First Data
beyond the transaction

A First Data White Paper

A Primer on Payment Security Technologies: Encryption and Tokenization

Solutions like end-to-end encryption and tokenization can help merchants go beyond the current requirements of PCI, solving for many vulnerabilities in the payments processing chain. Learn more about the technologies behind end to-end encryption and tokenization—what they are, the different ways they can be implemented, and the benefits and drawbacks of selecting a particular method of implementation.

By:
Tim Horton
Vice President, First Data

Robert McMillon
Director, RSA- The Security Division of EMC

Introduction

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. Following these guidelines will help a merchant vastly improve its information security, but PCI guidelines are just the beginning, not the endpoint—there still can be security gaps and risks in data handling.

Now there are technologies available that merchants can use to go beyond the current requirements of PCI. End-to-end encryption (E2EE) and tokenization solve for many vulnerabilities in the payments processing chain. E2EE addresses 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 a very strong way to secure data.

This paper is a primer on the technologies behind E2EE and tokenization—what they are, the different ways 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 E2EE and/or tokenization can be useful in their own environment.

Merchant Vulnerabilities in the Payment Process

Where merchants are concerned, 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 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.

It is incumbent on a merchant to look at its own environment to understand where vulnerabilities exist and, if necessary, to go above and beyond PCI to close these gaps and reduce the risks of a data breach.

Technologies from the Marketplace that Address the Security Gap

Players in the payments industry are now bringing solutions to market to address these two specific vulnerability points. The information that follows is intended to help explain the 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 (or algorithm) 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, cyberthieves know this and look for ways to capture a copy of 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 data breaches involving Hannaford Bros., TJX and the Dave & Buster's restaurant chain.)

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.

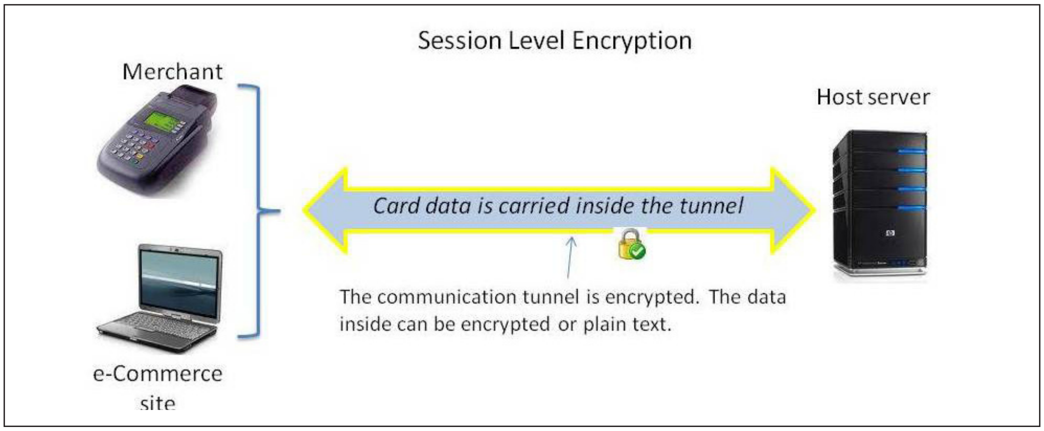
Multiple approaches to encryption within the payment process

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 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 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 yellow lock icon on a web page. Using SSL, all the information sent between the PC and the host server travels through an encrypted tunnel.

