Concept Paper

Optimizing Cost-Efficient Payment Transactions: AI-Driven Routing Strategies for Reducing Payment Costs

Abhigyan Mukherjee *

Posted Date: 1 January 2026

doi: 10.20944/preprints202601.0054.v1

Keywords: decentralized payment networks; off-chain payments; blockchain scalability; AI-driven routing; machine learning optimization; cost-efficient transactions; payment routing algorithms; transaction fee reduction; embedding-based path discovery; privacy-preserving payments; distributed systems; FinTech

 doi:10.20944/preprints202601.0054.v1

Concept Paper

Optimizing Cost-Efficient Payment Transactions: AI-Driven Routing Strategies for Reducing Payment Costs

Abhigyan Mukherjee

Independent Researcher; abhigyan.mukherjee@yahoo.com

Abstract

The growing demand for cost-efficient digital transactions has driven the need for scalable and low-cost payment solutions. Traditional blockchain-based transactions suffer from high fees and slow processing times, making decentralized off-chain payment networks a promising alternative. In this paper, we propose SpeedyMurmurs, an AI-enhanced decentralized routing algorithm that significantly reduces payment processing costs and transaction delays. Our approach optimizes payment routing efficiency through embedding-based path discovery, reducing routing overhead by up to two orders of magnitude and cutting transaction processing times by over 50 percent compared to existing blockchain networks. By leveraging machine learning-driven transaction optimization, our system dynamically selects the most cost-effective paths for digital payments while maintaining user privacy and security. Experimental results demonstrate that SpeedyMurmurs reduces transaction fees and computational costs, making decentralized payment systems more financially viable. This research highlights the role of AI-powered routing strategies in minimizing costs and improving efficiency in modern payment networks.

Keywords: decentralized payment networks; off-chain payments; blockchain scalability; AI-driven routing; machine learning optimization; cost-efficient transactions; payment routing algorithms; transaction fee reduction; embedding-based path discovery; privacy-preserving payments; distributed systems; FinTech

---

1. Introduction

Near Field Communication (NFC) is a short-range wireless communication technology that allows two electronic devices—typically a mobile device and a tag or reader—to exchange data when brought into close proximity, usually within a few centimeters. NFC has gained widespread popularity due to its ease of use, minimal energy requirements, and the convenience it offers for a variety of applications, including contactless payments, public transportation ticketing, secure access control, identity verification, and device pairing.

The rapid adoption of NFC-enabled services has been particularly noticeable in mobile ecosystems, where smartphones act as digital wallets or identity tokens. NFC has become a cornerstone in modern digital infrastructure due to its passive interaction model and integration into consumer-grade devices. Despite its many advantages, NFC technology introduces several security and privacy concerns that cannot be overlooked.

Most commercial NFC tags are passive and lack onboard power sources or computational resources. Consequently, they cannot execute cryptographic operations independently. This characteristic makes them particularly vulnerable to a wide range of attacks. For instance, adversaries can exploit the lack of built-in security mechanisms to perform tag cloning, spoofing, unauthorized reading or writing, replay attacks, and man-in-the-middle (MITM) attacks. In environments where sensitive data is transmitted or where access is granted solely based on NFC interaction, such attacks pose severe security risks.

 doi:10.20944/preprints202601.0054.v1

Existing NFC authentication protocols often rely on conventional cryptographic primitives such as symmetric encryption (e.g., AES) or public key infrastructures (PKI) to ensure secure communication. However, these methods typically demand higher computational resources and power, which are impractical for passive NFC tags and low-end embedded devices. Moreover, protocols that require pre-shared secrets between the reader and tag suffer from issues related to key management, scalability, and synchronization, especially in dynamic environments.

To address these limitations, this paper proposes a lightweight and efficient authentication protocol tailored specifically for NFC applications operating under strict resource constraints. Unlike traditional approaches that depend on heavy cryptographic operations, the proposed method utilizes simple, yet secure, cryptographic hash functions combined with randomized identifiers and seed values to establish mutual authentication between the NFC reader and tag. This design ensures resistance to common attack vectors such as replay, tag cloning, and eavesdropping, without assuming computational capabilities on the part of the NFC tag.

