



Guide to Data Field Encryption



Contents

Introduction	2
Common Concepts and Glossary	3
Encryption	3
Data Field Encryption	3
Cryptography	3
Keys and Key Management	5
Secure Cryptographic Device	7
Considerations for Data Field Encryption	8
Business Considerations for Data Field Encryption Systems	8
Data Elements to be Encrypted	8
Encryption with Tokenization	11
Other Security Considerations	11
Data Field Encryption Examples	12
Frequently Asked Questions	14
Is it only for Magnetic Stripe Transactions?	14
What if the Data Needs to be Used Again?	14
Data Field Encryption Summary	15
Appendix A—Visa Best Practices for Data Field Encryption, Version 1.0	16

Introduction

This guide is a supplement to Data Field Encryption Best Practices (Visa, 5 October 2009). It outlines how merchants, processors and other entities that transmit cardholder data can meet Visa's requirements and includes examples of how data field encryption may be implemented.



Common Concepts and Glossary

Encryption

Encryption is a reversible transformation of data, where the transformation is dependent only on the encryption algorithm and the value of a 'key'. This key is a large number, especially when using industry-standard algorithms, and therefore it is impossible to count through all possible key values. For the purposes of this guide, the use of encryption to protect cardholder data as it is stored, processed and/or transmitted during payment transactions is referred to as data field encryption.

The security of an encryption operation ultimately depends upon the secrecy of this key. Industry-standard algorithms are well known and well tested; it is only the secrecy of the key that prevents anyone from decrypting the encrypted data.

Although this guide does not intend to provide details on the nature of cryptography, it is important to understand that there are two different types of cryptography which are commonly used in data field encryption – symmetric cryptography and asymmetric cryptography.

Data Field Encryption

Data field encryption (sometimes called end to end encryption or E2EE), is a way to reduce the risk of exposing sensitive card data by using industry-standard encryption to render the data unreadable at the point of entry. Only entities that have access to the decryption key will be able to read sensitive card data.

Data field encryption ensures that cardholder data does not appear unencrypted (i.e. in cleartext) outside of a Secure Cryptographic Device (or SCD). This reduces the possibility for access to this data by unauthorized parties, such as criminals, increasing the security of the merchant's payment card acceptance and processing systems and helping to prevent costly card data compromises which reduce overall cardholder confidence and trust in electronic payments, and can hurt the compromised entity's brand reputation.

Cryptography

Symmetric Cryptography

Symmetric cryptography is where the key used for encryption is exactly the same as the key that is used for decryption. Examples are the algorithms Triple DES and Advanced Encryption Standard (AES). This type of cryptography is like a traditional mechanical lock and key – the encryption algorithm is the lock, and the cryptographic key can both lock and unlock the data, as illustrated in Figure 1.

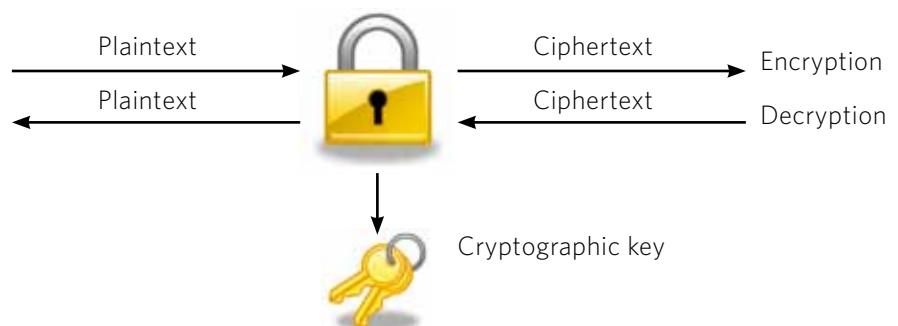


Figure 1: Symmetric cryptography

Asymmetric Cryptography

Asymmetric cryptography differs in that it allows for more than one key value – one key for encryption and one for decryption. It can be likened to an electronic lock on a safe (rather than a mechanical lock), where the action of locking the safe is different from unlocking the safe. Locking the safe (which is like encryption) involves pressing the 'LOCK' button and can be done by anyone.

