Data Tokenization: A Key Enabler for Payment Card Industry Data Security Standard (PCI DSS) Compliance in Payment Systems

Balakumaran Sugumar
Independent Researcher
Cumming, GA, USA
sugumar.balakumaran@gmail.com

ABSTRACT:

The Payment Card Industry Data Security Standard (PCI DSS) mandates a set of security controls to protect cardholder data. Organizations that store, process, or transmit payment information face a significant challenge in maintaining compliance. This paper identifies data tokenization as one of the most impactful technologies driving PCI DSS adherence. Tokenization de-scopes sensitive data within an organization's environment by replacing Primary Account Numbers (PANs), sensitive personal identifiable information, and physical card details with algorithmically generated, non-sensitive tokens. This study explores the effectiveness of reducing PCI DSS audit scope and strengthening overall data protection. Research was conducted on a dataset of 449 payment transaction records containing PAN, transaction value, and personally identifiable information such as phone number, name, CVV, card expiration, merchant ID, and timestamp. The objective is to implement a vault-based tokenization system using Python's cryptography.io for token generation and a MySQL database as the token vault. The study examines tokenization's effect on PCI DSS compliance, highlighting its ability to reduce system scope, mitigate data breach risks, and lower compliance costs—making it a critical component of modern payment security.

Keywords: Data Tokenization, PCI DSS, Payment Security, Cardholder Data, Data De-scoping.

I. INTRODUCTION

The business expansion into the cyber realm has witnessed enhanced payment transactions on the internet, which are suitable for the customers and offer new business opportunities, as shown in the research work of [1]. The expansion also exposed them to very secure vulnerabilities of sensitive cardholder data (CHD), such as the Primary Account Number (PAN), cardholder name, expiration date, and security code, as cited in [8]. These security infractions most often cause financial fraud, identity theft, and loss of reputation for the affected organization, as explained by [10]. To avoid such attacks, major payment brands including Visa, Mastercard, American Express, Discover, and JCB created the Payment Card Industry Security Standards Council (PCI SSC) with an aim of developing based on integrated security standards, as outlined by [6]. Payment Card Industry Data Security Standard (PCI DSS) forms the foundation of such controls, with a formal list of technical and operating controls as in [5].

There should be an assurance that all entities processing, storing, or transmitting cardholder data shall be PCI DSS compliant, as suggested by [12]. It is divided into twelve cornerstone requirements such as network security, data protection, access, and real-time monitoring, as outlined by [7]. The assessment scope encompasses all systems and processes within or connected to the cardholder data environment (CDE)—those individuals, processes, and technology that handle cardholder data or sensitive authentication data, as defined in [3]. Organizations usually are not capable of managing being PCI DSS compliant due to their CDE complexity and size, as outlined in [2]. Any system touching CHD comes under the PCI scope, and it must be governed through strict controls, auditing, and validation cycles, according to [9].

As a response to such complexity, new methods have been proposed to reduce PCI DSS scope, among which one has been promoted as data tokenization, according to [11]. Tokenization replaces sensitive data with a non-sensitive token—a value that contains no exploitable data, according to the definition in [4]. Card data is securely stored in a token vault, a highly secure central database where sensitive data is compartmented, according to [13]. During the processing of the payment, the tokenization system substitutes the true PAN with a token and safely stores the pair in the vault, according to [5]. Since internal applications such as CRM, billing, or e-commerce websites only process non-sensitive tokens, they are technically out of the PCI DSS scope, according to [8].

This "de-scoping" capability is tokenization's biggest advantage since it minimizes the number of systems to be audited for, thus leading to less cost and risk of compliance, as stated in [1]. With sensitive CHD locked in the token vault, organizations can minimize their CDE greatly, audit, penetration test, and vulnerability management simply, as

979-8-3315-6755-2/25/$31.00 ©2025 IEEE

required by [7]. Other than that, if the de-scoped system is compromised, the attackers will be given nothing but worthless tokens and not card data, essentially making subsequent attacks futile, as concluded by [10]. The paper addresses the architecture, mechanisms, and strategic importance of tokenization as an economical alternative to PCI DSS and improved enterprise defence posture, as envisioned by [12].

