Visa Best Practices for Tokenization Version 1.0

Introduction

In October 2009, Visa published the Visa Best Practices for Data Field Encryption to promote the proper encryption of sensitive card data that is transmitted, processed or stored by stakeholders throughout the payment system. As part of these best practices, Visa recommended that entities use tokens (such as a transaction ID or a surrogate value) to replace the Primary Account Number (PAN) for use in payment-related and ancillary business functions.

Tokenization can be implemented in isolation or in concert with data field encryption to help merchants eliminate the need to store sensitive cardholder data after authorization. Entities that properly implement and execute a tokenization process to support their payment functions may be able to reduce the scope, risks and costs associated with ongoing compliance with the Payment Card Industry Data Security Standards (PCI DSS).

How Tokenization Works

Tokenization defines a process through which PAN data is replaced with a surrogate value known as a "token." The security of an individual token relies on properties of uniqueness and the infeasibility to determine the original PAN knowing only the surrogate value. As a reference or surrogate value for the original PAN, a token can be used freely by systems and applications within a merchant environment.

Where properly implemented, tokenization allows merchants to limit the storage of cardholder data to within the tokenization system, potentially simplifying an entity's assessment against the PCI DSS. As a reference or surrogate value for the original PAN, a token can be used by systems and applications within a merchant environment without having to consider the security implications associated with the use of cardholder data.

The security and robustness of a tokenization system is dependent upon the secure implementation of four critical components, and the overall management of the system and any historical data:

- *Token Generation: Defines the process through which a token is generated.*
- *Token Mapping: Defines the process for associating a token to its original PAN value.*
- *Card Data Vault: Defines the central repository of cardholder data used by the token mapping process.*
- *Cryptographic Key Management: Defines the process through which cryptographic keys are managed and how they are used to protect cardholder and account data.*

Visa Best Practices for Tokenization, Version 1.0

The following are best practices for use of tokenization technology to protect cardholder data:

Domain	Best Practice
Tokenization System	1. **Network Segmentation:** The tokenization system must be adequately segmented from the rest of the network. The tokenization system must be deployed within a fully PCI DSS compliant environment and be subject to a full PCI DSS assessment. 2. **Authentication:** Only authenticated entities shall be allowed access to the tokenization system. 3. **Monitoring:** The tokenization system must implement monitoring to detect malfunctions or anomalies and suspicious activities in token-to-PAN mapping requests. Upon detection, the monitoring system should alert administrators and actively block token-to requests or implement a rate limiting function to limit PAN data disclosure. 4. **Token Distinguishability:** The tokenization system must be able to identify and distinguish between tokenized and cleartext cardholder data and avoid the propagation of tokens to systems expecting cleartext cardholder data. **Note**: In accordance with the *Visa Best Practices for Data Field Encryption*, cardholder data must remain encrypted from the point where it enters an entity's system up to the point it is tokenized to achieve the full benefits of a tokenization solution.
Token Generation	5. **Token Generation**: Knowing only the token, the recovery of the original PAN must not be computationally feasible. Token generation can be conducted utilizing **either**: • A known strong cryptographic algorithm (with a secure mode of operation and padding mechanism), **or** • A one-way irreversible function (e.g., as a sequence number, using a hash function with salt or as a randomly generated number)

	6. **Single-use vs. Multi-use Tokens**: Tokens can be generated as a single- or multi-use surrogate value, the choice of which depends largely on business processes: • **A single-use token** should be used when there is no business need to track an individual PAN for multiple transactions. Acceptable methods for producing a single-use token include, but are not limited to, hashing of the PAN with a transaction-unique salt value, using a unique sequence number, or encrypting the PAN with an ISO- or ANSI-approved encryption algorithm using a transaction-unique key. • **A multi-use token** should be used when there is a business need to track an individual PAN for multiple transactions. A multi-use token will always map the same input PAN to the same token. An acceptable method for producing a multi-use token includes, but is not limited to, hashing of the cardholder data using a fixed but unique salt value per merchant. **Notes**: If a salt value is used, the salt must be kept secret and appropriately protected. A salt should have a minimum length of 64-bits. If a token is generated as a result of using a hash function, truncated PAN data **must not** be stored or transmitted in conjunction with the tokenized data.
Token Mapping	7. **PAN Processing:** In order to limit / eliminate storage of PAN data, the tokenization system should not provide PAN data to a token recipient (e.g., a merchant). If PAN data is returned, the receiving system will be in the scope of the PCI DSS. The token mapping should perform the following: • Processes should not return the PAN as part of any response to the merchant. • The tokenization platform should allow for chargeback and refund processing without the need for the merchant to retain or have access to full PAN.
Card Data Vault	8. PAN data must be encrypted in storage. 9. The card data vault must be managed and protected per PCI DSS requirements.
Cryptographic Key	10. Encryption keys shall use a minimum bit strength of 112 bits.

<table>
<tr><td>Management</td><td>The following table summarizes equivalent bit strengths for commonly used approved algorithms:
<table><tr><th>Algorithms</th><th>Bit Length</th></tr><tr><td>TDES</td><td>112[1]</td></tr><tr><td>AES</td><td>128[2]</td></tr><tr><td>RSA</td><td>2048</td></tr><tr><td>ECC</td><td>224</td></tr><tr><td>SHA</td><td>224</td></tr></table>
11. Any cryptographic keys used by the tokenization system must be managed in accordance with PCI DSS.</td></tr>
<tr><td>Management of Historical Data</td><td>12. Any retained historical or existing repositories of cardholder data must be protected (per PCI DSS requirements), tokenized or eliminated as part of the implementation.</td></tr>
</table>

Conclusion

Visa supports tokenization as a means of replacing Primary Account Numbers (PANs) with non-sensitive surrogate values (known as “tokens”) to eliminate or reduce storage of cardholder data. Tokenization can be implemented independently or in concert with data field encryption for the protection of cardholder information. To support marketplace adoption of tokenization, Visa has developed best practices to assist merchants and other stakeholders in evaluating and adopting tokenization solutions. These best practices should be viewed as high level guidance to be considered for any such solution to assist stakeholders in the Visa payment system.

Respond With Comments by August 31, 2010

Visa would appreciate stakeholder feedback on these best practices by August 31, 2010. Please submit any comments via e-mail to inforisk@visa.com with "Best Practices for Tokenization" in the subject line.

Related Documents

“Visa Best Practices for Data Field Encryption” – October 2009

“Visa Best Practices for Primary Account Number Storage and Truncation” – July 2010

---

[1] *For the purpose of these best practices, two key TDES (112-bits) should not process more than 1 million transactions. In cases where the number of transactions potentially processed through the system using a single 112-bits TDES key greatly exceeds 1 million, three key TDES (168-bits) or AES should be used. Key management schemes that greatly limit the number of transactions processed by a single key, such as Derived Unique Key Per Transaction (DUKPT), can be used to ensure that any individual key is used only a limited number of times..*

[2] *The smallest key size usable with AES is 128 bits. This key size is stronger than needed, but if AES is to be used, it is the smallest available. Longer keys may be used if so desired.*