Unlocking the safe requires entry of a passcode that is known only to the person who owns the safe. Once the safe has been locked, the contents are secured and can only be accessed by someone who knows the passcode.

The two keys used in asymmetric encryption are often referred to as the 'public' key and the 'private' key. The public key, used for encryption, is like the LOCK button on the safe, and can be known by anyone. Knowledge of this key does not enable decryption of data, just as knowledge of the 'LOCK' button does not enable the safe to be opened. The private key, used for decryption, is like the passcode and must be kept secret.

Asymmetric cryptography is much slower than symmetric cryptography; therefore, it is usually not used to encrypt data directly. It is most often used for the secure transfer and/or storage of a symmetric key, which in turn is used as a data encryption key.

Examples of asymmetric encryption are the algorithms RSA and Elliptical Curve Cryptography (ECC).

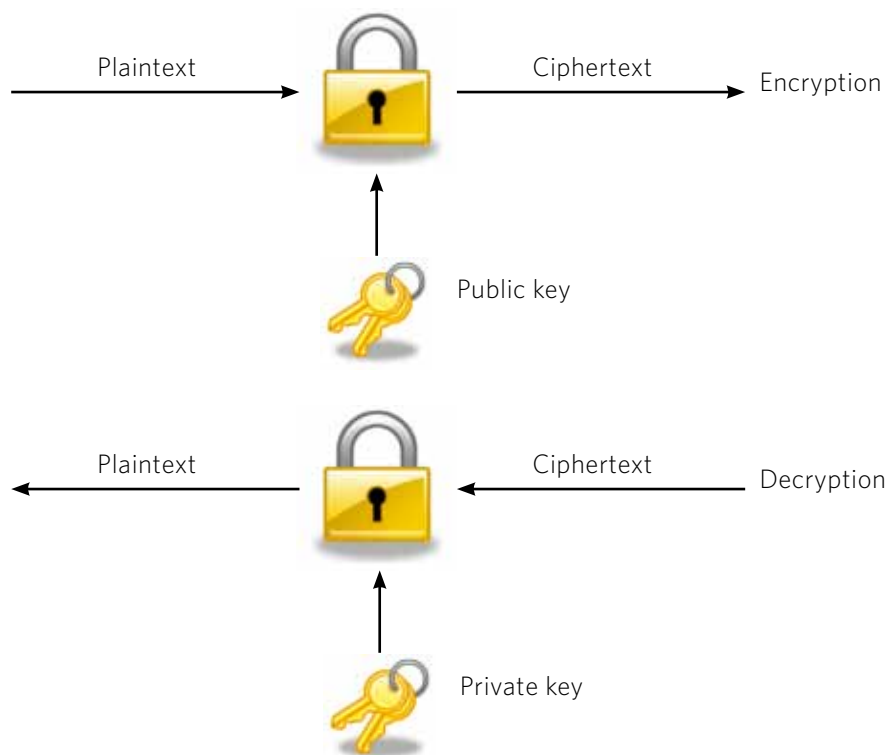


Figure 2: Asymmetric cryptography

Keys and Key Management

Acceptable Encryption Algorithms and Key Sizes

There are many different types of encryption, but only some of these can be used to provide acceptable security for data field encryption. Table 1 provides a list of these acceptable algorithms, and the length of the key which must be used for each of these algorithms.

Encryption Algorithm	Minimum Acceptable Key Length
RSA	2048 bits ¹
ECC	224 bits ²
Triple DES	112 bits ³
AES	128 bits

Table 1: Acceptable algorithms and acceptable key lengths

¹ Refers to the size of the modulus used in the RSA calculation

² Refers to the minimum order of the base point on the elliptic curve

³ Includes parity bits (effective size of minimum key is 112 bits)

Other algorithms, or even the algorithms referenced above when used with smaller key sizes, are not acceptable for use with data field encryption.

The use of an acceptable encryption algorithm and an acceptable key length is not in itself sufficient to meet the requirements for data field encryption. Other critical factors, such as key management and the actual mode of use of the cryptographic algorithms, contribute to the security of data field encryption systems.