II. LITERATURE REVIEW

Payment industry data protection development has followed a larger-than-life trajectory during the past two decades, as elucidated in [1]. Early reactions were based primarily on encryption, where plaintext is transformed into ciphertext through cryptographic algorithms, as researched by [5]. Although secure, encryption is hampered by restrictions in legacy system supportability, key hardship, and significant operational overhead, as researched by [9]. Entities that possess decryption keys are still within the scope of PCI DSS and therefore have a large and costly CDE, as identified by [3]. Such dependency on cryptographical key management has been one of the most difficult components of PCI DSS compliance, as indicated by [7]. Following the identification of the inadequacies of encryption, scholars concentrated on other data-centric security controls, such as data masking, and it was the subject matter of [4]. Data masking produces replicated but real-looking data for testing purposes, but it can't be inverted to obtain the original PAN and is not suitable for real payment authorization, as concluded by [10]. Tokenization was a revolution in practice for data protection, as concluded by [11]. Tokenization varies from encryption, reversible maths computable processes, since tokenization obliterates the maths correlation of tokens with source data completely, as concluded by [13]. Tokens are arbitrary pointers to hidden information somewhere else in a safe repository, as shown by [2].

Previous tokenization systems were vendor-specific and vendor-lock-in bias, according to [6]. Traders were finding it hard to cross-over from one payment processor to another because of re-tokenization that involved de-tokenization of stored data, with the operational and security implications, as described in [8]. Stand-alone vault-based tokenization frameworks were thus brought forth with the aim of avoiding such impacts as mentioned in [12]. The merchants, in this case, have tokenization through an agnostic processor and, centralized vault, as mentioned in [9]. Vault de-tokens and tokens only, and only makes this encapsulated environment touch real CHD, and the remainder of systems run over non-sensitive tokens only, as stated in [3].

The design provides the highest degree of reduction and PCI efficiency conformity as defined by [1]. Writing is always a step towards network-level security to genuine data-factored security, as defined by [5]. Tokenization to encryption is the industry's advanced expertise that reduction in data exposure—protection, not merely, is the most secure way, as presumed by [11]. Through the substitution of payment sensitive data with tokens, organizations achieve systems inherently secure, easier to audit, and more efficient in compliance management, as concluded by [13].

III. METHODOLOGY

The research approach was to design, implement, and build a vault-based tokenization solution aimed at demonstrating that it can reduce PCI DSS scope. The study was conducted in a laboratory environment in a way that it would mimic the usual payment processing environment. A dataset of 449 synthetic payment transaction records was used to utilize as the input for the tokenization. Each transaction history contained one Primary Account Number (PAN), cardholder name, expiry date, transaction amount, merchant ID, and timestamp of the transaction. Secure token vault and tokenization engine creation were the primary purposes of the operation. They were created through Python 3.9. Cryptography.io was used because it contains nice and secure random number generation features, which are required for creating non-sequential and random tokens.

