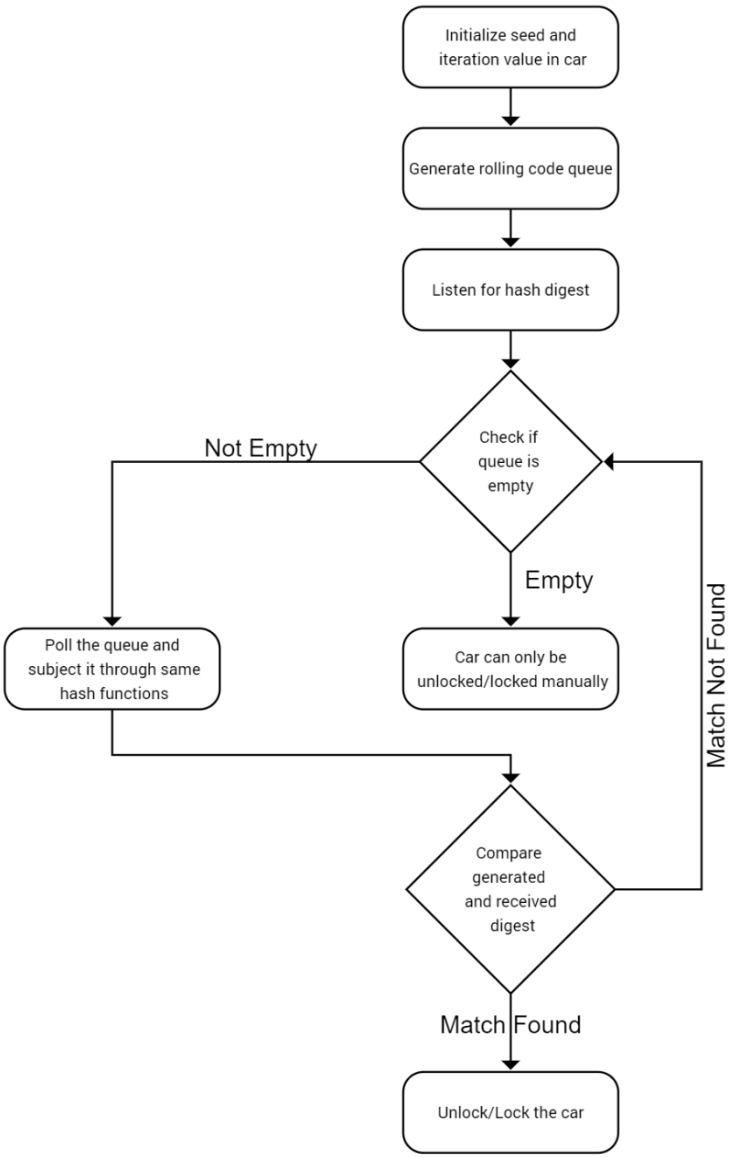

Figure 2. Flow chart for Car side

1. Initialize queue rq
2. Populate rq
3. Initialize TCP socket
4. Receive 256 bits of data → d
5. for each num in rq :
    a) hash(num) → $temp$
    b) if $temp$ == d:
        1. Lock or Unlock the car
6. if no match found:
    a) Restrict car's activities over internet

Algorithm 2: Car side algorithm

4 PERFORMANCE EVALUATION

4.1 Type

In the proposed model, the encryption function is a hash digest and the decryption function is a hash comparison. Hence making the program extremely secure towards brute force attacks.

4.2 Function Analysis

SHA256 is used for hashing, rolling code for disposable key generation and finally a MAC-then-encrypt architecture for the network authentication.

This sequential model involves using multiple tried and tested models which theoretically yields a non-brute forceable encryption standard at the cost of computational power while maintaining the integrity of the data.

4.3 Key Size

The proposed model uses a random number generator which yields a key of size 10^8. This can be increased depending on the computational power available. However, since the final output is hashed, the output is always a fixed size of 256 bits.

4.4 Rounds

The proposed model hashes in 2 rounds and ensures maximum encryption.

