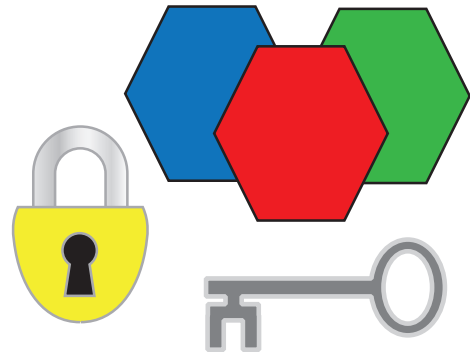


Tokenization and Enterprise Data Security

By Gary Palgon

To meet more rigorous security challenges, tokenization, a new data security model, is gaining traction.



The risk of data theft remains high despite the best efforts of security officers and the passage of numerous data security mandates and privacy laws. As organizations lock down sensitive and confidential data in one area, cybercriminals go after it in another. There is also the very real risk of internal theft or accidental loss.

Many retail merchants are doing a good job of protecting payment card data thanks to the Payment Card Industry's Data Security Standard (PCI DSS). Yet, as they gain experience with encryption, masking, hashing, enterprise key management, and the full life cycle of protection and compliance, they are finding that they want to explore other types of data protection architectures as well as protection for other types of customer and company data at risk. In fact, whether in response to privacy laws or the desire to better protect customers and employees, organizations of all types are moving beyond securing credit card numbers to seeking ways to guard the many types of personally identifiable information (PII) under their care. This presents new data encryption and storage challenges because PII data can be harder to locate and lock down. It typically resides in many places throughout an enterprise and trying to secure it can be a complex, resource-intensive exercise.

Part of the problem in securing PII is that encrypted data uses more space than cleartext data,¹ and many forms of PII contain many more characters than a 16-digit credit card number – all of which can pose a storage problem. This can also create field space issues within applications. If, for example,

the ciphertext version of a bank account number requires more space than allowed by the application, the application must be modified to accommodate a larger field.

To meet the more rigorous security challenges posed by protecting diverse types of information, a new data security model is beginning to gain traction – tokenization. Tokenization provides two distinct benefits that build on solid strong-encryption practices. First, it reduces the number of instances of sensitive data in an organization, and second, it reduces the scope of a PCI DSS audit.

Tokenization unwrapped

With traditional encryption, when a database or application needs to store sensitive data those values are encrypted and the ciphertext is returned to the original location. With tokenization, a token – or surrogate value – is returned and stored in place of the original data. The token is a reference to the actual ciphertext, which can be stored locally (“in-place tokenization”) or, in a newly-emerging model, in a central data vault. As long as the token is format-preserving, it can be safely used by any file, application, database, or backup medium throughout the organization, thus minimizing the risk of exposing the actual sensitive data and allowing business and analytical applications to work without modification. Note also that format-preserving tokens can simply match the expected data type or, more importantly, can expose a subset of the original value to simultaneously protect the information and enable applications and job functions to continue unmodified. For example, the token could expose the last 4 digits of the social security number or credit card number to enable call center operations.

Tokenization is an alternative data protection architecture that is proving ideal for some organizations' requirements. It reduces the number of points where sensitive data is stored within an enterprise, making it easier to manage and secure. Tokens use the same amount of storage space as the origi-

¹ Almost all encrypted data requires more space than cleartext data and different types of encryption require different field sizes. The better encryption algorithms, including AES, Triple-DES, RC6 and RSA, use what is called block ciphers to segment information to be encrypted. These algorithms encrypt several bits of plaintext data in one step, called an encryption block, which will vary in size along with the actual size of the block. AES, Triple-DES, RC6 and RSA all use different size blocks and calculate different lengths of the encrypted value. The AES algorithm, for example, can use 128, 192 or 256-bit blocks. To illustrate, a 20-byte cleartext data field encrypted using the AES algorithm with a 256-bit key and exported using Base64 encoding creates 44 bytes of encrypted data.

nal cleartext data instead of the larger amount of storage required by encrypted data. Furthermore, a token is not mathematically derived from the original data, making it arguably safer than exposing ciphertext. A token can be passed around the network between applications, databases, and business processes safely, all the while leaving the encrypted data it represents securely stored in a central data vault. Authorized applications that need access to encrypted data can only retrieve it using a token issued from a token server, providing an extra layer of protection for sensitive information and preserving storage space at data collection points.