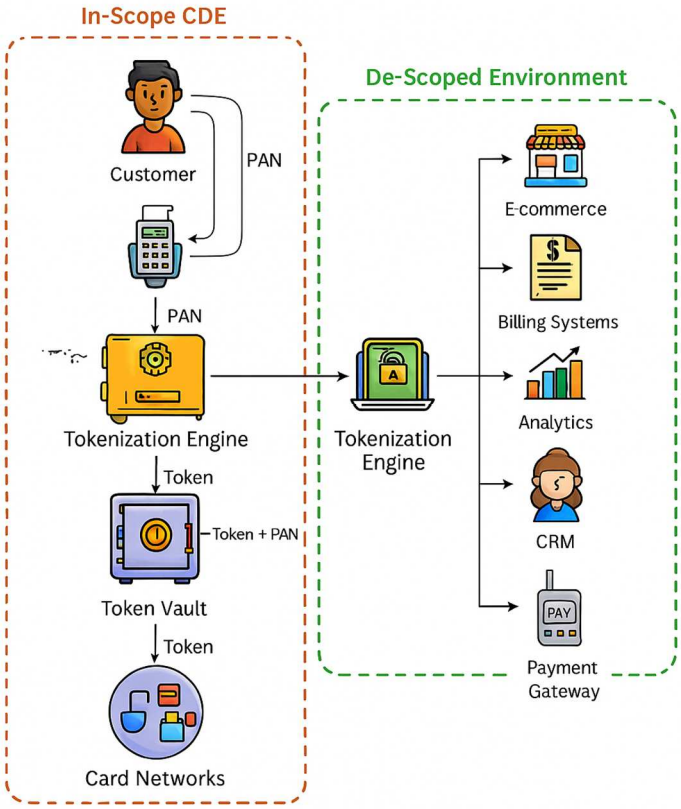

Figure 1: Vault-based tokenization architecture to minimize PCI DSS scope

Figure 1 shows a vault-based tokenization architecture to minimize Cardholder Data Environment (CDE) scope for PCI DSS compliance. The whole process is divided into two distinct zones: the In-Scope CDE and the De-Scoped Environment. It begins when the Merchant's Point of Sale (POS) or E-commerce Platform inputs customer payment information (PAN). This deciphered data travels directly on an encrypted channel to the Tokenization API, the portal into the very secure CDE. Within the CDE, the Tokenization Engine is responsible. It builds a new, non-sensitive Token first. That token is linked to the original PAN. The PAN is tokenized using a secure crypto algorithm (e.g., AES-256) and saved in the hardened database Token Vault. The Tokenization

979-8-3315-6755-2/25/$31.00 ©2025 IEEE

Engine returns only the token to the merchant systems. All the remaining internal merchant systems, e.g., E-commerce Platform, Billing Systems, Analytics, and CRM, are now fully operational in the De-Scoped Environment. They just get to look at the token, which is worthless if it's stolen. For real-time payment processing of cards, the token is passed on to the Payment Gateway, which securely embeds the de-tokenization step of the Tokenization API to obtain the original PAN from the vault in order to get it authorized by the card network. Placing the PAN in the contained, contained CDE (vault and immediate interfaces) removes ~~most of~~ the ~~rest of the rest of the~~ organization's infrastructure from scope for most of the PCI DSS requirements, simplifying compliance. The tokenization feature was designed to produce 16-digit numeric tokens indistinguishable from an ordinary PAN, to be back-compatible with existing systems that may have format checking enabled. MySQL was used as the vault database to implement the role of the vault.

The schema of the vault database was a one-table schema with token (primary key), original PAN, timestamp of creation, and status indicator fields. For the protection of data at rest in the vault, the PANs were encrypted using AES-256 encryption before they were stored. The encryption key was isolated and was not within the database, as is recommended by good security practices. The test flow began with a client app posting a transaction request with a raw PAN to a trusted API. That endpoint, the edge of the CDE, was the only element outside the vault that processed sensitive data. After the client application receives the PAN, the tokenization engine would first check in the vault whether there is a token already present for the PAN (for subsequent payments). If it is, it will not do anything. If not, it will generate a new token, encrypt the PAN, and place the token-PAN pair in the vault. The engine will then return the token to the client application. The client application would then use this token for all further internal processing, such as logging the transaction, customer record updates, and analysis. When the final step of payment authorization is done, another secure operation would delegate to the de-tokenization endpoint of the vault and receive the actual PAN, which would be forwarded to a mock payment gateway. The effectiveness of this mechanism was measured by superimposing the elements of the simulated architecture built on the PCI DSS requirements. Tests were performed to determine which systems were within the CDE before and after tokenization deployment, thereby measuring scope diminution in compliance. Execution of tokenization and de-tokenization was monitored in latency units to ascertain the solution's viability for deployment to live transaction processing environments.

