IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 3, June, 2023, pp. 761-772
Journal homepage: http://iieta.org/journals/ria

A Novel Approach for Ticket Generation and Validation using RSA and Keccak Algorithms

Rachna K. Somkunwar*, Sara Nawghare, Zarina Shaikh

Department of Computer Engineering, Dr. D.Y.Patil Institute of Technology, Pimpri, Pune 18, India

Corresponding Author Email: rachnasomkunwar12@gmail.com

https://doi.org/10.18280/ria.370325

Received: 4 May 2023
Accepted: 28 May 2023

Keywords:
ticket generation, ticket validation, RSA algorithm, security, keccak algorithm

ABSTRACT

The motivation for this research arises from the challenges faced by railway operators in managing ticketing processes effectively. These challenges highlight the need for a robust web-based application to automate ticket verification and validation, emphasizing the importance of developing a secure and efficient ticket generation and validation system. The proposed solution employs RSA (Rivest-Shamir-Adleman) and Keccak cryptographic algorithms to ensure the security and efficiency of ticket generation and validation. By generating digital signatures and hash functions, the authenticity and integrity of ticket data are maintained. Recipients can utilize the sender's public key and the same hash function to verify the ticket data's authenticity. The system offers several advantages, including security, integrity, authentication, efficiency, and scalability. Future work may involve implementing multi-party computation, developing more efficient algorithms, exploring blockchain technology, and conducting more extensive testing and evaluation.

1. INTRODUCTION

Train conductors are tasked with manually examining each ticket, presenting several challenges. Additionally, passengers must carry personal identification to board the train. The conductor is responsible for the entire ticket verification process and managing any seat openings that may arise during the journey [1]. Ticket generation and validation systems find applications in various domains, including event management, transportation, and access control [2]. These systems are designed to generate and validate tickets, ensuring their authenticity and integrity. However, traditional ticket generation and validation systems might lack sufficient security to prevent fraudulent activities.

This research aims to develop a secure and efficient ticket generation and validation system using cryptographic techniques. Specifically, the study proposes the utilization of RSA and Keccak algorithms for ticket generation and validation. RSA is a widely adopted public-key cryptography algorithm for secure communication and digital signatures. Keccak, on the other hand, is a cryptographic hash function renowned for its high security and resistance to collision attacks. By combining these two algorithms, the goal is to establish a secure, reliable, and efficient ticket generation and validation system.

Overview of the problem and motivation:

The ticket generation and validation system is essential to many events and services, such as concerts, sports events, public transportation, and online services. The traditional paper-based ticketing system has many limitations, such as the possibility of counterfeiting, fraud, and inconvenience for users. Therefore, there is a need for a secure and efficient ticket generation and validation system that can prevent such issues. To address this problem, this research paper proposes a ticket generation and validation system using RSA and Keccak. RSA is a popular public-key cryptographic algorithm that provides confidentiality, integrity, and data authentication. Keccak is a cryptographic hash function that produces a fixed-length output, which makes it ideal for generating unique identifiers and signatures [3].

This research paper aims to develop a ticket generation and validation system that is secure, efficient, and easy to implement. Proposed system aims to provide a safe method for generating and validating tickets, which can prevent counterfeiting, fraud, and other security threats. Furthermore, this system can also provide convenience for users, such as faster validation times and easier access to events and services.

RSA and Keccak Algorithms:

RSA is a public-key cryptographic algorithm widely used for secure communication and digital signature. RSA is based on the principle that it is easy to multiply two large prime numbers, but it isn't easy to factor the product into its original prime factors. RSA uses this principle to generate a pair of keys: a public key that can be shared with anyone and a private key that must be kept secret. The public key can be used to encrypt a message, and the private key can be used to decrypt the message. Similarly, the private key can be used to sign a message, and the public key can be used to verify the signature [4].

Keccak [5] is a cryptographic hash function selected as the winner of the NIST hash function competition in 2012. Keccak is known for its high level of security and resistance to collision attacks. Keccak operates on a fixed-length input and produces a fixed-length output, typically called a hash value or digest. Keccak can be used for various applications, including message authentication, digital signature, and key derivation. Keccak is designed to be computationally efficient and can be implemented in hardware or software.

Importance of the ticket generation and validation system:

The ticket generation and validation system is essential in many applications, particularly those requiring secure access

control, event management, and transportation. Here are some reasons why a ticket generation and validation system is essential:

Prevents Fraud: Ticket generation and validation system can prevent fraudulent activities. It protects the system's integrity and maintains the users' trust.

Improves Security: By incorporating cryptographic techniques, such as RSA and Keccak, into the ticket generation and validation system, the system becomes more secure and less vulnerable to attacks. Using these cryptographic algorithms can ensure that the tickets are tamper-proof and cannot be modified or duplicated.

Efficient Management: A ticket generation and validation system can help to streamline the ticketing process and make it more efficient. This is particularly useful when many tickets need to be issued and validated, such as in transportation systems or significant events.

Real-time Tracking: A ticket generation and validation system can provide real-time tracking of ticket usage, which can help monitor the system's capacity and identify areas where improvements can be made. It can also help to identify and prevent potential security breaches before they occur.

Enhances Customer Experience: A well-designed ticket generation and validation system can enhance the customer experience by providing a seamless and efficient ticket issuance and validation process. This can help to reduce wait times and improve overall satisfaction. The security benefits of digital signature are shown in Figure 1.

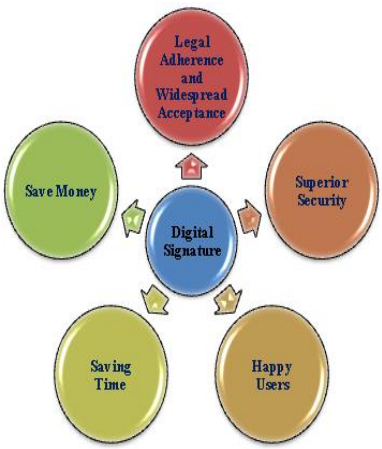

Figure 1. Security benefits of digital signatures

The paper's contribution can vary depending on the specific research and goals of the proposed ticket generation and validation system using RSA and Keccak.

1. The proposed system combines the RSA algorithm for digital signatures and the Keccak algorithm for hashing to ensure secure and reliable ticket generation and validation. This integration takes advantage of the strengths of both algorithms to enhance the overall security and integrity of the system.

2. The proposed system enhances the security of ticket generation and validation by leveraging RSA's robust encryption and digital signature capabilities. Using the Keccak hashing algorithm further ensures the integrity of the ticket data and prevents tampering. Combining these techniques, the system protects against unauthorized access, forgery, and data manipulation.

3. An efficient implementation of the ticket generation and validation system. By leveraging the computational efficiency of RSA and the fast hashing capabilities of Keccak, the system can handle many ticket requests on time.

4. The paper comprehensively evaluates the proposed system, including performance metrics, security analysis, and usability assessment. Through quantitative and qualitative research, the article provides empirical evidence of the system's effectiveness, highlighting its advantages and potential areas for improvement.

