First Data
beyond the transaction

A First Data White Paper

What Data Thieves Don't Want You to Know: The Facts About Encryption and Tokenization

Introduction

In 2011, 535 data breach incidents were perpetrated in the United States, resulting in the theft of over 30 million sensitive consumer records—including millions of debit and credit card account numbers.[1] Data breaches are constantly in the news, and recent high profile cases show that no organization is immune—especially as criminals develop increasingly sophisticated methods to exploit vulnerabilities in the payment system.

All merchants—whether they are brick-and-mortar, brick-and-click, or completely web-based—have both an obligation and an industry mandate to protect consumers' payment card data. The Payment Card Industry (PCI) Data Security Standards (DSS) provide guidelines on what merchants need to do to secure the sensitive data used in payment transactions. End-to-end encryption (E2EE) and tokenization solve for many of the vulnerabilities that exist in the payments processing chain. Encryption mitigates security weaknesses that exist when cardholder data has been captured but not yet authorized, and tokenization addresses security vulnerabilities after a transaction has been authorized. When combined, these two technologies provide an effective method for securing sensitive data wherever it exists throughout its lifecycle.

This paper is an overview of encryption and tokenization technologies—what they are, how they can be implemented, and the benefits and drawbacks of selecting a particular method of implementation. The purpose of the paper is to help merchants understand enough about the technologies to begin exploring how encryption and tokenization can be useful in their own environment.

Merchant Vulnerabilities in the Payment Process

Card data for a purchase transaction must flow through multiple systems and parties in order to be processed. This processing chain—which includes consumers, merchants, acquirers/processors, card brands and issuing banks—links many technologies, including communication lines, databases and sophisticated applications. Data thieves have become quite knowledgeable about how these technologies work, enabling them to exploit points of vulnerability across the payments processing chain.

For most merchants, there are two points in the payment process where sensitive cardholder data is at risk of being exposed or stolen:

1. Pre-authorization – When the merchant has captured a consumer's data and it is being sent or is waiting to be sent to the acquirer/processor.
2. Post-authorization – When cardholder data has been sent back to the merchant with the authorization response from the acquirer/processor, and it is placed into some form of storage in the merchant environment and used for analytics and other back-office processes.

It is incumbent upon a merchant to be confident that it is securing both parts of the process.

[1] Privacy Rights Clearinghouse, "Data Breaches: A Year in Review," December 16, 2011.

Technologies that Address the Security Gaps: Encryption and Tokenization

Fortunately, there are highly effective technologies available to address these two specific points of vulnerability. The information that follows is intended to help explain these technologies and describe where they fit in a merchant's environment.

Encryption

Encryption is the process of using algorithmic schemes to transform plain text information into a non-readable form called ciphertext. A key is required to decrypt the information and return it to its original plain text format.

Why encryption is important

Anytime that live cardholder data is in the clear—that is, in plain text format that is readable by a person or computer—it is extremely vulnerable to theft. Of course, criminals know this and look for ways to capture that data. For example, it's possible for a thief to siphon off the card data as it is transmitted in plain text from a card reader to the point of sale (POS) server or the merchant's central server. (This is what is suspected to have happened in several prominent data breaches.)

Encryption of either the data itself or the transmission path the data takes along the network, or both, can vastly reduce the vulnerability of the data, which in turn reduces a merchant's business risks.

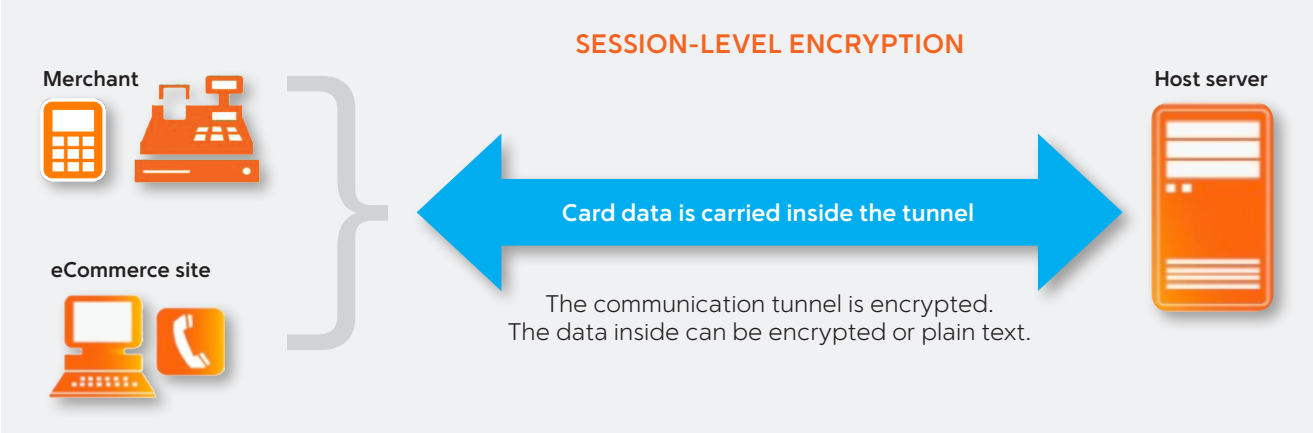
Multiple approaches to encryption within the payment process