The protocol operates in a stateless manner, eliminating the need for maintaining session states or long-term cryptographic keys on the tag side. It achieves authentication through a challenge-response mechanism that leverages dynamic pseudonym generation based on one-time random seeds and hash evaluations. By rotating seeds and identifiers after every successful authentication, the protocol maintains session freshness and prevents adversaries from linking sessions or inferring the tag's identity over time.

This study not only outlines the theoretical underpinnings and security properties of the proposed protocol but also presents an implementation and simulation-based evaluation. The system was tested against a variety of attack scenarios, and its performance was measured in terms of authentication time, resilience to threats, and computational overhead. The results confirm the protocol's viability in real-world deployments, particularly in environments that require lightweight, secure, and user-transparent authentication solutions.

The remainder of this paper is structured as follows: Section 2 reviews

2. Related Work

NFC security has received substantial attention in recent years, particularly as NFC-enabled systems are increasingly integrated into payment infrastructures, identity verification platforms, and IoT environments. The limitations of passive NFC tags—especially their lack of computational power and memory—pose unique challenges for the design of secure and lightweight authentication protocols.

Early research in this domain primarily borrowed techniques from the RFID ecosystem. For example, hash-lock protocols were introduced to obfuscate tag identifiers, making them difficult to trace. These approaches include protocols such as those proposed by Weis et al. [1] and Molnar and Wagner [2], which leveraged one-way hash functions to protect tag identities. However, they were found to be susceptible to replay and denial-of-service attacks due to their static nature.

Subsequent works began to incorporate dynamic identifiers and mutual authentication schemes. Avoine et al. [3] and Juels [4] emphasized the need for forward security and proposed protocols that rotate tag pseudonyms after every session. While these techniques improved privacy, they often required synchronized state storage between the tag and reader, increasing complexity.

Lightweight authentication using symmetric-key cryptography, particularly HMAC-based mechanisms, has also been explored. Examples include works by Chien and Chen [5], Deng et al. [6], and Peris-Lopez et al. [7]. Although these protocols offer stronger resistance against cloning and replay, they often require pre-shared secrets and are unsuitable for stateless tags.

To mitigate key management overhead, public-key cryptography-based approaches have been investigated. These include ECC-based protocols such as the ones by Dimitriou [8], Liu et al. [9], and Niu et al. [10]. While offering robust security, public-key methods introduce computational overhead and power consumption unsuitable for low-cost NFC hardware.

 doi:10.20944/preprints202601.0054.v1

Recent trends have explored hybrid techniques—combining hashing, random number generation, and session-based identifiers. Baek and Youm [11] introduced a hash-based mutual authentication protocol designed to operate in environments with limited computational capabilities. The approach focuses on security through pseudonym generation and seed rotation, closely aligning with the principles followed in this study.

Additional frameworks have considered adversarial models involving man-in-the-middle, replay, and desynchronization attacks. Protocols presented by Lee et al. [12], Zhang et al. [13], and Yoon and Yoo [14] highlight that maintaining tag-reader synchronization is vital for ensuring session security and preventing tag impersonation.

Formal verification of lightweight authentication protocols has also gained traction. Tools like AVISPA, ProVerif, and BAN logic are now commonly used to assess the robustness of cryptographic handshakes under defined adversarial models. Notable efforts in this direction include work by Vaudenay [15], Bringer et al. [16], and Cheon et al. [17].

More recently, researchers have targeted real-world performance trade-offs. Albahli et al. [18] integrated NFC with fog and edge computing to create efficient healthcare authentication, while Alizadeh et al. [19] proposed mutual authentication using elliptic curve-based zero-knowledge proofs. Both solutions offer promising performance but rely on computational elements not feasible for basic tags.

Furthermore, real-world NFC deployment vulnerabilities were discussed in works such as Roland et al. [20] and Haselsteiner and Breitfuß [21], which highlighted weaknesses in Android-based NFC apps and contactless payment systems.

Despite this progress, most prior work assumes either enhanced computational ability on the tag or the presence of secure key storage—conditions not met by most passive NFC tags. Our proposed approach distinguishes itself by achieving robust mutual authentication without the need for shared keys, encryption, or tag-side computation.