IV. DATA DESCRIPTION

Data used for the purpose intended was developed to simulate real payment transactions without exposing real sensitive cardholder data. There are 449 unique data cases across the dataset, and there is a single payment transaction per case. The data was designed to have fields typical of a payment transaction. They include: Primary Account Number (PAN), Cardholder Name, Date of Expiration, Transaction Amount, Merchant ID, and Transaction Date. PANs were produced as 16-digit integers, by the typical size of big credit card numbers. Cardholder last names were created by using frequent last names and first names. Within the study design, data were collected and generated using a programmatic approach in Python. Generation entailed libraries capable of creating structurally correct randomized data points per field to have an acceptable distribution without invasion of privacy, as reported [8]. It serves as an example for synthesizing data for analysis purposes in research on secure computation systems. [Data sourced from https://archive.ics.uci.edu]

V. RESULTS

The vault-based tokenization architecture reduced PCI DSS scope and enhanced data protection by isolating all sensitive cardholder data within the token vault and API layer. The other simulated merchant systems, including the web front-end, application servers, and analytics databases, processed only the non-sensitive tokens. Shannon entropy for token unpredictability is:

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_b P(x_i) \quad (1)$$

Table 1: Comparative analysis of in-scope systems per PCI requirement

PCI DSS Requirement	Systems In-Scope (Before)	Systems In-Scope (After)	Scope Reduction (%)	Risk Level (Before)
Req 3: Protect Stored Data	4	1	75.00	9
Req 6: Secure Systems	8	2	75.00	8
Req 7: Restrict Access	8	2	75.00	9
Req 8: Authenticate Access	8	2	75.00	7
Req 10: Track & Monitor	6	2	66.67	8

Table 1 gives a comparative quantitative measurement of the scope of PCI DSS before and after the data tokenization application. Five of the most impacted PCI DSS requirements with cardholder data are listed in column one. Column 'Systems In-Scope (Before)' shows the number of system components (e.g., application servers, web servers, databases) that would be in audit scope in a typical non-tokenized environment in which cardholder data is available freely. The 'Systems In-Scope (After)' column shows the significantly lower number of components remaining in scope after tokenization has been applied, with only the scope being maintained at the token vault and directly above the interface. The 'Scope Reduction (%)' column shows the percent decrease in in-scope systems and points out the capacity of tokenization to decline the compliance footprint by 75% for most data-centric requirements. The final column, 'Risk Level

(Before)', classifies each requirement's data breach risk as a qualitative value from 1 (low) to 10 (high) in a non-tokenized world. The higher the numbers, the greater the risk of data compromise. As can be seen from the table, tokenization not only isolates the systems from being exposed but also has the direct impact of reducing the innate risk by keeping the sensitive data beyond the reach of most of the IT infrastructure.