Tokenization in practice

There are two distinct scenarios where implementing a token strategy can be beneficial:

- Reduce the number of places sensitive encrypted data resides
- Reduce the scope of a PCI DSS audit

The hub and spoke model is the same for both and contains three components: a centralized encryption key manager to manage the life cycle of keys, a token server to encrypt data and generate tokens, and a central data vault to hold the encrypted values, or ciphertext. These three components comprise the hub. The spokes are the endpoints where sensitive data originates. Spokes can be, for example, the point-of-sale terminals in a retail store or the servers in a department, call center, or website.

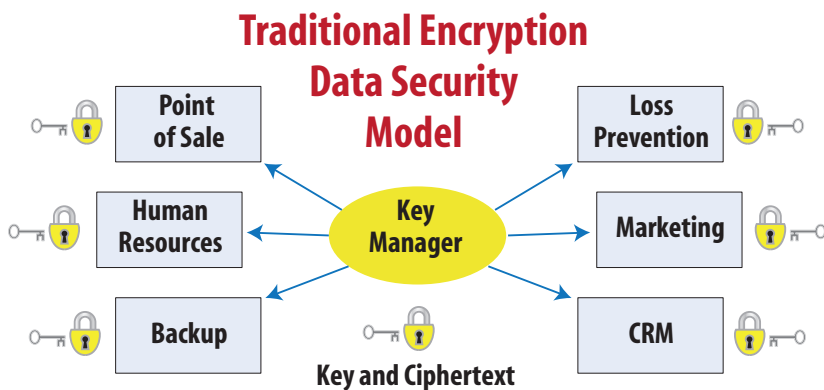


Figure 1 – Traditional Encryption Data Security Model

In the traditional data protection model, data is encrypted at the spokes and stored there or encrypted at headquarters and distributed back out to the spokes (Figure 1). Under the tokenization model, encrypted data is stored in a central data vault and tokens replace the corresponding ciphertext in applications available to the spokes, thereby reducing the instances where ciphertext resides throughout the enterprise (Figure 2). This significantly reduces risk because the only place encrypted data resides is in the central data vault until it is needed by authorized applications and employees.

In the second scenario, the model is the same but the focus is not on the advantage of a central data vault but rather the ability to utilize only tokens in spoke applications, thereby reducing scope for a PCI DSS audit. In this case, employees

Tokenization Data Security Model

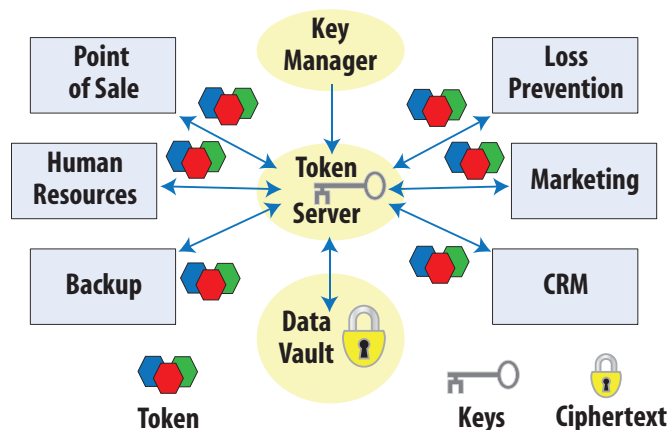


Figure 2 – Tokenization Data Security Model

will only need a “format-preserving token” where the token provides enough insight to perform their job function. For instance, the token will contain the last four digits of a credit card. In the traditional encryption model, ciphertext resides on machines throughout the organization both at the central hub and all of the spokes. All of these machines are “in scope” for a PCI DSS audit. In the tokenization model, many of the spokes can use the format-preserving tokens in place of ciphertext, which takes those systems out of scope for the audit.

The following use cases illustrate the two common scenarios that tokenization benefits.