There are multiple approaches to encryption in the payment process. A merchant will need to evaluate its own environment to determine which approach or approaches would work best to meet its needs.

Session-level encryption ("encrypt the pipe")

In session-level encryption, the communication path in which the transaction flows from point A to point B is encrypted; for example, from a POS terminal to a store's central host, or from a consumer's PC to an e-commerce web page. Think of the communication path as a tunnel, with transaction data flowing within the tunnel. With session-level encryption, the tunnel is layered with armor (i.e., encryption) so that it cannot be penetrated. The sensitive data inside the tunnel—which may be in clear text—is protected by the armor shield.

Session-level encryption is commonly used when the merchant doesn't control the path all the way out to the end user. Most notably, this is the case when purchases are made over the Internet. It's not practical for a merchant to encrypt the data on a consumer's PC, but it is easy to establish encryption for the communication session between the PC and the e-commerce web page. This is commonly called a secure socket layer, or SSL, and is often denoted by a lock icon on a web page. Using SSL, all the information sent between the PC and the host server travels through an encrypted tunnel.

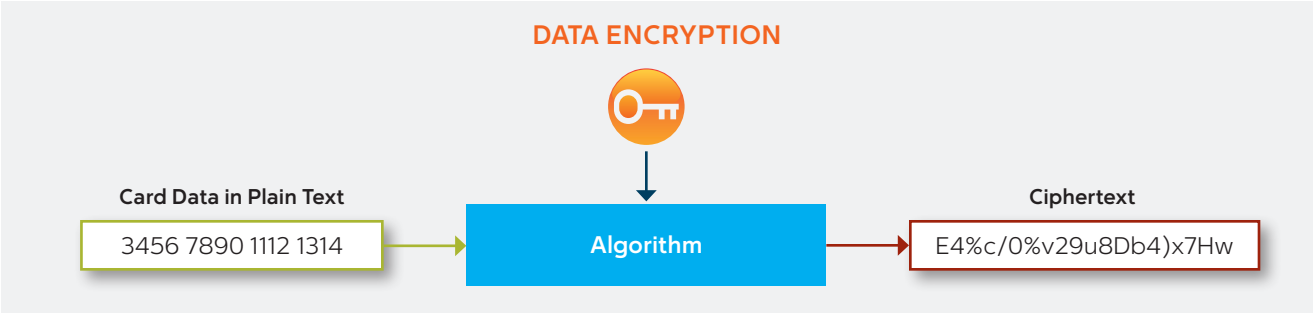


A merchant may find that depending entirely on session-level encryption alone may not be adequate, however. There are points along the way of the payment process where data transitions out of one encrypted tunnel and into the next. For a millisecond of time, the data is no longer in any kind of encrypted session and is vulnerable to theft. For example, data can travel through an encrypted path to go from the POS to a store server. The data then flows into a separate encrypted session to go from the store server to the acquirer/processor. There are known breach cases where thieves have injected malware at that transition point to gather card data in the clear and send it off to other servers for illicit purposes.

Another drawback of session encryption is that it keeps the outsiders out but does nothing to protect a merchant from insider fraud. A malicious employee could actually be the one planting the malware or other type of skimmer that steals the plain text data at vulnerable points.

Data-level encryption ("encrypt the payload")

In data-level encryption, the payload within the tunnel is encrypted. That is, encryption is applied to sensitive data elements such as the card number, the track data, the card security code (i.e., CVV, CVV2) and the expiration date.