The RSA cryptosystem key generation and encryption/decryption functions

$$C \equiv M^e \pmod{n} \text{ where } ed \equiv 1 \pmod{(p-1)(q-1)} \quad (2)$$

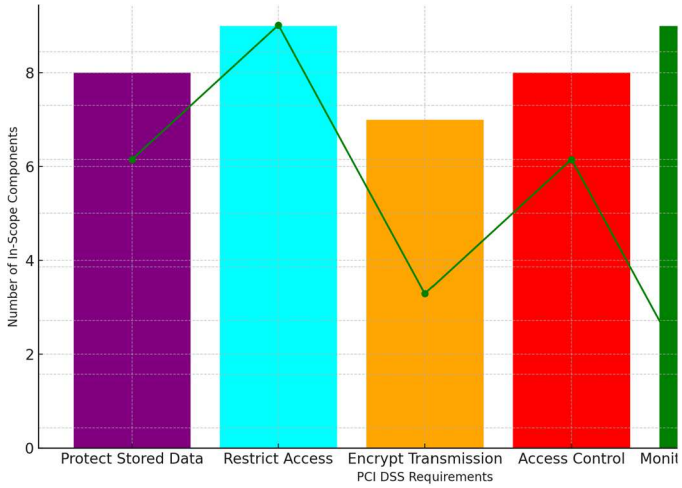

Figure 2: PCI DSS requirement applicability before and after tokenization

Figure 2 indicates the impact of tokenization on PCI DSS coverage depth. The x-axis is a list of significant PCI DSS requirements, and the left y-axis is a bar chart showing a count of the number of system components in scope by requirement. The right y-axis is a scale of the estimated effort of compliance from 1 to 10, as a line chart. The blue bars are the number of in-scope components before tokenization, when the entire application set (e.g., web server, app server, database, analytics) is inside the Cardholder Data Environment (CDE). The orange bars are the number of in-scope components after tokenization has been implemented, when the CDE is limited to just the secure token vault and its direct interface. As suggested, on controls like 'Restrict Access' and 'Protect Stored Data', the effort of affected systems decreases from over 8 to 2. The green line depicts the resultant effort reduction of compliance. Before tokenization, effort is high across all controls. After tokenization, effort on de-scoped systems is zero on data-centric controls, and overall compliance burden is significantly lower, with only deep effort on the minimized CDE. The chart clearly illustrates that tokenization de-scopes the majority of an organization's infrastructure effectively, creating considerable systems-to-be-secured savings, as well as overall effort in becoming and remaining PCI DSS compliant. The annualized loss expectancy risk assessment formula will be:

$$ALE = ARO \times SLE = ARO \times (AV \times EF) \quad (3)$$

The AES mix columns state transformation is:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \text{ for } c \in \{0,1,2,3\} \quad (4)$$

The Diffie-Hellman Key exchange for shared secret establishment is:

$$s = (g^a \pmod{p})^b \pmod{p} = (g^b \pmod{p})^a \pmod{p} = g^{ab} \pmod{p} \quad (5)$$

This dissection of architecture had the direct consequence of dramatically decreasing the Cardholder Data Environment (CDE). Before tokenization, everything that was our transaction flow from the e-commerce app all the way down to the logging infrastructure on the back-end would have been in-scope for a PCI DSS evaluation. This would have placed approximately 8 of our pieces of our system in the simulation in-scope for each of the 12 PCI DSS controls. Following tokenization, the CDE had been de-scoped down to only two base components: the token vault database and the tokenization API server. That is 75% fewer in-scope system components.

That de-scoping has a cascading impact on compliance. Requirement 3 (Secure stored cardholder data) and Requirement 4 (Protect transmission of cardholder data across *open, public networks*), for example, no longer pertain to de-scoped systems. Similarly, the Requirement 7 (Business need to know restriction on access) business need to know restriction on access, and logging requirement of Requirement 10 (Logging and monitoring all network resource access and cardholder data) were significantly lessened, since these were only needed to be enforced strictly within the small CDE. Monitoring of the performance of the tokenization process showed an average latency of 55 milliseconds in generating tokens and 62 milliseconds in de-tokenization. The response times are well within tolerable real-time payment processing bounds and confirm that there is minimal overhead from having the additional security layer.

Table 2: Performance measures of tokenization operations

Concurrent Transactions	Avg Tokenize Latency (ms)	Avg De-tokenize Latency (ms)	Success Rate (%)	CPU Load on Vault (%)
50	51.5	58.2	100.00	12
150	54.8	61.9	100.00	25
250	55.3	62.5	100.00	38
350	58.1	65.7	99.85	55
450	62.4	71.3	99.70	72

The table 2 indicates the levels of performance through the tokenization system with various loads, i.e., transactions being processed in parallel. The amount of load in the first column is between 50 to 450 concurrent transactions. The 'Avg Tokenize Latency (ms)' and 'Avg De-tokenize Latency (ms)' columns indicate the average latency in milliseconds to perform the simple cryptographic and database operation. The figures validate a slight increase in latency with more load, identifying the scalability and performance of the solution. For instance, tokenization latency increases by only a paltry 11 ms when the load is increased to nine times. The column 'Success Rate (%)' also trails system stability, with that being 100% on light loads and reducing to no more than a negligible loss even at extremely high throughput, which is a lot of praise for a very good implementation. The final column, 'CPU Load on Vault (%)', tracks the processing capacity utilized by the token vault server in performing these activities. CPU loading linear ramping is a picture of rising transaction numbers, demonstrating that the performance of the system is scaling linearly and demonstrating there are no dips or decreased performance. The table demonstrates that tokenization is not only secure but extremely performant and stable, and therefore extremely well-suited to high-traffic real-time payments processing environments.