Key Management

Key management concerns the entire life cycle of the cryptographic key, from its creation, through its distribution and installation, its use and management, and finally its destruction. As the security of any cryptographic operation entirely depends upon the secrecy of the cryptographic key which is used, it is vital that this key is properly secured, and knowledge of the actual value of the key is prevented. It is essential that any cryptographic key is:

- random
- of a size suitable for the algorithm being used (refer to Table 1)
- managed under dual control at all times
- only ever used for a single purpose

The above requirements exist for all keys whether symmetric (e.g. Triple DES, AES) keys or asymmetric (e.g. RSA, ECC) keys. Additionally, symmetric and private asymmetric keys (but not public asymmetric keys) must be:

- unique to each device (although the same key value may also be present at the point where the data protected by the data field encryption process is decrypted)
- managed using the principles of split knowledge

The exact meaning of each of each of the items is provided below.

Random Keys

As noted previously, cryptographic keys provide security to cryptographic algorithms by the fact that they are very large numbers which are therefore hard to predict. These keys must be generated randomly - there are many examples of attacks on cryptographic systems where the underlying algorithm used is secure, but the attack has exploited the fact that the key generation system generates values that are not truly random.

For example, if a key is generated from a pass phrase, the difficulty of guessing the value of that key is reduced to the difficulty of guessing the pass phrase on which it is based. This is usually easier than trying all possible key values.

Therefore, it is vital that the key is generated by a SCD which has been independently assessed to confirm that the random numbers it produces are sufficiently random for the generation of secure cryptographic keys.

Dual Control

Known as dual control, the installation of any key into a SCD should require the participation of at least two authorized individuals. This prevents any one person from replacing a valid key with one that allows them to compromise the encryption system, or from determining the cleartext value of a secret symmetric or private asymmetric key.

Single Purpose Keys

Cryptographic keys must be used only for their sole intended purpose. It can be tempting to use a cryptographic key for different purposes to reduce the overhead of managing different keys, but this often introduces potential security problems.

For example, a cryptographic key used for data field encryption must not be used to provide or verify authentication values across data transmissions, or as a method of securing firmware updates to the SCD. Doing so would compromise the security of the data field encryption device, and therefore remove the benefit of using such a device. Similarly, a device must not use the keys used for PIN encryption to encrypt other cardholder data. A PIN key must only ever be used for PIN encryption.

Unique Keys

It is also essential that secret symmetric and private asymmetric keys are always unique to each device. For symmetric cryptography, it is usually necessary to have a pair of devices (i.e. one at each 'end' of the encrypted connection) sharing the same key - one in the SCD that performs the encryption, and one in the SCD/HSM that performs the decryption. No other devices should share this key.

This prevents the compromise of a single device from facilitating the compromise of a large number of other devices. A number of methods enable the extraction of a cryptographic key from even an approved SCD - however, such attacks are usually either destructive (therefore preventing that SCD from being used after such an attack), or very costly. If the key to be extracted exists only in one SCD (i.e. the SCD under attack), such attacks are generally not commercially viable for criminals - the attack would cost more than the potential profit. However, if the key obtained is used in other devices, then the attack may become worthwhile.

For example, if an attack can be performed for \$100,000, but only returns \$10,000 per device, it is not a useful attack if the keys are unique per device. However, if there are more than ten devices sharing this same key, then each device can be compromised for \$10,000, for a sum total in excess of \$100,000 - therefore making the attack economically feasible.

Split Knowledge

Split knowledge requires that no one person should know, or be able to determine, any part (not even a single bit) of a secret symmetric or private asymmetric key. This can be implemented in a number of ways, but it is common for a device to allow for the input of cryptographic keys as two or more separate values, the same length as the final key, which are mathematically combined within the device so that the knowledge of any one of the input values does not provide any information on the actual value of the key that is created.

It should be noted that split knowledge does not mean taking a key and cutting it in half – this would allow each person to know half of the value of the cleartext key. Split knowledge requires that no one person knows any information about the key at all. Therefore, it is essential that any scheme that is used to provide split knowledge does not provide cleartext values of the key to any custodian.