4.5 Time complexity

The time complexity for each encryption / digest is,

$$\sum = T(n) = 2*({}^{29}C_1 + {}^{4}C_2)\, N_0 + 2*({}^{10140}C_1 + {}^{2}C_3 + {}^{64}C_4 + {}^{2}C_5)\, N_1 = \Theta(N) \quad (1)$$

From (1), it can be clearly observed that the encryption function is linear in time and hence can provide great benefits while not compromising on quality of the hash digest. This encryption function is running a total of k times. Hence, the total time complexity for the embedded system in the car is Θ (l*N). However, once the queue is generated, the time complexity reduces to Θ (N) and the time complexity to check becomes Θ (l*k*256) = 0(l*k). Where k represents the queue length and l represents the number of functions. The complexity of the sender-side and client-side encryption is Θ (n).

4.6 Space Complexity

For the car side,

$$\sum = S(n) = l*k*(256) = l*k \quad (2)$$

In (2), k are the cycles and l are the number of functions. Increasing the number of cycles will increase the space complexity, however, the space complexity is linear. This enhances the scalability of the proposed method. On the car side the space complexity is mere Θ (1).

4.7 Common attacks

1) Brute force - Brute force attacks are practically impossible since the hash digest is hashed again with a disposable key.

2) Replay attacks - Replay attacks are not possible because the rolling code produces a new key which cannot be reused every cycle.

3) Rolling code overflow - Rolling code overflow is a common issue in systems employing rolling code. The proposed model aims to minimize it with the help of a failsafe which blocks failed IP addresses however this is not a full proof method to solve this issue.

4) DoS attacks - DoS attacks can easily be addressed by employing a simple software firewall / IPS but it can lead to not being able to open the door over the internet. Firewalls provide a filter-based as well as a rule-based packet forwarding mechanism which blocks anomalous requests upon multiple failed attempts.

4.8 Comparison with DSA algorithm standard.

Digital Signature Algorithm (DSA) serves as the benchmark for the proposed method as it is the industry standard for authentication systems. DSA uses a private key to sign messages which can be verified in the client side using a public key. DSA provides users with integrity – clients can verify the contents of the message and non-repudiation – the server cannot claim that they have not signed the message. This is the reason DSA has been chosen as the evaluation algorithm.

1) Architecture – Both DSA and the proposed architecture employs a MAC-then-encrypt architecture for its foundation. However, the message is visible in digital signature algorithm and it is extremely prone to replay attacks.

2) Key – The key size in rolling code is variable and the keys are disposable and changes constantly.

3) Rounds – Both DSA and the proposed algorithm employs the same number of rounds.

4) Time complexity – DSA outperforms the proposed methodology in this comparison metric. DSA is Θ (k*l) times faster however since the values of k and l are relatively small, it can be considered that DSA is constant time faster than our proposed work.

5) Space complexity – Both the algorithms are similar in terms of space complexity.

4.9 Usage of SHA vs MD5 in our proposed method

SHA	MD5
Highly Secure as the final output is 256 / 512 bits	Exponentially less secure as the final output is only 128 bits
Half as fast as MD5	Double the speed of SHA
No known attacks	Many reported attacks known
Fixed input size	Any input size works

Table 1. Comparison of SHA and MD5

From Table 1, it is evident that SHA outperforms the MD5 hashing technique in the department of security. While SHA is more time consuming, it is not slow enough to overlook the security benefits it provides.

4.10 DSA vs proposed method

Proposed method	DSA
Based on MAC-then-encrypt architecture	Based on MAC-then-encrypt architecture
Slower compared to DSA	Faster than proposed method
Highly secure	Less secure
Replay attacks do not work	Susceptible to common replay attacks
New disposable key every cycle	Fixed key

Table 2. Comparison of DSA and the proposed method

From Table 2, it can be observed that the proposed method performs better than DSA in the security aspect. On the contrary, it lacks speed in comparison to DSA.