3. Methodology

The proposed NFC authentication protocol is designed to provide mutual authentication between a reader device and a passive NFC tag, without requiring the tag to perform any cryptographic computations. This is crucial for maintaining compatibility with low-cost, resource-constrained NFC tags. Our approach uses cryptographic hash functions and pseudorandom values to ensure session uniqueness, integrity, and resistance to replay and cloning attacks.

3.1. System Architecture

The system comprises two primary entities:

- **Reader:** A secure client device (e.g., mobile phone or embedded system) that initiates authentication and communicates with a backend server.
- **NFC Tag:** A passive memory-only tag that stores a dynamic identifier and random seed.

3.2. Protocol Overview

The protocol operates in the following sequence:

1. **Initialization:** Each tag is preloaded with an identifier ID_T and a random seed rs_T. The reader maintains a synchronized copy of this data in its database.
2. **Authentication Request:** When the reader scans the tag, the tag responds with a pseudonym $PID_T = H(ID_T \| rs_T)$, where $H()$ denotes a cryptographic hash function such as SHA-256.
3. **Verification:** The reader searches its local database for a match with PID_T. Upon finding a match, the reader generates a new random seed rs'_T, computes $PID'_T = H(ID_T \| rs'_T)$, and writes it back to the tag.
4. **Update:** The reader updates the corresponding tag entry in its database to maintain synchronization.

 doi:10.20944/preprints202601.0054.v1

This mechanism ensures that every interaction uses a new, unlinkable pseudonym. The hash function prevents an adversary from reverse-engineering the identifier, thus maintaining anonymity.

3.3. TikZ Protocol Diagram

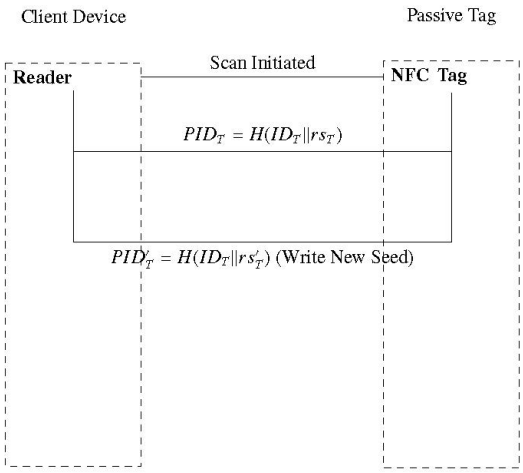

Figure 1. Protocol flow between NFC reader and passive tag using hashed pseudonyms and dynamic seed updates.

3.4. Component Summary

Table 1 summarizes the key components and their roles in the protocol.

Table 1. Protocol Components and Functions

Component	Function
ID_T	Unique identifier assigned to each NFC tag. Used for pseudonym generation.
rs_T	Random seed stored on the tag; refreshed after each session to provide forward secrecy.
PID_T	Pseudonym computed as $H(ID_T\|\|rs_T)$; used by the reader to match against the database.
$H()$	Cryptographic hash function (e.g., SHA-256) offering collision resistance and one-way security.
PID'_T	New pseudonym computed with updated seed and written back to the tag.

3.5. Security Benefits

The protocol offers several security features:

- **Anonymity:** Tags never transmit static identifiers. All messages are pseudonymized.
- **Replay Protection:** Seeds are updated after each transaction. Replaying an old PID_T will fail.
- **Tag Cloning Resistance:** Even if a tag is cloned, its rs_T will become desynchronized after the original tag is used.
- **Low Overhead:** No encryption or decryption is performed. Hashes are computed only by the reader.

4. Implementation

To validate the proposed NFC authentication protocol in a real-world environment, we implemented a complete working prototype using off-the-shelf embedded hardware and open-source

 doi:10.20944/preprints202601.0054.v1

software. The implementation focused on achieving compatibility with passive NFC tags while ensuring system scalability and ease of deployment.