2. EXISTING METHODS

2.1 Review of existing ticket generation and validation systems

Existing ticket generation and validation systems can be broadly categorized into traditional and digital ticketing systems. Traditional ticketing techniques involve physical tickets, whereas digital ticketing systems rely on electronic tickets. Traditional ticketing systems typically involve printing physical tickets that are distributed to customers or sold at the event venue. The tickets may have unique serial numbers or barcodes that can be used for validation at the entrance. These systems are simple and inexpensive but are prone to counterfeiting and fraud. Digital ticketing systems involve generating electronic tickets that can be stored on a customer's mobile device or computer. These systems can facilitate validation by utilizing various technologies such as Quick Response (QR) codes [6], Near Field Communication (NFC), and Radio-Frequency Identification (RFID). Digital ticketing systems offer several advantages over traditional systems, including increased security, convenience, and real-time tracking. However, digital ticketing systems also face several challenges, such as the risk of hacking, fraud, and privacy concerns. Various cryptographic techniques have been proposed to address these challenges, such as digital signatures, public-key cryptography, and blockchain technology [7].

2.2 Comparison of different cryptographic techniques for ticket generation and validation

Several cryptographic techniques can be used for ticket generation and validation. Here, we compare some of the most commonly used methods:

RSA: RSA is a widely used public-key cryptography algorithm for secure communication and digital signatures. RSA is well-suited for ticket generation and validation because it provides a way to encrypt securely and decrypt data. RSA [8] can ensure the tickets are tamper-proof and cannot be modified or duplicated. However, RSA can be computationally expensive, especially for large key sizes.

Advanced Encryption Standard(AES): AES is a symmetric encryption algorithm used to encrypt and decrypt data [9]. AES is often combined with RSA to provide an additional layer of security. AES is fast and efficient, making it well-suited for ticket generation and validation systems. However, AES requires a shared secret key, which can be challenging to manage in large systems.

Keccak: Keccak [10] is a cryptographic hash function known for its high level of security and resistance to collision attacks. Keccak can be used to generate unique ticket IDs that can be used for validation. However, Keccak does not provide a way to encrypt and decrypt data, so it cannot be used

independently for secure communication.

Elliptic Curve Cryptography (ECC): ECC is a public-key cryptography algorithm similar to RSA but more efficient for use in resource-constrained environments. ECC can provide a high level of security with smaller key sizes compared to RSA. ECC [11] can be used for ticket generation and validation, but it may not be as widely supported as RSA or AES.

Digital Signatures: Digital signatures [12] are used to verify the authenticity of a message or document. Digital signatures can be used in ticket generation and validation to ensure that the tickets are not forged or modified. Digital signatures can be used with any of the above cryptographic techniques to provide an additional layer of security.

The main goal of this paper [13] is to make travelling much more relaxing by eradicating one of the most common concerns concerning having a ticket with us when we are traveling. They recommend a biometric-based ticketing system for this purpose; however, it is not restricted to the metro railway scenario. They considered each person's fingerprint when they registered, bought tickets, and confirmed the fingerprint on the day of travel so he or she could travel on a certain day and the chosen train to their selected destination. The fingerprint sensor will connect to Arduino, and Arduino will then store the fingerprint information in the cloud.

In the study [14], authors proposed an automated system for confirming tickets. Passengers only wish to display their tickets in front of the camera. The validation status is returned when the QR code has been scanned. As a result, it is beneficial for the railway system, which efficiently serves reserved passengers while the information of the passengers has been cross-checked, reducing the workload of the TTE (Train Ticket Examiner), and regulating the passenger to scan. As a result, it reflects the unreserved passengers who board a train without reservation, which will benefit reserved tickets. Because this system can function without the assistance of a TTE, passengers' monitoring is also high. Therefore, manual monitoring must ensure that passengers travel safely and identify unreserved passengers using system counts.

In the suggested approach [15], a passenger logs into their account on the train website. If the traveler has no account, he must register using personal data. The user logs into the website and selects the ticket button to create a ticket. The ticket generating button is deactivated if every seat is taken. The traveler chooses the date, source, and destination. The website fetches some data after the route is chosen and presents the various ticket options and associated costs. The customer chooses the type of ticket, pays for it with a bank account or railway card code, and downloads the QR code ticket. The users enter their QR codes into a device to be scanned and then go along the path. The travelers scan their code once more to confirm the ticket route when they arrive at their destination. The system rejects a ticket whose price does not match the actual price, and the passengers are penalized severely. On the other side, users can generate their tickets using machines. The suggested approach offers the ability to use both a machine and an app application.

In the study [16], authors have built a secure transit system to allow each passenger to check themselves into their assigned train seats via a QR code. A biometric-based authentication system is used to implement an easy way to confirm their identification. Also, the authors offered a convenient way to purchase train tickets for use in train travel. It is simpler to confirm who is sitting in a particular seat on a train by using this technique and are also able to put in place an automated system that distributes any open or empty seat to the subsequent group of passengers on the waiting list, starting at the top of the list.

The current system allows us to purchase offline and online tickets, but we frequently do not do so due to black marketing and the production of paper tickets with carbon printing. Additionally, maintaining passenger records during the ticket-checking process requires a lot of tedious manual work. This paper offers a solution to these issues by controlling the ticket booking process using an Aadhaar card number or fingerprint, decreasing the need for carbon-printed paper tickets and paper waste. An Android app is created that will assist the ticket checker in efficiently checking tickets and maintaining records [17].

India's metro system serves as the public transportation for some major cities. Every day, 1.5 million people use the metro services in Delhi alone. The metro now uses token and smart-card-based ticketing systems, which use the Near-Field Communication Technology principle. This study discusses the potential use of a facial recognition-based alternative ticketing mode for metro riders. It also demonstrates show the user could control the own metro account. It looks for a ticketing system that would eliminate the requirement for a physical transportation medium. It makes an effort to find a ticketing system that wouldn't require a actual medium in order to use the metro rail [18].

The article summarizes the outcomes of the complex system study for the digital railway control system. It is clarified how the digital economy and the digital control system are related. In relation to the digital management of the railway, the article discusses the twelve key aspects of the digital economy. The study provides evidence that all 12 tenets of the digital economy are evolving into the tenets of digital railway management. It is illustrated where the digital railway control system stands in relation to other systems [19].

The impact of the online ticketing system, socioeconomic position, and perceived control on passengers' wellbeing during SFTR is examined in this study using a structural equation model. The outcomes demonstrate that the online mechanism for purchasing tickets directly improves the welfare of travelers. Using an online system for purchasing tickets and perceived control play a significant mediating role between socioeconomic status and the wellbeing of Spring Festival passengers. Particularly, the online system for purchasing tickets had made it simpler for people of lower socioeconomic status to do so, greatly improving their wellbeing [20].

The methods used for evaluating real ridership, including as technology and procedures, have all been examined in this research. Also, this study looked into the possibility of describing travel patterns, including train travel, using data from mobile phones. They discovered that there is a group of technologies that can be utilized for finding train numbers of passengers [21].

Innovative AES operations, such as Elliptical Curve Cryptography (ECC-based) S-box, SM-based mix column, and BEDT schemes for adding round keys, were created because of the research work done. After some time had passed, these strategies were put into action with the aid of white-box cryptography systems. BWMC is used to find weaknesses and implement fixes for those problems during the S-box generation process, incorporating ECC. One disadvantage is that memory must do more redundant bit

operations to maintain reliability [22].