Secure Cryptographic Device

A SCD is a device that provides physical and logical security to the cryptographic keys and operations for which it is used. Examples of a SCD include a PIN Entry Device (PED), a Hardware Security Module (HSM), or a physically secure encrypting card read head. Several international security evaluation standards apply to the different types of SCDs, and it is important that devices used with data field encryption have been assessed to such standards. A specific standard for devices that provide secure data field encryption does not exist at this point in time.

Examples of existing standards for SCDs and lists of devices evaluated to these standards can be found at:

- PCI PIN Transaction Security (PCI PTS) Standard
https://www.pcisecuritystandards.org/security_standards/ped/pedapprovalist.html
- FIPS 140-1 and FIPS140-2
<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm>
(Data field encryption devices should be approved to FIPS 140 level 3 or higher)

Note that the presence of a device on the list above (or absence from) should not be used as a sole indicator as to its suitability for data field encryption.

Considerations for data field encryption

Business Considerations for Data Field Encryption Systems

Although the use of data field encryption provides many benefits in regard to the security of cardholder data, there may be operational and business related impacts when implementing such systems. For example, there is currently no global standard for the implementation of data field encryption, and this means that a system provided by one vendor may not work with the equipment provided by another vendor.

Similarly, it is necessary to consider where the data will be decrypted when implementing a data field encryption system. If data is to be encrypted from the point of entry, all the way to the acquirer host, it will be necessary for the acquirer host systems to be integrated with the systems used for this data field encryption.

How other systems which normally handle the cleartext cardholder data will work with encrypted data also needs to be considered. Regardless of whether vendors of data field encryption systems provide the encrypted data in a form that appears similar to unencrypted data, the use of encrypted data may have operational impacts.

For example, if a data field encryption system is encrypting all track data at the point of entry (i.e. using an encrypting read head), this will prevent systems from determining both the type of card and the brand of the card if the card has a chip or is a magnetic stripe only card. This may prevent the correct operation of chip and PIN-based systems, or the correct routing of transaction data.

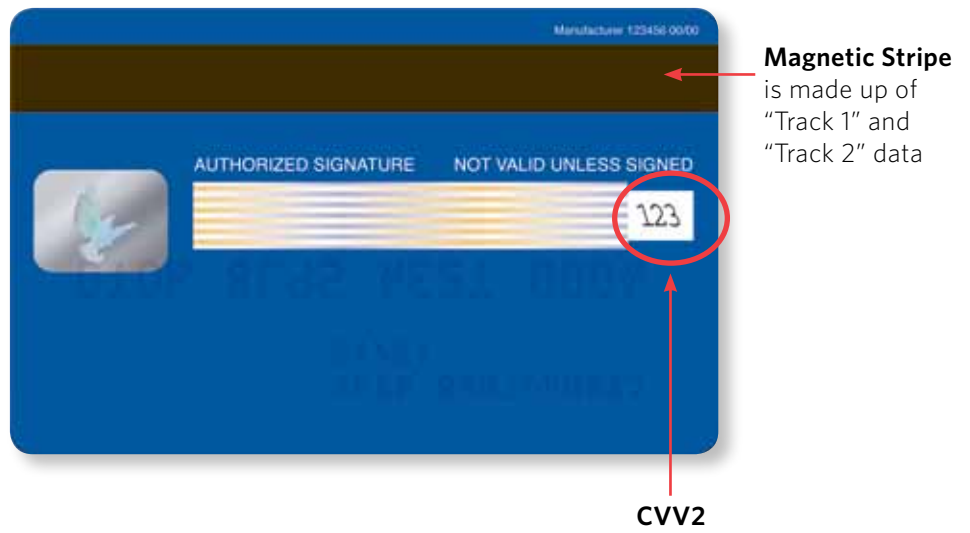
Similarly, there may be non-payment cards where data field encryption is not required, such as merchant loyalty cards or merchant identification cards.