Depending on where in the process the data elements are encrypted, the merchant could be protected from internal fraud as well as external fraud. In general, encrypting as close to the point of entry or capture is preferable—since the data will be unprotected for the least amount of time. If the card data that a merchant wants to protect is encrypted at the point of capture—for example, at the customer-facing PIN entry device in a multi-lane retailer or at the data entry web page of an e-commerce site—and if that data stays encrypted until it is received by the processor, the data is protected all along the way. This is what often is called end-to-end encryption (the PCI SSC prefers the term, "point-to-point" encryption since the data is usually decrypted before it reaches its final or end destination). Even if the transaction is intercepted at some point along the way, the encrypted card data is unreadable and means nothing to anyone other than the processor that holds the decryption key.

Where possible and practical, data-level encryption is preferable to having only session-level encryption. Of course, a merchant can combine session encryption with data encryption, whereby encrypted data moving through an encrypted tunnel would be doubly secured.

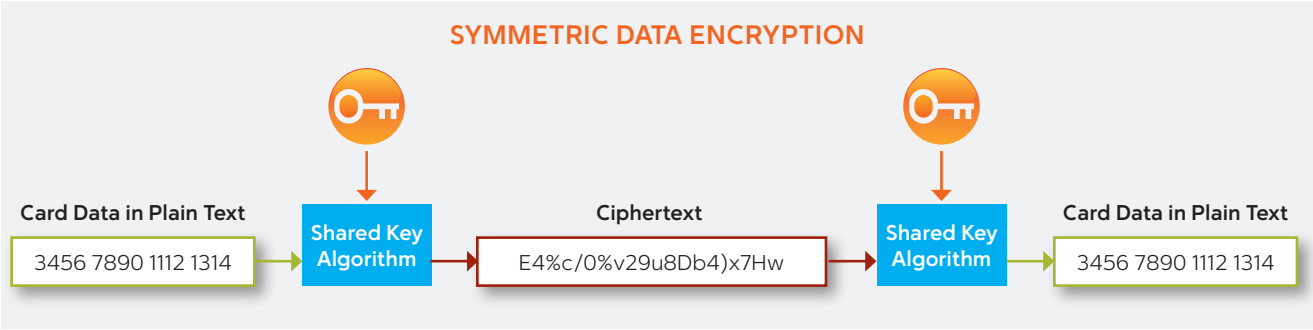
Multiple methods of data-level encryption

There are multiple ways that data-level encryption can be applied. Again, which method is most appropriate depends on a merchant's specific environment.

Symmetric encryption (single key)

The mathematical algorithm that turns plain text into ciphertext, or vice versa, requires the use of a key. In symmetric encryption, the key is a shared secret used to both encrypt and decrypt the data. Liken this to the lock on a door, where one key can both lock and unlock the door.

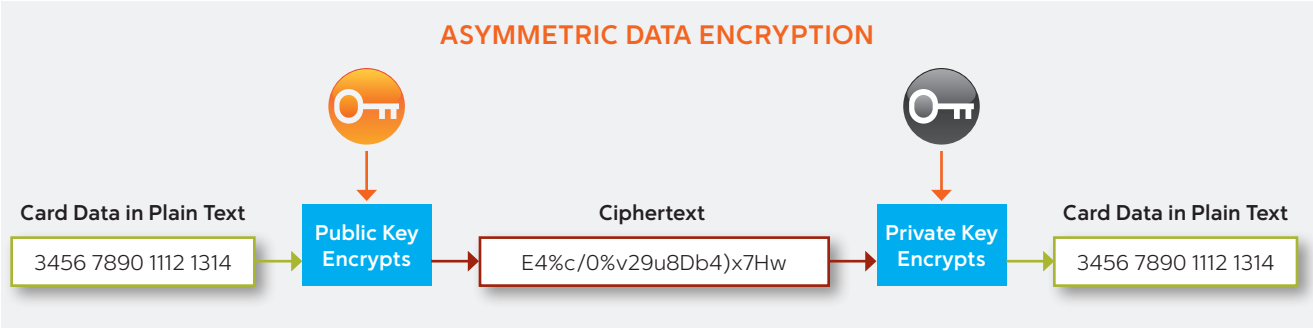
In a payment processing environment, a symmetric key that encrypts sensitive cardholder data can also decrypt it. Therefore, multiple security mechanisms need to be built into the encryption lifecycle in order to protect the key.