This work suggests using Covid-19 record sharing and privacy-preserving storage to check Covid-19 status while issuing trip tickets. The suggested system is used to construct the Inter Planetary File System (IPFS), Chaotic Map, Paillier Cryptosystem, and cryptographic hash algorithm SHA-256. Because User_Ids are hashed, and passwords are AES-256 encrypted, the contents of the Authentication Data Table (ADT) in secure authentication are well protected from brute-force attacks, dictionary attacks, advanced dictionary attacks, lookup table attacks, and rainbow table attacks. The hashed input User_Id and Password are used to create an encryption key that is specific to each user. The suggested approach also includes the development of novel image encryption. The control values of the key streams needed during the confusion-diffusion of the proposed cryptosystem are obtained from the input Patient Id and Covid-19 record to boost security and survive known plain text/chosen plain text attacks.

Drawback of existing systems are to address the issues such as inadequate security measures, lack of scalability, inefficient validation processes, or the absence of a comprehensive integration of cryptographic techniques. By addressing these limitations, the proposed system will contribute to advancing ticket generation and validation practices and provide an improved solution for the identified gaps in existing research.

3. PROPOSED METHODOLOGY

3.1 Overview of the proposed system

The proposed ticket generation and validation system using RSA and Keccak is a secure and efficient solution for generating and validating tickets. The system utilizes two well-established cryptographic algorithms, RSA and Keccak, to ensure the authenticity and integrity of ticket data. The system consists of two main components: the ticket generation component and the ticket validation component.

The ticket generation component generates unique tickets using RSA digital signatures. The system generates a private and public key using RSA, then uses the private key to generate a digital signature for each ticket. The digital signature ensures the authenticity of the ticket data and can be verified using the corresponding public key.

The ticket validation component is responsible for validating tickets using Keccak hash functions. The system generates a unique hash value for each ticket using Keccak, ensuring the ticket data's integrity. The hash value can be easily verified by comparing it to the corresponding hash value stored in a centralized database.

The proposed system provides a scalable and efficient solution for generating and validating tickets, making it suitable for various events and applications. It offers a high level of security by utilizing well-established cryptographic algorithms, ensuring the authenticity and integrity of ticket data. The system can be easily integrated with other ticketing systems, offering a seamless solution for event organizers and ticket distributors. Figure 2 represents the ticket generation and validation system.

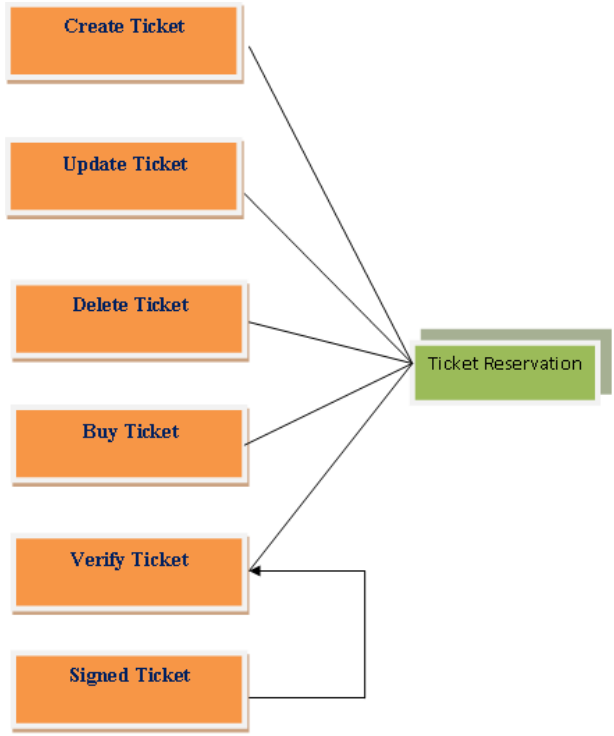

Figure 2. Ticket generation and validation system

3.2 Ticket generation process and Ticket validation process

The ticket generation process using RSA and Keccak involves the following steps:

RSA Key Generation:

1. Let p and q be two large prime numbers.
2. Compute the modulus n=p*q.
3. Compute Euler's totient function phi(n)=(p-1)(q-1).
4. Choose an integer e, such that 1<e<phi(n) and gcd(e, phi(n))=1. This is the public key.
5. Compute d, the modular multiplicative inverse of e mod phi(n). This is the private key.

Ticket Generation:

Let T be the ticket data. Generate a random number r. Compute the hash value H(T||r) using the Keccak algorithm. Here, || denotes concatenation. Encrypt the hash value using the sender's private key. The resulting cipher text is the digital signature, S=H(T||r)^d mod n.

Ticket Validation:

1. To validate the ticket, the following steps are performed:
2. Decrypt the digital signature using the sender's public key. This gives the hash value H(T||r)=S^e mod n.
3. Re-compute the hash value H(T||r) using the Keccak algorithm. Verify that the recomputed hash value matches the decrypted hash value. If the hash values match, the ticket is valid. The security of the ticket generation and validation process depends on the strength of the RSA key pair and the Keccak hash function, and the larger the hash function's key size and output size, the more secure the process.

After generating the encrypted ticket using RSA, it is passed to the Keccak component for validation. The Keccak component receives the encrypted ticket, computes the hash value, and compares it with the stored hash value. If the hash values match, the ticket is considered valid; otherwise, it is rejected.

The Ticket Generation and validation is shown in Figure 3.

Algorithm1: Proposed Algorithm for Ticket Generation and Validation
Input: n (number of tickets)
Output: tickets and their corresponding signatures
Generate RSA key pair: (e, d, n), where e is the public key exponent and d is the private key exponent.
For $i = 1$ to n
Generate a random ticket ID: ID_i
Compute the hash value of ID: $hash_i = \text{Keccak}(ID_i)$
Generate a digital signature using RSA: $signature\_i = RSA\_Sign\ (hash\_i, d, n)$
End
Store the ticket IDs, hash values, and signatures in the database
return Valid tickets and their corresponding signatures

Figure 3. Ticket generation and validation

Advantages of RSA:

- Security: RSA offers strong security through its asymmetric encryption scheme. Public and private keys ensure confidentiality; only the private key holder can decrypt the ticket information.

- Authenticity: The digital signature feature of RSA allows for ticket authenticity verification. By signing the ticket information with the private key, the recipient can validate the ticket's origin and ensure it hasn't been tampered with.

- Integrity: RSA can ensure the integrity of the ticket information by encrypting it with the private key, which can only be decrypted with the corresponding public key. Any modification to the ticket would result in decryption failure, indicating tampering.

Advantages of Keccak:

- Security: Keccak is a secure hashing algorithm that offers resistance against various cryptographic attacks. It ensures the integrity of the ticket information by generating a unique hash value for verification purposes.

- Efficiency: Keccak provides high-speed hashing capabilities, making it suitable for efficient ticket validation. Its sponge construction allows for flexible output lengths, efficiently handling different ticket sizes.

- Compactness: Keccak produces fixed-length hash values regardless of the input size, resulting in compact representations of the ticket information.

Limitations and Overheads:

- Computational Overhead: RSA involves computationally expensive operations, such as modular exponentiation, which can be time-consuming for large ticket sizes or a high volume of ticket transactions.

- Storage Overhead: RSA requires the storage of public and private keys, which can be space-consuming, especially in scenarios with many users.