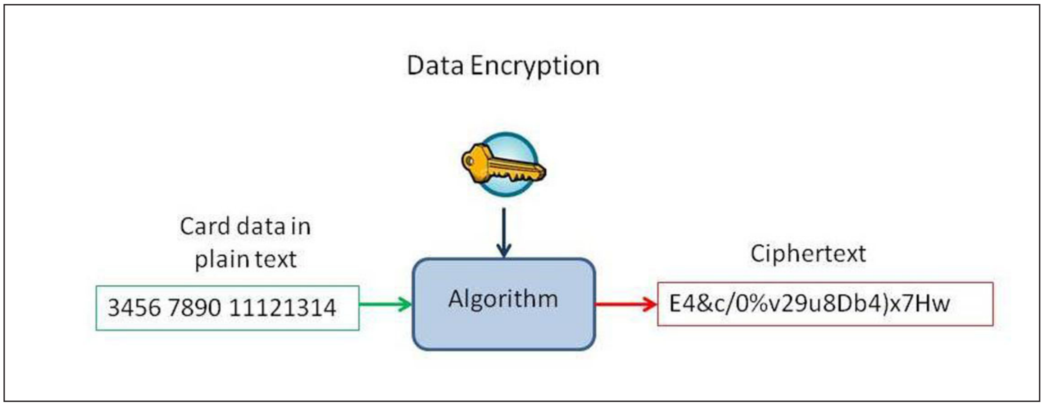

Session Level Encryption
Merchant
Host server
Card data is carried inside the tunnel
The communication tunnel is encrypted. The data inside can be encrypted or plain text.
e-Commerce site

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 the purpose of creating "white plastics" for fraud.

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

Data 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, etc.) 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. 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. Even if the transaction is intercepted at any point along the way, the encrypted card data is unreadable and it means nothing to anyone other than the processor that holds the decryption key.

Where possible and practical, data encryption is preferable to having only session level encryption. Of course, a merchant can combine session encryption with data encryption for a "belt and suspenders" approach to security. Encrypted data moving through an encrypted tunnel would be doubly secured.

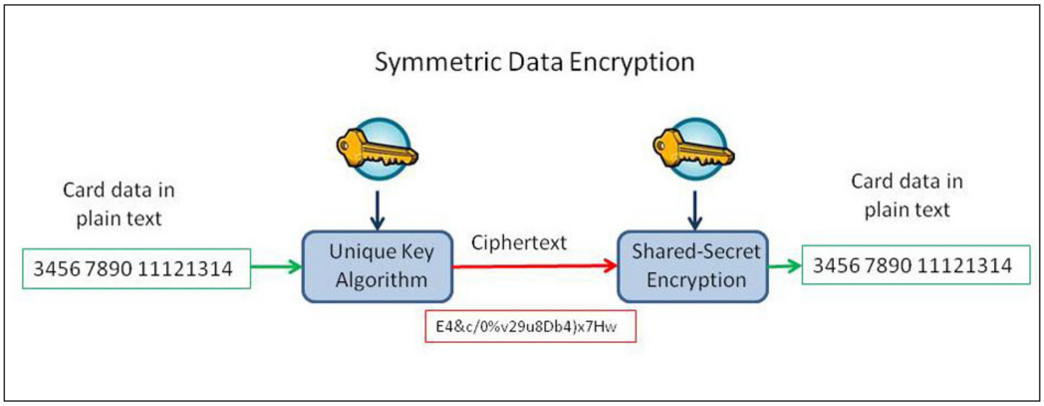
Multiple methods of data encryption

There are multiple ways that data encryption can be applied. Again, which method is "best" 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 "secret," called a key. In symmetric encryption, the key is a shared secret used to both encrypt and decrypt the data. Compare this to the lock on a door, where one key can both lock and unlock the door.