Use case: National retailer

The risk management group at a large national retailer had implemented strong encryption and a centralized key management system to protect customer credit card information at its headquarters and stores located throughout the U.S. After performing an audit, they determined that even though the credit card and customer loyalty data were encrypted, there was still some amount of risk associated with storing ciphertext at their retail stores on point-of-sales systems, merchandise return systems,

and servers prior to the data flowing back to headquarters for storage.

The risk management group began looking for ways to reduce this risk and determined that tokenization could reduce the number of places where sensitive data would exist within the organization. Using tokenization, the retailer was able to eliminate the storage of encrypted credit card information in all of its retail stores and confine it to the central data vault at headquarters and to a backup data center.

Now, when a credit card is used for a purchase, the number is immediately encrypted and transmitted in real time to the central data vault. A token representing that data is generated and returned to the store to take the place of the credit card number in the application. Once the data moves from

Tokenization Data Flow

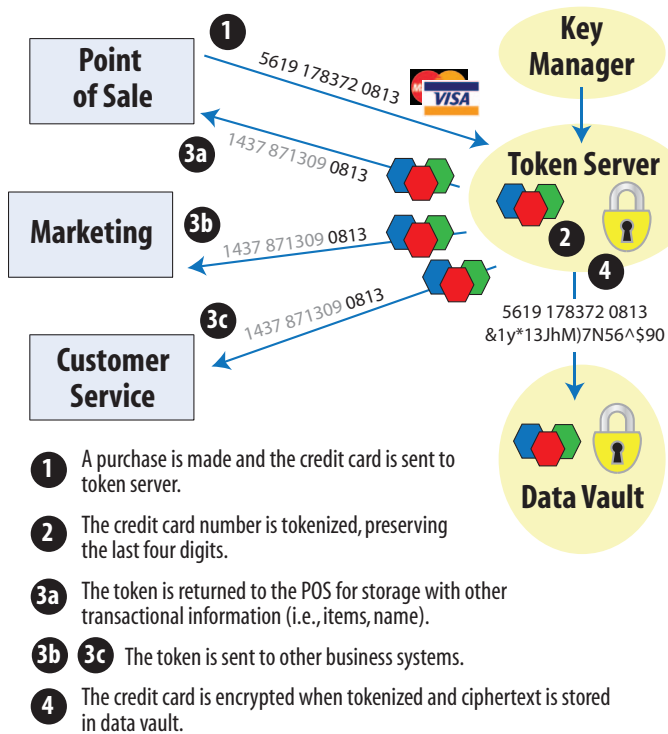


Figure 3 – Tokenization Data Flow

the store location back to the central data vault at corporate headquarters, the cleartext value is only available at the store to those with proper authority and then only for as long as it is needed to complete a transaction, such as decrypting a credit card number to facilitate a merchandise return. This limits the risk to the data vault and off-site backup data center for disaster recovery.

While protecting credit card information is of primary importance, the risk management team also recognized that protecting customer loyalty information and other PII was also necessary to reduce risk. The retailer had in its possession PII from millions of customers, some of which was public information such as names and addresses. However, it also had other types of information that customers regard as private, including items purchased, clothing sizes, payment history, mobile phone numbers, and email addresses, as well as user names, passwords, and password hints collected from online store purchases. The team extended tokenization to protect PII. Now, as customers provide PII information, the systems that require some of the more sensitive data are also tokenized and access to that data follows the same model as tokenization for payment card numbers.

The retailer also uses a data warehouse where they analyze individual customer buying habits. Based on the results of the analysis, the retailer sends its customers special offers based on past purchases. The retailer uses the customer's credit card number on file to link to actual items purchased. Using tokenization, the marketing department can still perform this analysis, but without

using actual credit card numbers. By tokenizing the credit card numbers, the data warehouse no longer contains credit card information, thereby eliminating the risk of theft but allowing the buying behavior analysis to still be performed.

Use case: Cable company

A cable company recently began tokenizing payment card information, but for a different reason than the retailer above. The security team at the cable company wanted to reduce the PCI DSS audit process by taking as many systems out of scope as possible.

The cable company keeps a bank account number or a credit card number on file for each customer who opts for automatic payment. Various individuals in the company need access to customer records to perform their jobs, but only the finance department needs to be able to use a bank account or credit card number for monthly billings. For example, a tech support representative needs to be able to see the last four digits of a customer's credit card number to verify the identity of the caller, but not the entire value.