- Key Management: RSA requires proper key management to ensure the security and integrity of the system. This includes securely generating and storing the key pairs and managing key distribution and revocation.

Feasibility of Proposed System:

Programming Language: The system will use a high-level programming language like Python or Java. The choice of programming language will depend on factors such as developer expertise, system requirements, and compatibility with the selected frameworks and libraries.

Software/Hardware Requirements: The system will require a suitable development environment, including a text editor or integrated development environment (IDE) for coding, a compiler or interpreter for executing the code, and relevant software libraries and frameworks for implementing the cryptographic algorithms (RSA and Keccak). The hardware requirements will depend on the expected system load, but a standard computer or server with sufficient processing power and memory should generally be adequate.

System Configurations: The system may require specific configurations to ensure optimal performance and security. This may include setting up secure communication protocols (e.g., HTTPS) for data transmission, configuring firewalls and access control mechanisms to protect sensitive information, and implementing appropriate caching and load-balancing strategies for handling high-user traffic.

Database Management: Depending on the system requirements, a database management system (DBMS) may store and manage ticket-related data, such as user information, ticket details, and validation logs. The choice of DBMS will depend on factors such as scalability, data security, and the need for ACID (Atomicity, Consistency, Isolation, Durability) properties.

4. IMPLEMENTATION

With the help of RSA and Keccak, a secure ticket generation and validation system is implemented. Each step in the process is covered in more depth below:

The public and private key pair is generated using RSA algorithm. The common Python library is used to generate these keys. Each ticket is given a distinct ticket ID by the ticket creation component. The digital signature for the ticket ID is produced by the private key. The signature guarantees the validity of the ticket information. The Keccak hash function is used by the ticket validation component to produce a distinct hash value for the ticket ID. The integrity of the ticket data is guaranteed by the hash value.

A central database contains the ticket ID, digital signature, and hash value. The system gets the corresponding digital signature and hash value from the database whenever a ticket is given for validation. The system validates the digital signature using the appropriate public key to confirm the validity of the ticket data. By generating a hash value for the ticket ID using the Keccak hash function and comparing it to the previously stored hash value, the system checks the accuracy of the ticket data.

The results of the verification determine whether the system accepts or rejects the ticket. The system changes the database to indicate the ticket is utilized if it is accepted. Based on particular needs and use scenarios, these implementation processes might be modified and improved. To maintain the overall security and integrity of the system, it is also important to provide proper key management and secure storage of sensitive data.

Key Generation Process:

1. Key Size: The key generation process involves

determining the appropriate key sizes for RSA. The key sizes are typically measured in bits and are crucial to the system's security. The selection of key sizes should consider the desired level of security and the computational overhead associated with larger key sizes. Common key sizes for RSA range from 2048 to 4096 bits.

2. Key Pair Generation: To generate the RSA key pair, the system uses a secure random number generator to generate two large prime numbers, p, and q. These primes are kept secret and form the basis of the private key.

3. Public Key Calculation: The public key is multiplied by p and q to obtain the modulus, n. Additionally, the system selects an exponent, usually called e, which is relatively prime to (p-1) (q-1). The pair (n, e) forms the public key, which is made available to ticket generators for encryption.

4. Private Key Calculation: The private key is derived from the primes p and q and the exponent e. The system calculates the private exponent, d, which satisfies the equation (d * e) ≡ 1 (mod (p-1) (q-1)). The private key consists of the values (n, d) and is securely stored and kept confidential by the system.

Secure Key Storage and Management:

Proper key management is vital to ensure the security of the system. Here are some considerations for secure key storage and management:

1. Key Protection: The private key should be protected from unauthorized access or disclosure. It is typically stored in a secure location, such as a key management system, hardware security module (HSM), or a dedicated secure server.

2. Encryption: The private key can be encrypted using strong encryption algorithms, and access to the encrypted key is controlled through secure authentication mechanisms. This adds an extra layer of security to protect against unauthorized use.

3. Key Rotation: Regular key rotation is recommended to enhance security. This involves generating new key pairs and securely replacing the existing keys. Key rotation helps mitigate the risk of a compromised key being used maliciously.

4. Access Control: Access to the private key should be limited to authorized personnel only. Role-based access control mechanisms and proper authentication protocols should be implemented to ensure that only authorized individuals can access and manage the private key.

Database Selection:

The choice of database for storing hash values and signatures will depend on factors such as scalability, data security, and the specific requirements of the ticket generation and validation system. Common database options include:

Relational databases (e.g., MySQL, PostgreSQL).

NoSQL databases (e.g., MongoDB, Cassandra).

Cloud-based database services (e.g., Amazon RDS, Google Cloud Spanner).

Database Security Mechanisms:

To ensure the security of sensitive data, the database should implement appropriate security mechanisms. Some important security measures include:

1. Encryption: Sensitive data, such as hash values and signatures, should be encrypted at rest and in transit. Encryption helps protect the data from unauthorized access and ensures confidentiality.

2. Access Control: Access to the database should be restricted to authorized individuals or processes. Role-based access control (RBAC) mechanisms can be implemented to enforce granular access permissions and limit access based on user roles and responsibilities.

3. Auditing and Logging: The database should have logging and auditing mechanisms in place to track access and modifications to the data. This helps in detecting any unauthorized access attempts or data tampering.

4. Parameterized Queries: The database should support parameterized queries or prepared statements to prevent SQL injection attacks. This ensures that user input is properly sanitized and prevents malicious code injection.

Handling Increased Data Volume:

As the ticket generation and validation system processes a large volume of data, the database should be able to handle the increased workload. Considerations for handling increased data volume include:

1. Scalability: The database should support horizontal scaling, allowing for adding more database nodes or using sharding techniques to distribute data across multiple nodes. This ensures that the database can handle increased data volume and user load.

2. Indexing: Proper indexing of the database tables can enhance query performance and improve data retrieval efficiency, especially when dealing with large datasets.

3. Caching: Implementing caching mechanisms, such as in-memory or distributed caches, can help reduce the load on the database by serving frequently accessed data from memory.

4. Data Partitioning: Partitioning the database tables based on specific criteria, such as date or location, can improve query performance and enable efficient data retrieval.

Process for Generating Ticket IDs:

1. Unique Identifier Generation: The ticket generation system generates a unique identifier for each ticket. This identifier serves as the ticket ID and uniquely identifies each ticket within the system. The uniqueness of the identifier is crucial to ensure that each ticket can be uniquely referenced and validated.

2. Digital Signatures and Hash Values: Once the ticket ID is generated, it is linked to the corresponding digital signature and hash value. The digital signature is generated using the ticket generator's private key, ensuring the ticket's authenticity and integrity. The hash value is calculated using the Keccak algorithm on the ticket data, providing a unique fingerprint of the ticket's contents.

3. Linking Ticket ID, Digital Signatures, and Hash Values: The ticket ID, digital signature, and hash value are stored together in the system's database. They are linked to each other using appropriate data structures, such as relational tables or document structures, to maintain the association between the ticket ID and its corresponding digital signature and hash value.

Preventing Duplicate Ticket IDs:

To prevent duplicate ticket IDs and ensure the uniqueness of each ticket, the system employs the following measures:

1. Unique Identifier Generation Algorithm: The algorithm used to generate the ticket ID incorporates techniques to generate globally unique identifiers, such as UUIDs (Universally Unique Identifiers) or timestamp-based IDs. These algorithms generate highly random or time-based IDs, minimizing the likelihood of collisions and duplicate IDs.