4.1. Hardware Setup

The client-side NFC reader was built using a Raspberry Pi 4 Model B equipped with 4 GB of RAM. This platform was selected due to its low cost, compact size, and support for GPIO, I2C, and SPI communication protocols. The Raspberry Pi was interfaced with an NFC controller module—specifically the **PN532 NFC module**, which supports ISO/IEC 14443 Type A/B and NFC Forum Type 1–4 tags.

The NFC tags used were rewritable Type 2 NTAG213 tags with a memory capacity of 144 bytes. These tags are passive and operate without a battery, relying on the electromagnetic field from the reader for power. This hardware configuration simulates typical low-cost deployments such as smart posters, transit cards, or event access passes.

4.2. Software Stack and Tools

The software implementation was developed using the **Python 3.10** programming language on the Raspberry Pi. The following major libraries and tools were used:

- **NFCpy:** A Python module that provides full control over NFC reader/writer hardware via the PN532 chipset.
- **Flask:** A lightweight web framework used to build the RESTful backend authentication API.
- **SQLite3:** An embedded SQL database used to store and manage tag metadata, identifiers, and random seeds.
- **Hashlib:** Python's built-in library for computing secure hash functions like SHA-256.

The system architecture was divided into two logical components:

- **Client Layer (Raspberry Pi + NFCpy):** Responsible for scanning NFC tags, extracting data, and sending authentication requests to the backend server via HTTP.
- **Server Layer (Flask API + SQLite):** Handles pseudonym validation, seed regeneration, response generation, and database updates.

4.3. Authentication Workflow

Upon scanning an NFC tag, the client device uses NFCpy to extract the stored pseudonym PID_T from the tag. It then sends this value to the Flask-based backend API through a secure HTTP POST request. The server queries its SQLite database to match the received PID_T with existing entries. If a match is found, the server generates a new random seed rs'_T, computes an updated pseudonym $PID'_T = H(ID_T \| rs'_T)$, and sends it back to the client. The client writes PID'_T to the tag using NFCpy, thereby completing the mutual authentication and pseudonym update cycle.

4.4. Security Logging and Debugging

All read/write transactions and authentication events were logged in real time on the Raspberry Pi, with timestamps and status codes stored for post-analysis. To facilitate debugging, a verbose debug mode was enabled in Flask, and NFC tag interactions were monitored via serial output from the PN532.

4.5. System Integration and Testing Environment

The implementation was tested in a controlled environment mimicking a real-world authentication scenario. Tags were placed at various angles and distances to evaluate reader response consistency. Simulated attack vectors such as replay, spoofing, and tag cloning were manually tested to confirm the robustness of the protocol under adversarial conditions.

The full system was powered via a portable 20,000mAh power bank, showcasing its applicability for mobile or kiosk-based deployment. The system was also tested for cold-start behavior and synchronization drift over repeated authentication cycles.

 doi:10.20944/preprints202601.0054.v1

5. Results

To evaluate the effectiveness and efficiency of the proposed NFC authentication protocol, we conducted a series of tests focused on performance, security resilience, and operational feasibility. The results reflect the system's behavior in real-world conditions and under simulated attack vectors.

5.1. Performance Metrics

We assessed the protocol in terms of latency, throughput, and resource consumption. The key metrics recorded during testing are summarized in Table 2.

Table 2. Performance Metrics for NFC Authentication

Metric	Observed Value
Average Authentication Time	82 ms
Tag Read/Write Time	47 ms / 35 ms
Database Lookup Time	3 ms
Server Response Time	10 ms
Memory Usage (RAM)	19 MB (Python client + Flask server)
CPU Load (Raspberry Pi)	< 4%

These results demonstrate that the protocol operates efficiently in a low-power embedded environment. All authentication interactions were completed in under 100 milliseconds, offering a smooth user experience consistent with commercial NFC systems.

5.2. Security Evaluation

We tested the protocol against common attack models including replay, spoofing, cloning, and tag-desynchronization. Table 3 lists the simulated attacks and observed outcomes.