In this tokenization model, the technical support department is a spoke in the "hub and spoke model," as are the sales call center, the finance department, business services, and installation departments. The company now uses tokens in lieu of sensitive data in customer files. Using tokenization, only those employees who need to access to customer confidential information will have access to ciphertext and have the authority to decrypt it. Employees in other departments will only see format-preserving token data in place of ciphertext,

Retrieve Credit Card from Token

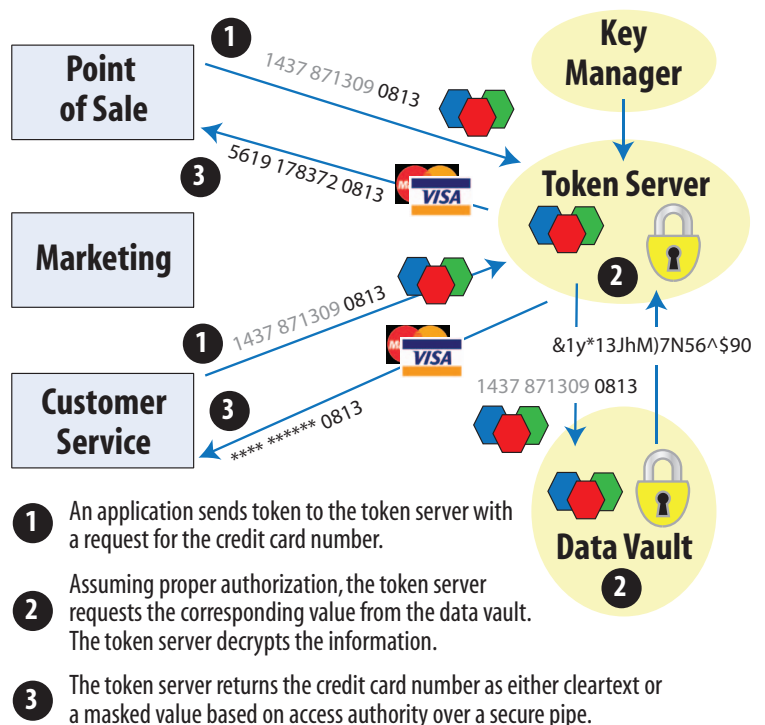


Figure 4 – Retrieve Credit Card from Token

removing the risk of unauthorized people decrypting and using this information.

In this case, the first 12 digits of a credit card number are tokenized so that they appear as nonsensical characters in the application. The last four digits appear in cleartext so that the tech support department can verify customer identities. Unlike ciphertext that takes up more room than cleartext, tokens take the same amount of room as the cleartext credit card value they replace. Using a format-preserving token, the application does not have to be modified. The combination of tokens and cleartext data in the customer record is comprised of 16 digits, the same as the credit card number.

When a customer wants to change the credit card on file, the tech support representative now simply transfers the customer to the billing department instead of taking the number. Only billing department employees have the authority to record a credit card number. When a replacement credit card number is taken, it is immediately encrypted and stored in the central data vault. The token server then generates a new token for the new credit card number, which is returned to the billing system for future use as well as in the customer record that is used by other departments. As a result, the applications and systems that contain tokens instead of cleartext or encrypted credit card information no longer have to be audited. They are rendered “out of scope” for PCI DSS compliance and pose no risk for theft.

While the focus of the cable company was initially to provide greater security for customer payment card numbers and reduce the amount of work they had to do to pass each PCI DSS audit, the IT department is now in the process of applying tokenization to secure customer loyalty information.

Conclusion

For companies that want to further reduce the risk of data theft – from both cybercriminals and internal data theft or accidental loss – or reduce the scope of the annual PCI DSS audit, tokenization enhances the security provided by strong encryption without modifications to applications. Requiring an encryption key manager, token server, and central data vault, tokenization is relatively easy to implement and provides an extra layer of defense for organizations that want to better secure the payment card numbers and personally identifiable information under their protection.

About the Author

Gary Palgon is vice president of product management for data protection software vendor nuBridges, Inc. He is a frequent contributor to industry publications and a speaker at conferences on eBusiness security issues and solutions. Gary can be reached at gpalgon@nubridges.com.