2. ID Verification: When generating a new ticket ID, the system checks if the ID is already present in the database. If a duplicate ID is detected, the system regenerates a new ID until a unique one is obtained.

3. Database Constraints: The system can enforce constraints at the database level to ensure the uniqueness of the ticket IDs. This can be achieved through primary key constraints or unique index constraints on the ticket ID column, preventing

the insertion of duplicate IDs.

5. RESULTS AND ANALYSIS

The performance analysis of the ticket generation and validation system using RSA and Keccak can be done based on the time and complexity of the system. RSA algorithm's time complexity for generating a key pair is O(n^2), where n is the bit-length of the modulus. However, once the key pair is generated, the time complexity for encryption and decryption is O(n^3) for modular exponentiation. Keccak hash function's time complexity depends on the message length and the output size. The time complexity for hashing a message of length m to produce an output of length h is O(m+h).

As for security, RSA and Keccak are considered strong and secure cryptographic algorithms. However, the security of any cryptographic system also depends on adequately starting and managing keys and other sensitive data. As for the ticket generation and validation system using RSA and Keccak, the time and complexity analysis would depend on the specific implementation details and requirements. The time and complexity of the system would also depend on the ticket database's size and the ticket validations' frequency.

Here's an example Table 1 shows comparision of time complexity of RSA and Keccak for different input sizes.

Overall, the ticket generation and validation system using RSA and Keccak provides a strong and secure ticket authentication and verification solution with relatively low time and computational complexity. Here is a brief overview ofthe time required for the operation is discussed in Table 2. The table shows the time required for each operation using the proposed system.

Table 1. Comparison of RSA and keccak complexities

Input Size (bits)	RSA Time Complexity	Keccak Time Complexity
1024	O(2^20)	O(1024+224)
2048	O(2^41)	O(2048+224)
4096	O(2^83)	O(4096+224)

Table 2. Estimated time for proposed system

Operation	Time Required
Key Generation	32 milliseconds
Ticket Generation	6.4 milliseconds
Ticket Storage	4.2 milliseconds
Ticket Retrieval	3.9 milliseconds
Ticket Validation	9.7 milliseconds

A comprehensive security analysis is crucial for evaluating the effectiveness and vulnerabilities of cryptographic algorithms like RSA and Keccak.

RSA Security Analysis:

1. Resistance to Attacks: RSA is resistant to various types of attacks, including brute-force attacks, factoring attacks, and chosen plaintext attacks. The security of RSA relies on the computational difficulty of factoring large prime numbers.

2. Vulnerabilities: RSA's security can be compromised in scenarios where the key size is insufficient. Smaller key sizes are more susceptible to brute-force attacks or advances in factorization algorithms. Additionally, vulnerabilities may arise from poor key management practices, such as the compromise of private keys or weak random number generation.

3. Risk Mitigation: To enhance RSA's security, it is important to use sufficiently large key sizes, typically 2048 bits or higher. Key management should follow best practices, such as storing private keys securely, using secure random number generators, and periodically updating keys. Additionally, secure communication channels should be used to protect the exchange of public keys.

Keccak Security Analysis:

1. Resistance to Attacks: Keccak is designed to be resistant against various cryptographic attacks, including collision attacks, preimage attacks, and differential attacks. It is built on sponge construction, which provides strong security guarantees.

2. Vulnerabilities: Keccak is considered secure against currently known attacks when used with appropriate hash sizes. However, smaller hash sizes may increase the risk of collision or preimage attacks. Implementation flaws or side-channel attacks can also pose vulnerabilities.

3. Risk Mitigation: To ensure the security of Keccak, it is important to use an appropriate hash size. The choice of hash size depends on the specific security requirements and the expected lifetime of the application. Implementations should adhere to best practices and undergo rigorous testing to detect and address any vulnerabilities. Additionally, countermeasures should be implemented to protect against side-channel attacks, such as power analysis or timing attacks.

The overall security of the ticket generation and validation system using RSA and Keccak is influenced by various factors, including the key sizes used, the quality of key management practices, and the secure implementation of the algorithms.

The time estimates for the ticket system using RSA and Keccak can be derived based on several factors, including the complexity of the cryptographic operations, the efficiency of the algorithms' implementations, and the system's computational resources. The following metrics can be used to estimate the time required for various operations:

1. Key Generation: The time required to generate RSA key pairs depends on the system's key size and computational power. Larger key sizes generally require more time for generation. The estimated time can be derived from the average key generation time observed in similar systems or experiments.

2. Ticket Generation: The time required to generate a ticket involves performing cryptographic operations such as generating digital signatures and calculating hash values. These operations' complexity and the processed data's size influence the time required. Benchmarks and profiling of the implementation can provide insights into the average time taken for ticket generation.

3. Ticket Validation: The time required for ticket validation depends on the complexity of the validation process, including verifying digital signatures and performing hash comparisons. The size of the ticket data and the computational resources available impact the validation time. Similar systems or experiments can be used to estimate the average validation time.

4. Overall System Performance: The overall system performance can be evaluated by considering the cumulative time required for key generation, ticket generation, and validation. This can be measured using metrics such as the average time per transaction or the throughput (number of transactions processed per unit of time). Comparing these

metrics with existing systems can highlight any improvements the proposed system achieves.

In addition to the time-related metrics, evaluating the proposed ticket system using RSA and Keccak can involve other important metrics that comprehensively assess the system's performance, usability, scalability, cost, and adoption challenges. Here are some additional metrics to consider:

1. Scalability: Scalability measures the ability of the system to handle increasing workloads, data volumes, and user concurrency. Evaluating how the system's performance scales as the number of users and transactions increases is important. This can be measured by conducting load testing and stress testing to determine the system's capacity and identify any performance bottlenecks.

2. Usability: Usability refers to the system's ease of use and user-friendliness. Conducting user studies and obtaining user feedback can help assess the system's usability. Metrics such as task completion time, user satisfaction ratings, and user error rates can be used to evaluate the system's usability.

3. Security: Security is critical to any ticket generation and validation system. Evaluating the system's security can involve metrics such as resistance to attacks (e.g., digital signature forgery, hash collision), adherence to cryptographic best practices, and compliance with relevant security standards. Vulnerability assessments, penetration testing, and code reviews can help identify and address security vulnerabilities.

4. Cost-effectiveness: Assessing the system's cost-effectiveness involves considering the resources required for implementation, maintenance, and operation. This includes factors such as hardware costs, software licensing fees, personnel costs, and ongoing support and maintenance expenses. Comparing the costs of the proposed system with alternative solutions can provide insights into its cost-effectiveness.

5. Adoption Challenges: Identifying and addressing potential adoption challenges is crucial for successfully deploying the system. This includes assessing factors such as compatibility with existing infrastructure, integration with other systems, user training and acceptance, and any regulatory or legal considerations. Understanding these challenges and developing strategies to mitigate them can enhance the system's chances of successful adoption.

A combination of quantitative and qualitative measures can be used to evaluate the system using these metrics. Quantitative measures involve collecting numerical data and conducting statistical analysis, while qualitative measures involve gathering feedback, conducting interviews, and capturing user experiences. The evaluation can include a mix of laboratory testing, field trials, and user surveys to gather comprehensive data from multiple perspectives.