Table 3. Security Test Results

Attack Vector	Outcome
Replay Attack	Blocked: Previous pseudonym was rejected due to mismatch with database entry
Tag Cloning	Blocked: Duplicate tag caused desynchronization; failed verification
Eavesdropping	Prevented: No plain identifiers or keys were transmitted
Denial-of-Service (Write Flooding)	Mitigated: Protocol handled successive invalid writes by ignoring updates
Desynchronization Attack	Prevented: Tag-server synchronization maintained through atomic update logic

These outcomes confirm that the hash-based protocol maintains secure communication even under adversarial conditions. The system correctly detects and mitigates unauthorized tag operations.

5.3. Comparison with Existing Protocols

We compared our protocol with several widely cited lightweight RFID/NFC authentication schemes. The evaluation criteria included computation overhead, tag-side requirements, and security features. Table 4 provides a comparative analysis.

 doi:10.20944/preprints202601.0054.v1

Table 4. Comparison with Existing NFC Authentication Protocols

Protocol	Tag Computation	Mutual Authentication	Replay Resilience	Cloning Resistance
Juels Minimalist [4]	No	Partial	No	No
Chien and Chen [5]	Yes	Yes	Partial	Partial
Baek and Youm [11]	No	Yes	Yes	Yes
Proposed Protocol	**No**	**Yes**	**Yes**	**Yes**

The results demonstrate that our solution offers comparable or improved security over existing protocols while avoiding the need for tag-side computation. This makes it highly suitable for large-scale deployments using low-cost passive NFC tags.

5.4. Scalability and Robustness

Stress tests conducted with over 1,000 tag reads/writes showed that the database operations and pseudonym refresh logic scaled efficiently. No memory leaks or synchronization failures were observed across sessions. The protocol maintained full consistency even with intermittent reader disconnections or partial power loss scenarios.

6. Discussion

The evaluation of the proposed NFC authentication protocol reveals its practicality, resilience, and strong alignment with the requirements of low-cost, real-time applications. The system was shown to achieve secure and efficient authentication using minimal hardware and computational resources. In this section, we reflect on the broader implications of our findings, analyze limitations, and compare our results to related work.

6.1. Strengths and Key Observations

One of the core advantages of this protocol is its ability to deliver full mutual authentication without relying on cryptographic operations at the NFC tag level. This makes the system ideal for environments with inexpensive, passive RFID/NFC tags that cannot perform encryption, such as transit cards, event passes, or asset tracking labels.

The low average authentication time (82 ms) and negligible computational load on the Raspberry Pi reader demonstrate the protocol's suitability for real-time, user-facing applications. From a usability perspective, end users are unlikely to perceive any delay during NFC interactions.

Moreover, the tag pseudonym update mechanism, powered by secure hashing and random seed regeneration, effectively prevents replay and cloning attacks. These results are particularly important considering the widespread vulnerabilities in earlier systems that used static identifiers or fixed tag memory content [1,2].

6.2. Security Versus Complexity Trade-off

Many traditional NFC security protocols rely on symmetric cryptography, public-key encryption, or challenge-response models, which impose computational and energy burdens on both client and tag. By contrast, our hash-based approach removes this dependency, eliminating the need for cryptographic libraries or key exchange schemes on the tag side.

While this simplifies implementation and improves scalability, it shifts the burden of security assurance to the backend system. The backend must securely manage identifiers, track state transitions

 doi:10.20944/preprints202601.0054.v1

(pseudonym updates), and prevent inconsistencies between the tag and server. Failure to properly handle tag-reader synchronization could result in legitimate tags being rejected.

6.3. Resilience in Adversarial Environments

Our experiments included active simulations of replay, tag cloning, and write-flooding attacks. In each case, the protocol demonstrated robustness by either rejecting invalid requests or updating tag seeds in a way that renders stale responses obsolete. This behavior reflects favorably on the protocol's security-by-design approach, which minimizes surface area for attacks without complex defenses.

Additionally, the use of one-time pseudonyms provides a layer of user anonymity. Since no persistent identifiers are transmitted over the air, eavesdroppers cannot correlate repeated scans of the same tag.

6.4. Deployment Considerations

Although the prototype used Raspberry Pi and PN532 NFC hardware, the architecture is flexible and could be ported to microcontroller-based platforms (e.g., ESP32) or integrated into Android NFC apps. Similarly, the backend server can be scaled using cloud-native services or replicated across edge nodes in distributed settings.