Therefore, it is important that operation of current systems is well understood when implementing a data field encryption system to ensure that the most appropriate solution is chosen. Businesses must determine what risk is presented by the types of card data and different presentment methods that they accept in their environment, and ensure that any data field encryption system used correctly addresses this risk.

Data Elements to be Encrypted

It is important that cardholder data which can be used to authorize a transaction is encrypted. This data includes:

- The account number (sometimes called the primary account number or PAN). This is often printed on the front of the card, and is also contained on the magnetic stripe and the chip if it is a chip card (either contact or contactless).
- The CVV2 number. This is used to verify the authenticity of the card, often during card-not-present transactions and sometimes card-present transactions. It is printed on the back of the card directly to the right of the signature panel.
- The CVV number. This is used to verify the authenticity of the card during card-present transactions, and is contained on the magnetic strip and the chip for chip cards (either contact or contactless).
- The PVV number. This is an optional value that the card issuer can provide to allow Visa to verify a cardholder's PIN on his/her behalf. This is contained on the magnetic strip, and may also be contained on the chip of a chip card.
- Other discretionary data which appears on the magnetic stripe, after the card service code.



The card account number, plus a three-digit Card Verification Value 2 (CVV2) is printed on the signature panel.

Figure 3: Back of Visa Card



Figure 4: Track 1 Data

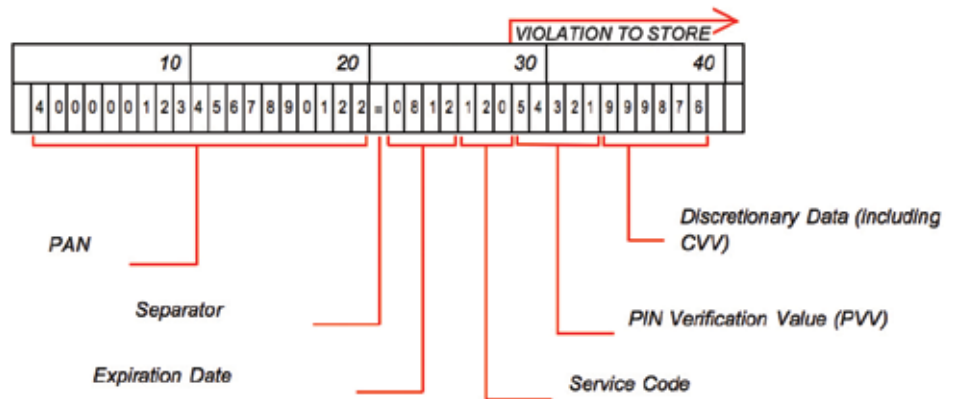


Figure 5: Track 2 Data

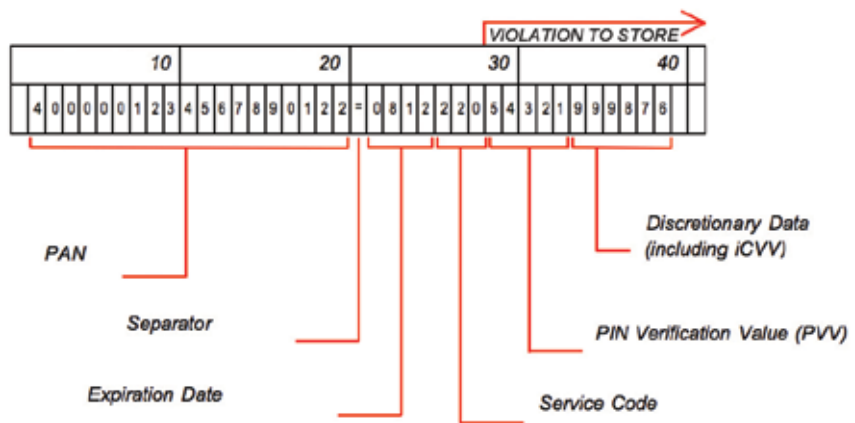


Figure 6: Magnetic Strip Image (Chip)

It is often necessary to provide some cardholder data to the merchant or payment system in cleartext - that is, without being protected using encryption. This may include the expiry date, the card service code and/or the cardholder name. These values are often used by systems between the merchant and the processor to perform basic payment validation, receipt formatting, or to check if the card is of a specific type (e.g. if the card has a chip, or is a magnetic stripe only card). Similarly, the first six and/or last four digits of the cardholder's primary account number (truncated PAN) may be required for routing of the transaction, or for validation of the cardholder account used in that transaction at some later time.