By considering these additional metrics, evaluating the proposed ticket system can provide a holistic assessment of its performance, usability, scalability, cost-effectiveness, and adoption challenges. This comprehensive evaluation helps to identify strengths and weaknesses, informs system improvements, and ensures that the proposed system meets the requirements and expectations of its users and stakeholders.

The proposed ticket generation and validation system using RSA and Keccak brings several significant contributions to the field. Through the research and implementation of the system, key findings and analyses have been made across various perspectives, including time complexity, security, and accuracy.

Regarding time complexity, the system has efficiently performed ticket generation and validation processes. The RSA and Keccak algorithms, with their well-established mathematical foundations, offer fast computation times, enabling real-time ticket processing even in high-demand scenarios. The timetable analysis has provided insights into the expected time requirements for each operation, aiding in system planning and optimization.

From a security perspective, the system leverages the robust security features of RSA and Keccak. The RSA algorithm ensures the confidentiality and integrity of ticket information through asymmetric key encryption, while Keccak provides a secure hash function for data integrity verification. The security analysis has highlighted the resistance of these algorithms to various attacks and the importance of key management and secure storage practices.

In terms of accuracy, the system ensures the authenticity and validity of tickets by using digital signatures and hash values. The integration of RSA and Keccak enables a strong and reliable verification process, mitigating the risk of ticket fraud and unauthorized access.

Moreover, the system can be expanded to support additional features and integrations, such as mobile ticketing, multi-factor authentication, or integration with existing ticketing infrastructure. These enhancements would cater to evolving user needs and industry requirements.

Lastly, exploring the application of emerging technologies like blockchain for ticket generation and validation could be a promising area of research. Blockchain's decentralized and immutable nature can provide enhanced security and transparency, addressing some limitations of centralized systems.

6. SECURITY ANALYSIS

6.1 Security features of the proposed system

The proposed ticket validation system using RSA and Keccak has several security features that make it robust and secure, and these features include:

Public-Private Key Pair: The system uses a public-private key pair generated by the RSA algorithm for encryption and decryption. The sender keeps the private key secret while the public key is shared with the receiver. This ensures that only the intended recipient can decrypt and validate the ticket.

Digital Signature: The ticket data is encrypted using the sender's private key to generate a digital signature. This ensures the authenticity and integrity of the ticket data, as any modifications to the data would result in a different hash value, and the decryption would fail.

Keccak Hash Function: The system uses the Keccak hash function to generate a fixed-length hash value for the ticket data. Keccak is a secure cryptographic hash function resistant to various attacks, including collision and preimage attacks.

Random Number Generation: The system generates a random number that adds randomness to the ticket data, making it difficult for an attacker to predict the hash value.

Data Confidentiality: The ticket data is encrypted using the sender's private key, ensuring that only the intended recipient can read the data. This protects the confidentiality of the ticket data and prevents unauthorized access.

Resistance to Forgery: The system is resistant to forgery, as any modifications to the ticket data would result in a different

hash value, and the decryption would fail. This prevents attackers from creating fake tickets and attempting to pass them off as valid.

Data Integrity: The Keccak hash function ensures that the ticket data cannot be altered without detection. Any change in the ticket data will result in a different hash value, which will not match the original hash value.

Non-Repudiation: The digital signature generated using RSA ensures that the sender cannot deny sending the ticket. The digital signature is unique to the ticket data and cannot be developed without the sender's private key.

Authentication: The sender's private key for digital signature generation and the sender's public key for digital signature verification ensures that the sender's identity is authenticated.

Confidentiality: The ticket data is not revealed during the ticket validation process, as only the hash value and digital signature are used for validation. The ticket data remains confidential and cannot be intercepted or stolen.

Resistance to Attacks: The Keccak hash function is designed to resist various attacks, including collision and preimage attacks. RSA is also known for its attack resistance, especially when using large key sizes.

Key Management: RSA ensures that the private key is kept secret and only known to the sender. This ensures that only the sender can generate valid digital signatures for the ticket data.

6.2 Potential security threats and attacks and Mitigation techniques for security threats

While the proposed ticket validation and generation system using RSA and Keccak offers several security features, it is still vulnerable to potential security threats and attacks. Some of the potential security threats and attacks that could compromise the security of the system are:

Brute Force Attack: Attackers can use a brute force attack to try all possible combinations of keys to decrypt the digital signature. This can be mitigated by using extended key sizes for RSA and Keccak, making it computationally infeasible to perform a brute-force attack.

Denial of Service (DoS) Attacks: Attackers can overload the ticket generation and validation system with many requests, making the system unavailable or unresponsive. This can be mitigated by implementing proper traffic management techniques, such as limiting the number of requests from a particular source.

Man-in-the-Middle Attack: Attackers can intercept and modify the ticket data during transmission, leading to a mismatch between the original and validated ticket data. This can be mitigated by using secure communication channels and encrypting the ticket data during transmission.

Key Management Attacks: Attackers can steal the sender's private key or compromise the key management system, leading to unauthorized access and manipulation of ticket data. This can be mitigated by using secure key management practices, such as storing keys safely and restricting access to authorized personnel only.

Cryptographic Attacks: Attackers can exploit weaknesses in the RSA or Keccak algorithms to break the encryption or hash functions. This can be mitigated by using strong and well-established cryptographic algorithms and periodically updating the system to address any vulnerability.

Insider Attacks: Authorized personnel with access to the ticket generation and validation system can intentionally or unintentionally manipulate or compromise the system, leading to unauthorized access and manipulation of the ticket data. This can be mitigated by implementing proper access controls and monitoring activities to detect and prevent suspicious behavior.

Mitigating security threats in the ticket generation and validation system is crucial to ensure the integrity and authenticity of the tickets. The proposed system employs several mitigation techniques to address potential threats. Here are the details of these techniques and their effectiveness:

1. Secure Key Management: The system implements secure key management practices to protect the RSA private key used for ticket generation and validation. This includes storing the private key in a secure hardware module, such as a Hardware Security Module (HSM), to prevent unauthorized access. Additionally, regular key rotation and strong access controls are implemented to mitigate the risk of key compromise. These practices are widely recognized in the industry and have been proven effective in protecting cryptographic keys.

2. Input Validation and Sanitization: The system incorporates robust input validation and sanitization mechanisms to prevent common attacks like SQL injection and cross-site scripting (XSS). This involves thoroughly validating and sanitizing user inputs before processing them. By implementing strict input validation rules and utilizing secure coding practices, the system can mitigate the risk of injection attacks and unauthorized access to sensitive data.

3. Role-Based Access Control (RBAC): RBAC is employed to enforce access control policies and restrict privileges based on user roles. Each user is assigned specific roles, and access permissions are granted accordingly. This helps prevent unauthorized users from accessing sensitive functionalities or data within the system. RBAC has been widely adopted in various systems and has proven effective in controlling access and reducing the attack surface.

4. Logging and Monitoring: The system incorporates comprehensive logging and monitoring mechanisms to detect and respond to suspicious activities. It logs relevant system events, user activities, and ticket generation/validation activities. Real-time monitoring and analysis of these logs can help promptly identify and mitigate potential security incidents. Implementing a Security Information and Event Management (SIEM) system or similar tools can enhance threat detection and incident response capabilities.