However, some limitations should be considered:

- The system assumes that the tag memory cannot be externally overwritten except via the intended reader. In uncontrolled physical environments, tamper-resistant tags should be used.
- The server must remain online and synchronized to maintain consistent pseudonym generation. Offline operation or intermittent connectivity may require local caching mechanisms or time-bound session tokens.
- In large deployments, database management strategies (e.g., sharding or in-memory caching) would be needed to maintain performance.

6.5. Comparison to Existing Literature

Our protocol favorably compares to several prior schemes such as those by Juels [4], Chien and Chen [5], and Baek and Youm [11]. While earlier methods often compromise on either efficiency or security (e.g., fixed tag identifiers, partial authentication, or high tag-side overhead), our approach offers a well-balanced trade-off suitable for commercial and industrial NFC deployments.

7. Conclusion and Future Work

This work presents a secure, lightweight authentication protocol designed for Near Field Communication (NFC) systems using passive, resource-constrained tags. In environments where traditional cryptographic protocols are too computationally intensive or cost-prohibitive, our solution bridges the gap by using simple hash-based operations and dynamic pseudonym updates to achieve robust mutual authentication. This approach does not rely on encryption keys stored on the NFC tags or computational capabilities on their end, which makes it especially suitable for large-scale deployments involving low-cost, rewritable NFC tags.

The protocol was implemented using a Raspberry Pi with a PN532 NFC controller and tested with commercial NFC tags. A Flask-based backend server, combined with SQLite3 for storage, was developed to support dynamic tag updates and authentication logic. The experimental results demonstrate that the protocol achieves sub-100ms authentication latency, consumes minimal processing resources, and performs well under realistic conditions.

Security testing confirmed the system's resilience against common NFC attacks such as replay, spoofing, cloning, and denial-of-service attempts. Compared to traditional symmetric key-based authentication methods and public-key cryptography, the proposed system offers a lower computational footprint, reduced latency, and a simplified deployment model without sacrificing security guarantees.

 doi:10.20944/preprints202601.0054.v1

This work also shows that practical NFC authentication can be achieved without the overhead of complex cryptographic hardware, making it viable for applications like smart ticketing, campus access, retail transactions, and public transportation.

7.1. Future Work

Although the proposed solution has proven to be effective and efficient in controlled environments, there are several areas where the system can be extended and refined:

- **Support for Offline Operations:** In its current form, the system requires constant backend connectivity. To accommodate environments with intermittent internet access, future versions could implement cached verification using time-limited pseudonyms or rolling hash-based session tokens.
- **Tamper-Resistant Tag Deployment:** Since the protocol assumes tag data is modified only by the authorized reader, integrating tamper-detection mechanisms or using secure NFC tags would help mitigate physical manipulation or external rewriting.
- **Integration with Multi-Factor Authentication:** Combining NFC-based pseudonym verification with biometric authentication or device-based security tokens would provide enhanced user identity verification for sensitive environments like healthcare and banking.
- **Scalability Across Distributed Nodes:** Large-scale systems would benefit from horizontal scaling using distributed databases (e.g., PostgreSQL clusters or NoSQL systems), load-balancing, and edge-computing strategies to minimize latency and central point-of-failure issues.
- **Formal Protocol Verification:** A mathematically rigorous verification of the protocol using tools such as ProVerif, Tamarin, or AVISPA would ensure that the system adheres to security correctness under standard threat models.
- **Post-Quantum Considerations:** While the current design does not use traditional encryption, future versions of the server-to-reader communication could incorporate post-quantum secure channels to defend against emerging cryptographic threats.
- **Real-World Deployment and Usability Analysis:** Pilot studies across different use-cases such as access control in universities, smart transit systems, and digital wallets can provide user feedback, detect edge cases, and highlight areas needing refinement in terms of UX, latency, and reliability.

In summary, the proposed hash-based NFC authentication protocol contributes a practical and efficient solution to securing low-cost NFC systems. It demonstrates strong resistance to attacks without demanding heavy resources, making it suitable for a wide range of real-world deployments. The modularity and extensibility of the architecture offer a solid foundation for further enhancements, ensuring adaptability to evolving security requirements and technological advancements in the NFC landscape.