One of the most common security mechanisms is key rotation, or changing the key periodically to reduce the amount of harm that can be done if the key is compromised. The requirements for key management include determining how often the key should be rotated, how to securely distribute the key to the encryption point(s), whether data that was encrypted with an old key should be decrypted and re-encrypted with a new key, and where and for how long old keys should be stored.

Asymmetric encryption (public key/private key)

Asymmetric encryption uses two separate keys, each of which has a specific function. One key encrypts the data, while a different key decrypts the data. The encryption key can be shared freely, hence it is often referred to as the public key. The public key can be distributed without the key management challenges of symmetric keys, since it can only encrypt and never decrypt data.



In a payment environment, the public key can be distributed to a merchant or to the end POS device, and that device can store the key in hardware or software. Even if that key is extracted by someone who shouldn't have rights to it, all that the person can do is encrypt data with the key; he can't decrypt anything. On the other hand, the corresponding private key, which enables decryption, must be handled very securely.

Encryption Algorithms: Format Preserving vs. Non-Format Preserving

Format-preserving ciphertext looks like the data that came in. For example, a 16-digit credit card number would be encrypted in a way that produces 16-digit ciphertext output. Format-preserving encryption may be easier to implement because it requires fewer changes to systems in the payment stream. However, it is exclusively symmetric, making the key management more complex.

Considerations for key management

There are many considerations for key management, including who holds the keys; how they are generated and distributed; the process for rotation (i.e., creating new and retiring old keys); and how the keys are protected when stored. Without the proper handling of keys during their life cycle, the keys could be disclosed, modified, or substituted by unauthorized personnel who could then intercept sensitive cardholder data.

The U.S. National Institute of Science and Technology (NIST) provides detailed guidelines for key management. Those details are beyond the scope of this document, but suffice it to say that encryption key management is a sophisticated process requiring significant effort and expertise. A merchant can extricate itself from much of the process by outsourcing key management to a third party service provider.

Data encryption in hardware (TRSM)

The process of encrypting cardholder data can be done with hardware, in a tamper resistant security module (TRSM). A TRSM device has the ability to destroy itself and render useless any data or keys stored in it if someone attempts to tamper with it. A merchant that is using symmetric data encryption should always store the key in a TRSM device.

There are models of card readers that have a TRSM inside so that data can be encrypted immediately at the point of capture. Hardware-based encryption offers a higher degree of overall security than software-based encryption

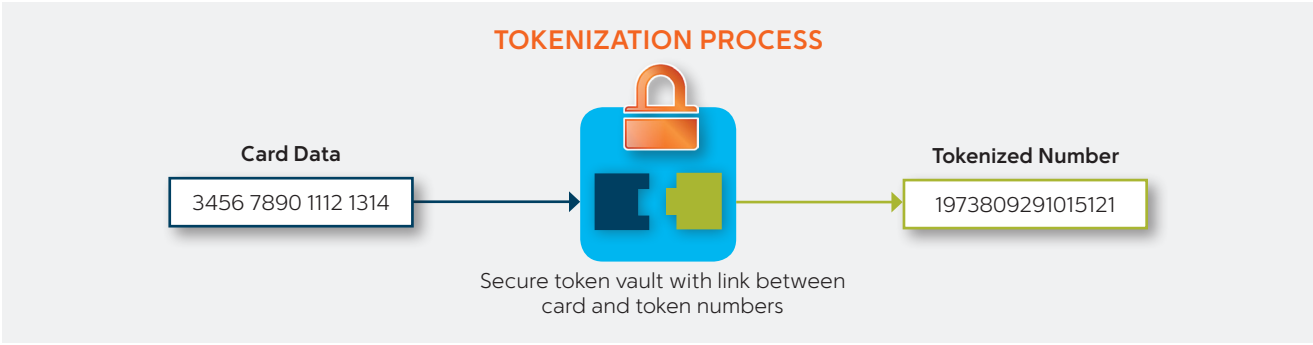
because it prevents key tampering or theft; it is considered "the best of best practices." While deployment may involve considerable hardware costs, it should be noted that the PCI SSC deems the combination of hardware-based encryption and hardware-based decryption the optimal method for reducing PCI scope. Deploying this kind of card reader provides excellent security, but deployment may be cost prohibitive for a merchant that must acquire hundreds or thousands of the devices.