However, at all times, sensitive authentication data, and any digits of the cardholder PAN between the first six and last four digits, must be encrypted.

Encryption with Tokenization

At the start of this guide, encryption was defined as a reversible transformation of data, where the transformation depends only on the encryption algorithm and the value of a key. Tokenization is where the card data is replaced with a unique value which can only be correlated to the value of the original card data by the tokenization system itself.

In this way, tokenization and encryption provide similar services – both are reversible and both protect the original cardholder data with a value that can only be restored with access and control over the system that performed the encryption/tokenization operation. Tokenization systems must be secured in similar ways to encryption systems, but the tokenization system must protect the information that allows for the correlation between the token and the original card data (e.g. a database of values), whereas for an encryption system, it is the encryption key which must be protected.

Visa best practices require that any tokenization system produce the transaction token using methods that cannot be correlated to, or reversed to produce, the actual input PAN value (Best Practice Item 14). Examples of such methods include the use of cryptographic digests, or ‘hash’ algorithms, across the PAN data appended to a random value called a ‘salt’. This ‘salt’ value ensures that a malicious party will not be able to ‘pre-compute’ a large list of all possible PAN value outputs – as they will not know the actual value of the ‘salt’ beforehand.

Other Security Considerations

In addition to meeting all of the encryption key management requirements, a data field encryption system should also ensure:

- the same data input is encrypted to a different data output each time
This prevents ‘dictionary’ attacks, whereby an attacker would generate a large dictionary-like listing of possible input values correlated with their encrypted output value. With such a list, the attacker needs only to look up any encrypted output to determine the actual input value, thereby bypassing the security of the encryption system.
- any decryption services provided cannot be misused by an attacker to decrypt the cardholder data
This ensures the system cannot be misused to undo the encryption it has previously provided to any cardholder data. If an attacker compromises the facility in which the data field encryption system is implemented, the attacker remains unable to compromise any data which is protected by encryption through the data field encryption system.
- the way that the encryption algorithm is used does not create vulnerabilities
Often it is the implementation of a particular encryption system, not the algorithm itself, where weaknesses lie. For example, standards bodies such as the American National Institute of Standards and Technology (NIST) provide documents covering the acceptable ‘modes of operation’ for particular algorithms; a poor choice of mode may result in the encryption system not conveying security or confidentiality to the data being encrypted.
- To assist with compatibility with existing payment systems, some vendors are creating new modes of operation for the encryption process. Visa requires that any mode of operation used for data field encryption has been extensively reviewed, and has been ratified by an independent body such as NIST.

Data Field Encryption Examples

This section contains examples of how data field encryption may be implemented.

Encryption within a PED

In this example, a card is entered into a PCI PTS-certified PED. The data is encrypted by the PED for transmission to the acquiring host, where it is decrypted by a FIPS 140-2 or PCI PTS-certified HSM. This example shows a number of PEDs within a store – each having its own unique encryption key. The HSM at the acquirer stores the encryption keys for all of the PEDs.

The PEDs cannot decrypt data once it is encrypted, but they do output the expiry date, service code, and the first six and last four digits of the PAN in the clear (for use by the attached POS system). Entry of cardholder account numbers manually (e.g. for damaged cards) is performed on the PED keypad and card data is never entered into any other system. The PEDs accept and encrypt data from both magnetic stripe and chip-embedded cards.

In this example, sensitive cardholder data never appears unencrypted in any of the merchant systems.



Figure 7: Encryption within a PED

Encryption with a Card Reader

In this example, the card is swiped through a magnetic stripe card reader, using security components (certified to FIPS140-2) that encrypt the cardholder data before passing it to the POS system for transmission to the acquiring host, where it is decrypted by a FIPS 140-2 or PCI PTS-certified HSM. This example shows a number of encrypting card readers within a store – each having its own unique encryption key. The HSM at the acquirer stores the encryption keys for all of the PEDs.