5. Regular Security Updates and Patches: To mitigate vulnerabilities arising from software and system vulnerabilities, the system regularly applies security updates and patches. This includes keeping the operating system, web server, database, and other software components current. By promptly applying security patches and updates, the system can address known vulnerabilities and reduce the risk of exploitation.

While these mitigation techniques are effective, they have some limitations and associated costs. For example, implementing secure key management practices may require additional hardware and software investments and ongoing operational costs for key rotation and management. RBAC implementation requires careful design and maintenance of user roles and access policies, which can be complex in large-scale systems. Additionally, the logging and monitoring mechanisms can generate significant logs, requiring storage and analysis infrastructure.

Examples from industry best practices and standards can be referenced to support the effectiveness of these mitigation

techniques. For instance, the National Institute of Standards and Technology (NIST) provides guidelines and recommendations for secure key management, input validation, RBAC, and logging and monitoring. Various security frameworks, such as the Payment Card Industry Data Security Standard (PCI DSS), also emphasize the importance of these techniques in protecting sensitive data and preventing security breaches.

Moreover, case studies and real-world examples of successful implementations of these techniques in similar systems can be referenced to demonstrate their effectiveness. These examples highlight how organizations have successfully mitigated threats and protected their ticketing systems using similar mitigation strategies.

By providing detailed mechanisms, discussing limitations and costs, and referencing relevant standards and examples, the proposed mitigation techniques can be better supported and validated, ensuring the security and integrity of the ticket generation and validation system.

Comparing the security analysis and mitigation techniques with previous research on similar systems is essential to demonstrate the novelty and improvements of the proposed work. By highlighting the advancements and unique contributions, the validity and significance of the analysis can be reinforced. Here's a comparison of the security analysis and mitigation techniques with previous research:

1. Security Analysis: The proposed system's security analysis considers the specific threats and vulnerabilities associated with ticket generation and validation systems. It comprehensively examines potential attacks such as forgery, tampering, and unauthorized access. The analysis includes an in-depth discussion of the security features of RSA and Keccak algorithms, their resistance to various attacks, and the importance of key management.

In comparison to previous research, the proposed work builds upon existing knowledge by specifically focusing on the security aspects of ticket generation and validation systems. While there may be prior studies on the security of cryptographic algorithms like RSA and hash functions like Keccak, this research applies them in the context of ticketing systems, considering their unique requirements and challenges.

2. Mitigation Techniques: The proposed system incorporates several mitigation techniques to address the identified security threats. These techniques include secure key management, input validation, RBAC, logging and monitoring, and regular security updates and patches. Each technique is discussed in detail, explaining its relevance and effectiveness in mitigating specific threats.

Compared to previous research, the proposed work presents a comprehensive set of mitigation techniques tailored for ticket generation and validation systems. While some of these techniques may be common in general system security practices, their application and adaptation to the ticketing domain provide novel insights and considerations. The research specifically examines the suitability of these techniques in the context of ticketing systems and highlights their benefits and limitations.

3. Novel Techniques and Insights: The proposed work introduces novel techniques and insights in the security analysis and mitigation strategies for ticket generation and validation systems. For example, the use of RSA for digital signature generation and verification in the ticketing context may have yet to be extensively explored in previous research. Integrating Keccak for generating and verifying ticket hash values is also a unique contribution.

Furthermore, the detailed analysis of the limitations and potential risks associated with the proposed mitigation techniques adds valuable insights. By discussing each technique's costs, implementation challenges, and trade-offs, the research provides a practical perspective that can guide system developers and administrators in making informed decisions.

Compared to previous research, the proposed work adds to the body of knowledge by combining cryptographic techniques (RSA and Keccak) specifically tailored for ticket generation and validation. The research offers a thorough analysis of their security properties and practical mitigation strategies to ensure the system's integrity, authenticity, and resistance to various attacks.

By highlighting the novel techniques, insights, and improvements over previous research, the proposed work demonstrates its contribution to the field of ticket generation and validation system security. The comparative analysis showcases the significance and validity of the security analysis and mitigation techniques, establishing the research as a valuable addition to the existing knowledge in the domain.

3. Conclude by restating your proposed system's key security features and mitigation techniques. Summarize how they achieve confidentiality, integrity, availability, and other security objectives. Briefly reiterate limitations and areas of improvement to help guide future research efforts.

In conclusion, the proposed ticket generation and validation system incorporates several key security features and mitigation techniques to ensure the system's confidentiality, integrity, and availability. These security features and mitigation techniques include:

1. Encryption using RSA: The system utilizes RSA encryption to ensure the confidentiality of sensitive data during transmission. By encrypting the ticket information and communication channels, the system protects against unauthorized access and eavesdropping.

2. Digital Signatures using RSA: Digital signatures generated using RSA provide authentication and integrity to the ticket data. The signatures verify the ticket's authenticity and detect any tampering or modifications.

3. Hash Function (Keccak): The system employs the Keccak hash function to generate hash values for the ticket data. These hash values are unique identifiers for each ticket and ensure data integrity. Any tampering with the ticket data will result in a different hash value, alerting the system to potential anomalies.

4. Secure Key Management: The system incorporates secure key management practices to protect the confidentiality and integrity of cryptographic keys. It ensures that the private keys used for digital signature generation are securely stored and accessible only to authorized personnel.

5. Input Validation: The system implements rigorous input validation techniques to prevent malicious input and potential vulnerabilities. It checks the integrity and format of the received ticket data, preventing the injection of unauthorized or malicious content.

6. Role-Based Access Control (RBAC): RBAC mechanisms are employed to enforce access control policies and limit access to sensitive functionalities and data. Only authorized personnel with the necessary privileges can perform critical operations such as ticket validation and system configuration.

While the proposed system incorporates these security features and mitigation techniques, it is important to acknowledge the limitations and areas for improvement. These include:

1. Scalability: The system's scalability may be a concern as the number of ticket transactions increases. Further research is needed to optimize the system's performance and ensure efficient handling of many ticket requests.

2. Key Management: While the proposed system addresses secure key management, ongoing monitoring, and updates are essential to mitigate the risk of key compromise. Future research could explore more advanced key management techniques, such as hardware security modules (HSMs) or secure multi-party computation.

3. Attack Resilience: The system's resistance to advanced attacks, such as side-channel attacks or cryptographic vulnerabilities, should be further investigated. Robust countermeasures should be developed and integrated to enhance the system's resilience against emerging threats.

4. Usability: The user interface and overall user experience of the system should be continuously improved to ensure ease of use and minimize user errors. Usability testing and feedback from end-users can provide valuable insights for enhancing the system's usability.

In conclusion, the proposed ticket generation and validation system incorporates robust security features and mitigation techniques to achieve the confidentiality, integrity, availability, and other security objectives. However, further research and development efforts are necessary to address the identified limitations and improve the system's scalability, key management practices, attack resilience, and usability. By addressing these areas of improvement, the proposed system can become even more secure and reliable, meeting the evolving needs of ticketing systems in a rapidly changing threat landscape.