Data encryption in software

Data encryption also can be performed by a software program. This approach provides more flexibility in where the encryption takes place, as it can be added to virtually any terminal, POS device or e-commerce server where card data is presented. In addition, software encryption can be used with devices that simply don't have TRSM available to them, such as older pieces of hardware.

Tokenization

An increasingly popular approach for the protection of sensitive data is the use of data substitution with a token as a replacement for a real payment card number. In the process of tokenization, actual cardholder data is used in a payment transaction and, once the transaction is authorized, this sensitive data is sent to a centralized and highly secure server called a "vault," where it is stored. At the same time, a random unique number is generated and returned to the merchant's systems for use in place of the cardholder data. The vault manager maintains a reference database that allows the token number to be exchanged for the real cardholder data if it is needed again for, say, a chargeback. Meanwhile the token number, which cannot be monetized by anyone but the merchant that owns the token, can be used in all subsequent post-authorization business processes.



Why tokenization is important

Tokenization is important for two reasons. First, it vastly reduces a merchant's security risk in the event of a data breach because the process eliminates sensitive cardholder data from a merchant's environment after a transaction has been authorized. If token numbers are breached, they are meaningless to anyone who would attempt to use them because the tokens are simply random numbers. Second, using token numbers instead of real card data in back-end business applications shrinks the merchant's cardholder data environment (CDE) that is subject to PCI compliance requirements and audits. This reduction of PCI scope can save a merchant significant time and money. The PCI Security Standards Council notes: "Tokenization solutions do not eliminate the need to maintain and validate PCI DSS compliance, but they may simplify a merchant's validation efforts by reducing the number of system components for which PCI DSS requirements apply."[2]

[2] PCI Security Standards Council, Information Supplement: PCI DSS Tokenization Guidelines, August 2011

The design of the token

In its most general sense, a token is just a random series of characters that is not created through any reversible means (such as an algorithm). For the purpose of the payment process, however, the design of the token is important. The token number will fit into most merchant environments without significantly disrupting any business processes if the following design considerations are applied:

- The tokenized number has the same number of digits as a real card number. Observation of this rule is important to ensure that tokenized numbers can easily replace real card numbers in ancillary post-authorization applications, such as business analytics or loyalty marketing, without requiring extensive modification of the applications.
- There is some degree of card number preservation, such as the last four digits being the same as in the real card number. In this instance, the tokenized number (or a portion of it) can be printed on the customer's receipt, and he can see that there is a reference to his actual credit card number.
- The tokenized numbers shouldn't start with any of the traditional numbers of the major brands – 3's, 4's, 5's, and 6's. This rule eliminates the likelihood that the random number of the token will match a valid payment card number.
- The numbers of a token will always fail a Mod 10 check. Only valid card numbers can pass a Mod 10 check. If a tokenized number cannot pass the test, there is no way the token can be mistaken for a valid number, and thus used for fraud.

Why format-preserving encryption (FPE) isn't tokenization

There are some potential drawbacks to using format-preserved encrypted numbers instead of tokenized random numbers. Because encryption keys should be rotated regularly, the output value of encrypting a given PAN will change as the key changes. This means the FPE output loses its value for performing merchant analytics. Also, the PCI council has ruled on encryption, saying that encrypted card data is considered compliant, but is still part of the merchant's card data environment. The PCI Security Standards Council notes: "where token generation is based on a reversible encryption method (where the token is mathematically derived from the original PAN through the use of an encryption algorithm and cryptographic key), the resultant token is an encrypted PAN."[3] The reduction in PCI scope is one of the key benefits merchants gain from implementing random number tokenization.

Multiple approaches to tokenization within the payment process

As with encryption, there are multiple approaches to tokenization. Which approach is best for a merchant depends on how the merchant plans to use the tokenized numbers in its business applications.

Card-based tokens

In the case of a card-based token, there is a life-long relationship between a real card number and a tokenized number. Every time a consumer uses his card at a merchant's store, the same token number is extracted from the vault and provided back to the merchant in the auth response. (This assumes the payment processor is the creator/keeper of the tokens.)

The advantage of a card-based token is that the merchant can aggregate a consumer's transactions over time to build a buying history on that customer—at least at the card level, if not at a higher level from a loyalty program. Many merchants want to understand their customers' buying habits, but today it is too risky to compile a database of transactions based on a real card number. Card-based tokens enable this activity without the risk of exposing sensitive data. Moreover, it is easier for a merchant to use an identifying number that is already in use (i.e., for tokenized payments) than a completely artificial one.