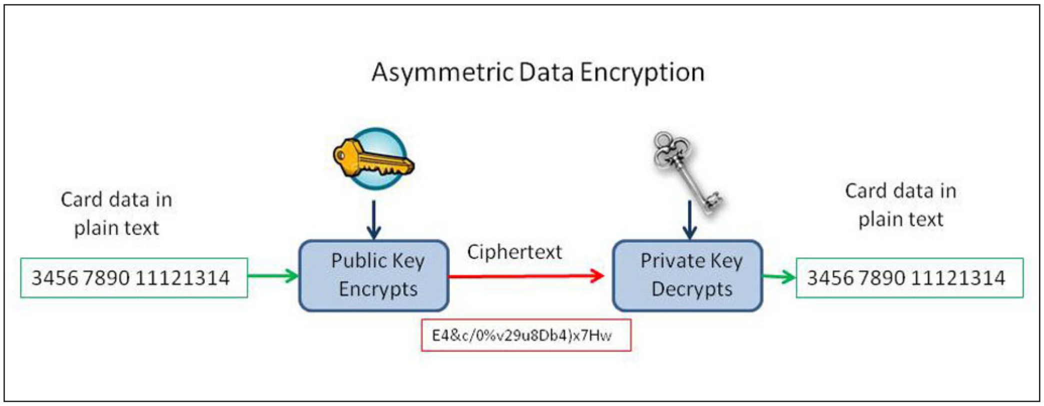
In a payment processing environment, a symmetric key that encrypts sensitive cardholder data can also decrypt it. A merchant wouldn't want to have too much data being encrypted with a single symmetric key because if that key is compromised, then all the data is compromised. Therefore, multiple security 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. But key rotation introduces its own set of challenges. How often should the key be rotated? How do you securely distribute the key to the encryption point(s)? What should be done about data that was encrypted with an old key? Should it be decrypted and re-encrypted with a new key? Where should old keys be stored, and for how long? These questions may add significantly to the complexity of an encryption project.

Asymmetric encryption (public key/private)

Asymmetric encryption uses two separate keys, each of which has a specific function. A public key encrypts the data, while a private key decrypts the data. The public key can be freely 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 where the decryption occurs must be handled very securely.

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 in 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.

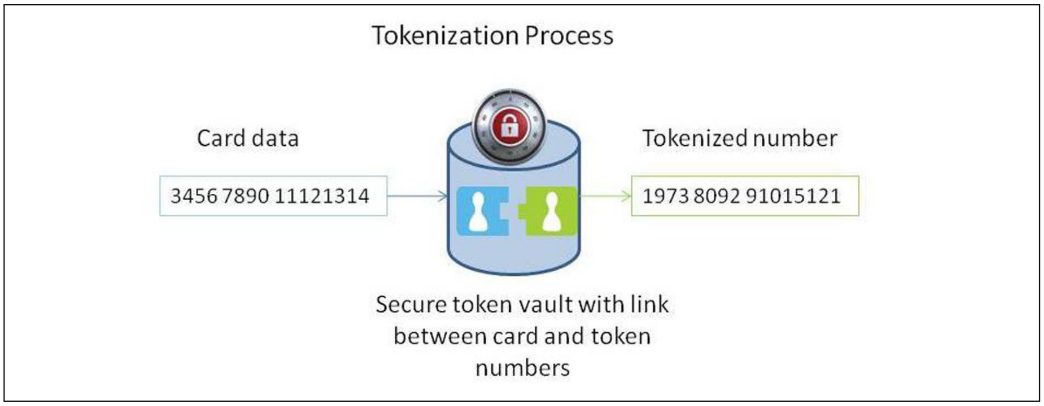
There are models of card readers that have a TRSM inside so that data can be encrypted immediately at the point of capture. Hardware-based encryption offers a higher degree of overall security than software-based encryption because it prevents key tampering or theft; it is considered "the best of best practices." 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. Moreover, at this time, many merchants are in a wait-and-see mode with Chip and PIN technology potentially on the horizon, and thus are hesitant to acquire new readers in the near term.

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. Many merchants appreciate that adding software-based encryption doesn't require a capital investment in new equipment.

Tokenization

An increasingly popular approach for the protection of sensitive data is the use of data substitution with a token (or alias) as a replacement for a real credit card number. In the process of tokenization, actual cardholder data is used in a payment transaction and, once the transaction is authorized, this very sensitive data is sent to a centralized and highly secure server called a "vault" where it is stored securely. 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, can be used in various auxiliary business applications as a reliable substitute for the real card data.



Why tokenization is important

Tokenization is important for two reasons:

1. it vastly reduces a merchant’s 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.
2. 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 design of the token

In its most general sense, a token is just a random series of characters that are 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 should have 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 customer need not know he is looking at a token number instead.

- → 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

It's possible to use an encryption routine that generates an output value in the same format as the input value (i.e., the cardholder number). This would allow the encrypted number to be used in place of clear text card numbers in back-end ancillary applications without significant disruption of regular business processes.