5 CONCLUSION

The proposed algorithm which aims to secure the IoT based car lock system incorporates the basic notion of MAC-then-encrypt architecture layered with cryptographic mathematical function and pseudorandom number generator function. This results in increasing the randomness of the serial keys generated while double securing the signal being sent over the internet. This provides the smart car with improved layer of security for locking mechanism and a secure solution for the booming industry of IoVs. Even though the algorithm requires multiple computations to be performed, it doesn't compromise on its efficiency, making the proposed methodology highly efficient. The algorithm succeeds in strongly preventing any intruder launching a replay or session hijacking attack due to the encryption key's dynamic nature. The integrity was tested against several attacks and proved to remain uncrackable. Despite providing good security, the time and space complexity can be further optimized to elevate the performance. On a large scale, it may prove to be fatal due to large amount of storage space being required to store the dynamic changing keys hence further research is required for management of encryption keys.

6 REFERENCES

1. Ray, P., Sultana, H. P., & Ghosh, S. (2019). REMOVING RF VULNERABILITIES FROM IOT DEVICES. *Procedia Computer Science*, *165*, 421-427.

2. Valanarasu, M. R. (2019). Smart and secure IoT and AI integration framework for hospital environment. *Journal of ISMAC*, *1*(03), 172-179.

3. Urquhart, C., Bellekens, X., Tachtatzis, C., Atkinson, R., Hindy, H., & Seeam, A. (2019). Cyber-security internals of a skoda octavia vRS: A hands on approach. *IEEE Access*, *7*, 146057-146069.

4. Jamjoom, L., Alshmarani, A., Qaisar, S. M., & Akbar, M. (2018, February). A wireless controlled digital car lock for smart transportation. In *2018 15th Learning and Technology Conference (L&T)* (pp. 46-51). IEEE.

5. Auer, S., Nagler, S., Mazumdar, S., & Mukkamala, R. R. (2022). Towards blockchain-IoT based shared mobility: Car-sharing and leasing as a case study. *Journal of Network and Computer Applications*, 103316.

6. Cashion, J., & Bassiouni, M. (2011, October). Robust and low-cost solution for preventing sidejacking attacks in wireless networks using a rolling code. In *Proceedings of the 7th ACM symposium on QoS and security for wireless and mobile networks* (pp. 21-26).

7. Greene, K., Rodgers, D., Dykhuizen, H., McNeil, K., Niyaz, Q., & Al Shamaileh, K. (2020, January). Timestamp-based defense mechanism against replay attack in remote keyless entry systems. In *2020 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-4). IEEE.

8. Jamjoom, L., Alshmarani, A., Qaisar, S. M., & Akbar, M. (2018, February). A wireless controlled digital car lock for smart trans portation. In *2018 15th Learning and Technology Conference (L&T)* (pp. 46-51). IEEE.

9. Auer, S., Nagler, S., Mazumdar, S., & Mukkamala, R. R. (2022). Towards blockchain-IoT based shared mobility: Car-sharing and leasing as a case study. *Journal of Network and Computer Applications*, 103316.

10. Lagarias, J. C. (1993). Pseudorandom numbers. *Statistical Science*, *8*(1), 31-39.

11. James, F. (1990). A review of pseudorandom number generators. *Computer physics communications*, *60*(3), 329-344.

12. Oswald, D. F. (2016, October). Wireless attacks on automotive remote keyless entry systems. In *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices* (pp. 43-44).

13. Garcia, F. D., Oswald, D., Kasper, T., & Pavlidès, P. (2016). Lock It and Still Lose It—on the ({In) Security} of Automotive Remote Keyless Entry Systems. In *25th USENIX Security Symposium (USENIX Security 16)*.

14. Moradi, A., & Kasper, T. (2009, December). A new remote keyless entry system resistant to power analysis attacks. In *2009 7th International Conference on Information, Communications and Signal Processing (ICICS)* (pp. 1-6). IEEE.

15. Tang, J., & Tian, Y. (2016). A systematic review on minwise hashing algorithms. *Annals of Data Science*, *3*(4), 445-468.

16. Gupta, S., Goyal, N., & Aggarwal, K. (2014). A review of comparative study of md5 and ssh security algorithm. *International Journal of Computer Applications*, *104*(14).

17. Hamdan, Y. B. (2021). Smart home environment future challenges and issues-a survey. *Journal of Electronics*, *3*(01), 239-246.