[3] PCI Security Standards Council, Information Supplement: PCI DSS Tokenization Guidelines, August 2011

In addition, all the business processes a merchant has built over the years to mitigate losses and provide for data analysis can continue to function if the card-based token is designed properly. (See the section on token design above.)

The permanent association between a card number and a tokenized number helps the merchant perform many important back-office business processes like data analysis and customer profile management. It is also useful for facilitating recurring payments, repeat card-not-present purchases and loyalty transactions.

Transaction-based tokens

In the case of transaction-based tokens (referred to as single-use tokens by PCI), a different token number is generated for each use of a card. A transaction-based token is fine if a merchant simply wants to ensure it isn't storing card data between the time in which the payment was authorized and the time in which the merchant gets paid. The drawback is that the merchant loses the ability to associate the token number to a specific customer for the purpose of other business applications. Transaction-based tokens are better suited to small merchants that do not use post authorization data for any purpose other than collecting their money from the day's transactions.

The greatest advantage of using tokens–whether card-based or transaction-based–is the elimination of actual cardholder data from the merchant's environment, thus vastly lowering the risk of a data breach and the time and the cost of PCI compliance and validation.

The token vault

The token vault is the reference database that stores both the real cardholder data and the linked token numbers. This is the only place where a token value can be exchanged for real data, such as in the case of a chargeback.

Obviously, the vault requires strong security measures to protect it. For example, the card data in the vault would be encrypted; backup copies of the database would be encrypted; physical and virtual access to the server hosting the database would be tightly controlled; and strong user authentication would be used to access the server and the database. Whoever manages the vault has high levels of responsibility and liability.

There are two approaches to managing the token vault: insourced (i.e., performed by the merchant) and outsourced (i.e., performed by a third party such as the payments processor). The approach a merchant chooses depends on how much it wants to reduce its cardholder data environment for PCI compliance, and how well prepared the merchant is to assume full responsibility for vault security. In actual practice, only a few large merchants to date have chosen to insource the token vault management. Most merchants prefer to let an outside provider with more expertise in data security have that responsibility.

When a merchant controls the token vault in its own datacenter, the merchant can eliminate all live cardholder data from its environment except in two instances: capturing the card data for auth processing, and storing the card data in the vault. All other ancillary applications can be made to use token numbers from the vault. So, for example, if the merchant originally had twelve applications that used real cardholder data, after implementing the token vault, the merchant can reduce that number of applications down to two. The CDE has been effectively reduced; however, securing the vault is a huge responsibility (as described above) with many associated costs.

A second option is to outsource the tokenization process to the payments processor or another third party provider. In this scenario, a merchant sends real card data—preferably in an encrypted format—to the payments processor for authorization, and when the auth response is returned, a tokenized number is also sent to the merchant. This approach shrinks a merchant's CDE to the smallest possible footprint: the POS system that holds live pre-authorization card data. It also relieves the merchant of the burden of managing and securing the token vault.

Conclusion

Payment security is complex, with risks and vulnerabilities at every point of the processing chain. The combination of increasingly burdensome PCI compliance costs and constantly emerging new data security threats make it essential for merchants to implement effective risk management technologies to limit costs and avoid the disastrous consequences of a data compromise event.

Encryption and tokenization solve for mutually-exclusive security weaknesses in the payments process, and in doing so, can reduce a merchant's PCI scope and compliance costs. Encryption protects data that has been captured by the merchant but has not yet been used for the transaction authorization process. Tokenization solves the problem of storing and using real card data in business processes that are downstream from authorization.

Although neither encryption nor tokenization is currently required by PCI DSS, the combination of these technologies is widely recognized as the most powerful way to protect against data theft. According to the PCI DSS Council's "Data Storage Do's and Don'ts," merchants should use strong cryptography to render stored cardholder data unreadable, and use other layered security technologies to minimize the risk of exploits by criminals.[4] For more information about implementing a comprehensive data security solution that utilizes end-to-end encryption and tokenization to protect vulnerable card data, please contact your sales representative or visit firstdata.com.

[4] https://www.pcisecuritystandards.org/documents/PCI%20Data%20Storage%20Dos%20and%20Donts.pdf