There are some potential drawbacks for 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 reduction in PCI scope is one of the key values in 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.

Keep in mind that tokenization creates a random number that simply represents a real card number. The token number is not mathematically reversible. Therefore, someone must create the token and store it in a secure reference database ("vault") in order to have a static relationship between the token number and the actual cardholder number.

> *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, which is most common.)

The advantage of a card-based token is that the merchant can aggregate a consumer's transactions over time and start to build a buying history on that consumer—at least at the card level if not at a higher level from a loyalty program. Many merchants want to understand their consumers' buying habits, but today it is too risky to pull together a list of transactions based on a real card number. Card-based tokens enable this process without the risk of exposing sensitive data.

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.)

However, these business processes may be broken if a merchant splits his business across two or more payment processors/token providers and the processors return different token numbers for the same card number. For example, if a merchant submits credit card transactions to one processor and PIN debit transactions to another, there will be different token numbers generated for the same card. To take full advantage of a card-based tokenization model, a merchant needs to have all its business go through a single processor. Another option is for the merchant to be its own provider/keeper of the tokenized numbers. This scenario is covered in more detail below.

In general, card-based tokens are most beneficial in a larger merchant arena where the merchant is keeping data for more than just the purpose of getting paid. The permanent association between a card number and a tokenized number help the merchant perform ancillary business tasks. That said, there is no drawback for smaller merchants who use card-based tokens. In fact, smaller merchants may find that they are leaving the door open to future data usage possibilities by using card-based tokens.

Transaction-based tokens

In the case of transaction-based tokens, 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 token numbers–whether card-based or transaction-base –is the elimination of actual cardholder data from the merchant's environment, thus vastly lowering the risk from a data breach and the time and 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-auth card data. It also relieves the merchant of the burden of managing and securing the token vault.

Conclusion

Encryption and tokenization solve for mutually-exclusive security weaknesses in the payments process. 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.

At this writing, neither technology is fully required as part of PCI DSS, although PCI does require data encryption for card data stored in the merchant environment. Tokenization is being considered for incorporation into a future version of the PCI DSS guidelines. Use of one or both of these technologies today goes beyond the minimum security requirements of PCI, but there are sufficient benefits to merchants to warrant implementation of both end-to-end encryption and tokenization. Using the two technologies together is a very strong way to secure data.

Recommended Reading

To learn more about end-to-end encryption and tokenization and how they can help merchants secure their cardholder data and reduce their PCI liability, please see:

→ First Data white paper: Implementing Tokenization is Simpler Than You Think

→ First Data white paper: Data Encryption and Tokenization: An Innovative One-Two Punch to Increase Data Security and Reduce the Challenges of PCI DSS Compliance

→ First Data white paper: Where Security Fits in the Payments Processing Chain

→ RSA's Speaking of Security Blog: Encryption

→ RSA's Speaking of Security Blog: Tokenization

The Global Leader in Electronic Commerce

First Data powers the global economy by making it easy, fast and secure for people and businesses around the world to buy goods and services using virtually any form of payment. Serving millions of merchant locations and thousands of card issuers, we have the expertise and insight to help you accelerate your business. Put our intelligence to work for you.

About the Authors

Tim Horton is vice president of merchant product management and oversees First Data's TransArmor® secure payment processing offering and other security-related products. Horton joined First Data in 1995, and has since held a variety of leadership roles of increasing responsibility. Prior to his current position, he was in Corporate Strategy working on large company initiatives and with third-parties on potential partnerships. He also served as vice president of Product and Business Development for the DDA Payment Services Organization, now known as CONNECTPAY, an innovative ACH Payment Card solution. Before that, was he was director of Strategy and Market Development and helped create a new retail market organization. Horton holds a master's degree in business administration from the University of Nebraska at Omaha.

Robert McMillon is director of solution development for the joint partnership between First Data and RSA. McMillon has extensive experience in security strategy, corporate governance, business process transformation and product development. He helped start the first independent security team in the Southeast, founded the managed security services company that eventually became Verizon's managed security business, and has brought many successful security products to market.

For more information, contact your First Data Representative or visit firstdata.com