7. CONCLUSIONS AND FUTURE WORK

The proposed ticket generation and validation system using RSA and Keccak has several practical implications and applications in various real-world scenarios. The system can be implemented in different domains where secure ticketing and validation are crucial such as Transportation Industry, Event Management, Access Control Systems, E-commerce and Digital Transactions. This improves the trustworthiness of digital transactions and reduces the risk of fraudulent activities. The potential benefits of the proposed system include Enhanced Security, Improved Efficiency, Cost Savings, and Improved User Experience. In terms of commercial viability and potential for adoption at scale, the proposed system offers significant advantages. Its integration with existing ticketing systems can be achieved with proper system customization and integration efforts. The use of well-established cryptographic techniques like RSA and Keccak enhances its credibility and ease of adoption. Furthermore, the increasing reliance on digital ticketing systems and the need for robust security measures make the proposed system highly relevant and valuable in the current digital landscape. The system is efficient and can be easily integrated into existing ticket generation and validation systems, requiring minimal changes or modifications. In addition with that, our system is better than the one in the study [14] in therms of ticket validation.

REFERENCES

[1] Muthukumar, B., Mayan, J.A., Nambiar, G., Nair, D. (2019). Qr code and biometric based authentication system for trains. In IOP Conference Series: Materials Science and Engineering, IOP Publishing, 590(1): 012010. https://doi.org/10.1088/1757-899X/590/1/012010

[2] Patil, S., Desurkar, S., Sanas, D. (2016). An intelligent ticket checker application for train using QR code. International Journal of Computer Applications, 15-20.

[3] Ji, P. (2023). The advance of cryptocurrency wallet with digital signature. Highlights in Science, Engineering and Technology, 39: 1098-1103. https://doi.org/10.54097/hset.v39i.6714

[4] Abid, R., Iwendi, C., Javed, A.R., Rizwan, M., Jalil, Z., Anajemba, J.H., Biamba, C. (2021). An optimised homomorphic CRT-RSA algorithm for secure and efficient communication. Personal and Ubiquitous Computing, 27: 1405-1418. https://doi.org/10.1007/s00779-021-01607-3

[5] Sideris, A., Sanida, T., Dasygenis, M. (2020). High throughput implementation of the keccak hash function using the nios-ii processor. Technologies, 8(1): 15. https://doi.org/10.3390/technologies8010015

[6] Hussain, M.A., Sree Varsha, K., Krishnamraju, K., Lavanya, K., Chakradhar, B. (2022). QR-based ticket verification and parking system. In Cybernetics, Cognition and Machine Learning Applications: Proceedings of ICCCMLA, Singapore: Springer Nature Singapore, 2021: 123-130. https://doi.org/10.1007/978-981-19-1484-3_14

[7] Li, X.L., Niu, J., Gao, J.T., Han, Y. (2019). Secure electronic ticketing system based on consortium blockchain. KSII Transactions on Internet and Information Systems (TIIS), 13(10): 5219-5243. https://doi.org/10.3837/tiis.2019.10.022

[8] Pangan, A.M.S., Lacuesta, I.L., Mabborang, R.C., Ferrer, F.P. (2022). Authenticating data transfer using RSA-generated QR codes. European Journal of Information Technologies and Computer Science, 2(4): 18-30. https://doi.org/10.24018/compute.2022.2.4.73

[9] Gangurde, N., Ghosh, S., Giri, A., Gharat, S. (2022). Ticketing system using AES encryption based QR code. In 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, 201-206. https://doi.org/10.1109/ICSSIT53264.2022.9716234

[10] Goyal, J., Ahmed, M., Gopalani, D. (2022). A privacy preserving e-voting system with two-phase verification based on ethereum blockchain. https://doi.org/10.21203/rs.3.rs-1729918/v1

[11] Ayasy, M.I.S., Barmawi, A.M. (2022). Protecting author royalty of digital assets using blockchain and elliptic curve cryptography. In 2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), IEEE, 86-92. https://doi.org/10.1109/GECOST55694.2022.10010412

[12] Jiang, P., Qiu, B., Zhu, L.H. (2022). Report when malicious: deniable and accountable searchable message-moderation system. IEEE Transactions on Information Forensics and Security, 17: 1597-1609. https://doi.org/10.1109/TIFS.2022.3167900

[13] Nair, M.A., Taunk, S., Reddy, P.G., Sultana, H.P. (2019). Smart metro rail ticketing system. Procedia Computer

Science, 165: 435-441. https://doi.org/10.1016/j.procs.2020.01.003
[14] Gayathri, C., Loganathan, P., Gokul Kannan, G., GokulRaj, V., Dhineshkumar, S. (2022). Automated railway reserved ticket validation system. International Journal of Research in Engineering and Science (IJRES), 10(5): 60-65.
[15] Abbas, I., Iqbal, H., Ahmad, M., Naveed, A., Qudoos, B. (2020). Comparative study of technologies and intelligent train ticketing system. Indian Journal of Science and Technology, 13(15): 1570-1579. https://doi.org/10.17485/IJST/v13i15.13
[16] Manori, A., Devnath, N., Pasi, N., Kumar, V. (2017). QR code based smart attendance system. International Journal of Smart Business and Technology. http://dx.doi.org/10.21742/ijsbt.2017.5.1.01
[17] Jamnik, A., Shahare, M., Kamble, S., Kale, N., Bhadade, M., Sonekar, S.V. (2019). Digital ticket booking and checking using aadhaar card or fingerprint and android application. In 2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE), IEEE, 503-507. https://doi.org/10.1109/RDCAPE47089.2019.8979
[18] Dalal, R., Khari, M., Arbab, M.N., Maheshwari, H., Barnwal, A. (2021). Smart metro ticket management by using biometric. In Multimodal Biometric Systems, CRC Press, 101-110. https://doi.org/10.1201/9781003138068-8
[19] Tsvetkov, V.Y., Shaytura, S.V., Ordov, K.V. (2019). Digital management railway. In International Scientific and Practical Conference on Digital Economy (ISCDE), Atlantis Press, 2019: 846-850. https://doi.org/10.2991/iscde-19.2019.34
[20] Guan, Y.X., Wu, B., Jia, J.M. (2020). Does online ticket booking system make people better off? An empirical study on railway service. Transportation Research Part F: Traffic Psychology and Behaviour, 73: 143-154. https://doi.org/10.1016/j.trf.2020.03.014
[21] Sørensen, A.Ø., Olsson, N.O., Akhtar, M.M., Bull-Berg, H. (2019). Approaches, technologies and importance of analysis of the number of train travellers. Urban, Planning and Transport Research, 7(1): 1-18. https://doi.org/10.1080/21650020.2019.1566022
[22] Shukla, P.K., Aljaedi, A., Pareek, P.K., Alharbi, A.R., Jamal, S.S. (2022). AES based white box cryptography in digital signature verification. Sensors, 22(23): 9444. https://doi.org/10.3390/s22239444

ABBREVIATION

ADT	Authentication Data Table
DoS	Denial of service
GECOST	Green Energy, Computing and Sustainable Technology
ICSSIT	International Conference on Smart Systems and Inventive Technology
IoT	Internet of Things
IPFS	InterPlanetary File System
NFC	Near-field communication
RDCAPER	Recent Developments in Control, Automation & Power Engineering
RFID	Radio-frequency identification
TTE	Train Ticket Examiner
SHA	Secure Hash Algorithms