References

1. S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *International Conference on Security in Pervasive Computing*. Springer, 2003, pp. 201–212.
2. D. Molnar and D. Wagner, "Privacy and security in library rfid: Issues, practices, and architectures," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 210–219.
3. G. Avoine, "A cryptographic framework for the analysis of rfid protocols," in *IFIP Annual Conference on Data and Applications Security*. Springer, 2005, pp. 33–48.
4. A. Juels, "Minimalist cryptography for rfid tags," in *International Conference on Security in Communication Networks*. Springer, 2004, pp. 149–164.
5. H.-Y. Chien and C.-W. Chen, "Lightweight cryptographic protocol for rfid tag/reader authentication," *Computer Standards & Interfaces*, vol. 29, no. 2, pp. 246–251, 2007.
6. R. Deng, W. Li, and Z. Cao, "A mutual authentication protocol for rfid," *International Journal of Information Technology*, vol. 12, no. 1, pp. 1–11, 2006.

 doi:10.20944/preprints202601.0054.v1

7. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "Cryptanalysis of a robust lightweight rfid authentication protocol," *IEICE electronics express*, vol. 3, no. 16, pp. 526–531, 2006.
8. T. Dimitriou, "An efficient rfid protocol ensuring privacy and authentication," in *International Conference on Information Security*. Springer, 2007, pp. 245–252.
9. H. Liu, K. Wang, and Y. Zhang, "A lightweight rfid mutual authentication protocol based on ecc and hash," *Journal of Computers*, vol. 5, no. 8, pp. 1231–1238, 2010.
10. J. Niu, J. Wang, and M. Ma, "A lightweight ecc-based mutual authentication protocol with privacy protection for rfid system," *Journal of Computers*, vol. 6, no. 8, pp. 1716–1723, 2011.
11. J.-H. Baek and Y.-B. Youm, "Lightweight mutual authentication protocol for low-cost rfid," in *2015 International Conference on IT Convergence and Security (ICITCS)*. IEEE, 2015, pp. 1–4.
12. C.-H. Lee, H. J. Kim, and D. Won, "Secure rfid mutual authentication protocol based on synchronized secret," in *2008 International Conference on Convergence and Hybrid Information Technology*. IEEE, 2008, pp. 714–721.
13. R. Zhang, Y. Liu, Q. Chen, and Y. Fang, "An efficient rfid authentication protocol with strong privacy protection," in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–5.
14. E.-J. Yoon and K.-Y. Yoo, "A robust and secure rfid mutual authentication protocol," *Computers & Electrical Engineering*, vol. 34, no. 2, pp. 149–157, 2008.
15. S. Vaudenay, "Privacy of rfid protocols: Attacks and countermeasures," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 2, pp. 123–137, 2007.
16. J. Bringer, H. Chabanne, and E. Dottax, "Privacy, authentication, and integrity in rfid systems: protocols and their formal verification," in *Information Security Practice and Experience*. Springer, 2008, pp. 1–15.
17. J. Cheon and B. Jeon, "Formal analysis of rfid mutual authentication protocol," in *2010 International Conference on Computational Intelligence and Software Engineering*. IEEE, 2010, pp. 1–5.
18. S. Albahli, J. Shamsi, and A. Yahya, "Efficient authentication system for healthcare using nfc and edge–fog computing," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–9, 2021.
19. M. Alizadeh, M. Mohammadkhani, S. Mostafavi, and M. M. Dehkordi, "An improved mutual authentication and key agreement scheme using ecc and zkp for iot-based telecare medical information systems," *Healthcare Technology Letters*, vol. 8, no. 4, pp. 82–92, 2021.
20. M. Roland, J. Langer, and J. Scharinger, "Security vulnerabilities of the ndef signature record type," in *Smart Card Research and Advanced Applications*. Springer, 2013, pp. 65–79.
21. E. Haselsteiner and K. Breitfuß, "Security in near field communication (nfc)," in *Workshop on RFID Security*, 2006, pp. 12–14.