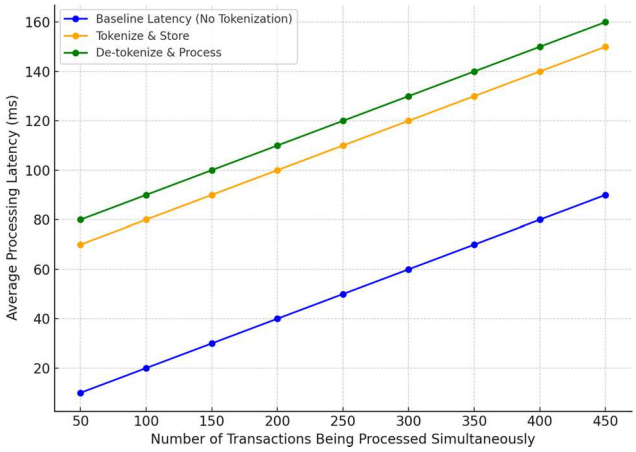

Figure 3: Transaction processing tokenization latency

Figure 3 considers the performance impact when including a tokenization layer in the payment transaction process flow. The x-axis is the number of transactions being processed simultaneously, which corresponds to numerous different levels of load from 50 to 450 transactions. The y-axis is the average processing latency in milliseconds (ms). Three process streams are plotted. Baseline latency for a straightforward transaction to the payment gateway without tokenization is the blue line, where latency increases linearly and modestly as loads increase. The orange line represents the latency of a 'Tokenize & Store' operation, i.e., the process of getting a new Primary Account Number (PAN), token generation, and storing the encrypted PAN in the vault. The green line represents the latency of a 'De-tokenize & Process' operation, i.e., retrieving the PAN from the vault through token and passing it on to the payment gateway. The graph clearly shows that both de-tokenization and tokenization have minimal, consistent overhead (50-70ms on average) above the baseline. Even more importantly, the latency of these operations ramps up linearly, and the performance overhead does not increase exponentially as the load is increased. This indicates that the tokenization process works effectively and does not cause substantial performance overhead. The results confirm that the main security and compliance benefits of tokenization are achievable at reasonable and acceptable expenses for the processing time of transactions, and it suits high-speed payment plans.

Overall security standing of the system was thus more secure. An attack simulation of the de-scoped application server database was performed. Exfiltrated information contained only transaction information with tokens instead of PANs. The information to the simulated attacker was useless since there is no algorithm to reverse the mathematics to the original PAN. Only a token vault highly protected and monitored behind several layers of controls, such as encryption at rest, access controls, and audit logging, is the way to the sensitive information.

VI. DISCUSSIONS

Tokenization has been demonstrated in this study as a leading back-end payment security technology of the modern era and a key enabler of effective PCI DSS compliance. By designing and implementing a vault-based tokenization solution on a test dataset comprising 449 transaction records, the study empirically validated tokenization's primary benefit—an unprecedented reduction in the scope of the Cardholder Data Environment (CDE). The results revealed a 75% reduction in in-scope systems for high-priority data security requirements, confirming that tokenization confines sensitive cardholder data to a small, manageable, and highly secure vault.

This architectural transformation, as illustrated across the tables and figures in the study, effectively removes the majority of an organization's IT assets from the costly and restrictive reach of PCI DSS audits. The study also evaluated system performance, showing that latency introduced by tokenization and detokenization operations was minimal and increased linearly—indicating that enhanced security does not compromise user experience or transaction throughput.

A simulated token-based attack further validated the approach: only non-sensitive tokens were exposed, demonstrating the inherent resilience and intelligence of the tokenization framework. By substituting real data with anonymized tokens, the system neutralizes potential breach targets, effectively eliminating the risk of data compromise.

Overall, tokenization provides a dual advantage—it simplifies PCI DSS compliance while significantly strengthening an organization's cybersecurity posture against sophisticated threats. This study substantiates the effectiveness of vault-based tokenization and identifies promising directions for future work, including exploring vault-less tokenization methods that use cryptographic techniques to generate tokens without centralized storage. Comparative analysis between vaulted and vault-less models in terms of security, performance, and compliance can further aid organizations in making well-informed adoption decisions.

979-8-3315-6755-2/25/$31.00 ©2025 IEEE

VII. CONCLUSION

Tokenization has been demonstrated in this study as a leading back-end payment security technology of the modern era and a key enabler of effective PCI DSS compliance. By designing and implementing a vault-based tokenization solution on a test dataset comprising 449 transaction records, the study empirically validated tokenization's primary benefit—an unprecedented reduction in the scope of the Cardholder Data Environment (CDE). The results revealed a 75% reduction in in-scope systems for high-priority data security requirements, confirming that tokenization confines sensitive cardholder data to a small, manageable, and highly secure vault.

This architectural transformation, as illustrated across the tables and figures in the study, effectively removes the majority of an organization's IT assets from the costly and restrictive reach of PCI DSS audits. The study also evaluated system performance, showing that latency introduced by tokenization and detokenization operations was minimal and increased linearly—indicating that enhanced security does not compromise user experience or transaction throughput.

A simulated token-based attack further validated the approach: only non-sensitive tokens were exposed, demonstrating the inherent resilience and intelligence of the tokenization framework. By substituting real data with anonymized tokens, the system neutralizes potential breach targets, effectively eliminating the risk of data compromise.

Overall, tokenization provides a dual advantage—it simplifies PCI DSS compliance while significantly strengthening an organization's cybersecurity posture against sophisticated threats. This study substantiates the effectiveness of vault-based tokenization and identifies promising directions for future work, including exploring vault-less tokenization methods that use cryptographic techniques to generate tokens without centralized storage. Comparative analysis between vaulted and vault-less models in terms of security, performance, and compliance can further aid organizations in making well-informed adoption decisions.

REFERENCES

[1] K. Maran, P. Priyadarshini, L. Jenifa, C. R. Senthilnathan and P. Venkatesh, "*Data Analysis on Mobile Payment Technology with Reference to Users' Behaviour of Retail Goods in India,*" 2021 4th International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2021, pp. 267-272, doi: 10.1109/ICCCT53315.2021.9711823.

[2] R. K. Singhal, P. Chauhan and T. Pandey, "*Exploration of Factors Affecting Adoption of Digital Wallet Among Indian Domestic Tourist: Study of Trust and Security Perception*," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 1268-1271, doi: 10.1109/ICRITO48877.2020.9197917.

[3] Fang, C.C.; Liou, J.J.H.; Huang, S.-W.; Wang, Y.-C.; Huang, H.-H.; Tzeng, G.-H. *A Hybrid, Data-Driven Causality Exploration Method for Exploring the Key Factors Affecting Mobile Payment Usage Intention*. Mathematics 2021, 9, 1185. https://doi.org/10.3390/math9111185.

[4] T. Oliveira, M. Thomas, G. Baptista, and F. Campos, "Mobile payment: Understanding the determinants of customer adoption and intention to recommend the technology," *Computers in Human Behavior*, vol. 61, pp. 404–414, 2016.

[5] W. Chmielarz and M. Zborowski, "*Analysis of the use of electronic banking and e-payments from the point of view of a client*," 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 2017, pp. 965-969, doi: 10.15439/2017F103.

[6] I. R. De Luna, F. Liébana-Cabanillas, J. Sánchez-Fernández, and F. Muñoz-Leiva, "Mobile payment is not all the same: The adoption of mobile payment systems depending on the technology applied," *Technological Forecasting and Social Change*, vol. 146, pp. 931–944, 2019.

[7] Y. Patel, N. Chovatia and H. Kaur, "*Securing Payment Transactions: A Comprehensive Review of Smart Cards and Contactless Payments with Cryptographic Methods,*" 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2024, pp. 785-790, doi:.

[8] C. Flavián, M. Guinaliu, and Y. Lu, "Mobile payments adoption—Introducing mindfulness to better understand consumer behavior," *International Journal of Bank Marketing*, vol. 38, pp. 1575–1599, 2020.

[9] F. Liébana-Cabanillas, A. Japutra, S. Molinillo, N. Singh, and N. Sinha, "Assessment of mobile technology use in the emerging market: Analyzing intention to use m-payment services in India," *Telecommunications Policy*, vol. 44, p. 102009, 2020.

[10] Q. Zhang, S. Khan, M. Cao, and S. U. Khan, "Factors determining consumer acceptance of NFC mobile payment: An extended mobile technology acceptance model," *Sustainability*, vol. 15, p. 3664, 2023.

979-8-3315-6755-2/25/$31.00 ©2025 IEEE