The encrypting card readers cannot decrypt data once it is encrypted, but they do output the expiry date, service code, and the first six and last four digits of the PAN in the clear (for use by the attached POS system).

The card reader only provides a magnetic read interface, therefore, manually entered data (e.g. for a damaged card) or data presented on a chip card cannot be encrypted. Encryption of this data from the merchant to the host is provided by the POS system, using an SSL connection, or some other type of encrypted connection.

In this example, only the magnetic stripe data is encrypted within the store – data from chip (both contact and contactless) and manual entry transactions remains in plaintext in the store environment. Therefore, the risk for exposure of this data remains higher than that for the data which is encrypted directly at the swipe.

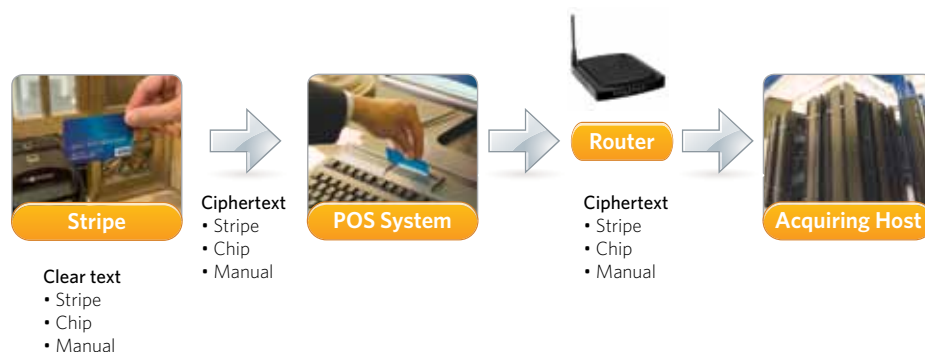


Figure 8: Encryption with a card reader

Frequently Asked Question

Is it only for Magnetic Stripe Transactions?

Cardholder data may be presented in a number of ways – either on the cardholder magnetic stripe, on the chip contained within a cardholder smartcard that is used for contact or contactless transactions, or printed on the front or rear faces of the card. For data field encryption to provide the maximum benefit, it is essential that all cardholder data, regardless of how it is entered into the merchant environment, is encrypted.

Cardholder data can be presented in many different ways – for card-present transactions, cardholder data may be obtained from the magnetic stripe, through a contact or contactless chip interface, or by entering the details into the payment system through key entry. For card-not-present transactions, the cardholder provides their card information by entering it into a web browser, reading it over the phone, or manually recording it electronically or on paper.

Therefore, data field encryption does not only apply to transactions which use the cardholder's magnetic stripe – although these types of systems may form part of an overall data field encryption system. It is important that an overall data field encryption solution covers all methods by which the cardholder may provide their payment data, and all methods by which this data may be used in the merchant's or processor's environment.

For example, if a cardholder presents a card with a damaged magnetic stripe, the merchant may enter the cardholder data into a point-of-sale (POS) system through a standard keyboard – bypassing the secure device which would normally be used to provide the encryption to the cardholder data. It may be that any manual entry is strictly performed on the keypad of the secure device rather than the keyboard of the cash register, thereby ensuring that the protection of this data with the end-to-end encryption system is maintained.

What if the Data Needs to be Used Again?

Any possible system should be implemented so that cardholder data is not reused. For example, it may be possible to perform authorization transactions with the cardholder data, and finalize the transaction only with a reference number from the authorization response. For tracking and auditing purposes it may be sufficient to keep only the first six and last four digits of the cardholder account number, thereby removing the need to retain, or allow for the recovery of, the encrypted cardholder data.

If a system must allow for the recovery or reuse of an entered cardholder account number, then the data field encryption system must ensure that it is not possible to use this function to obtain the PAN details in cleartext. It should also be noted that in all circumstances it is forbidden to allow for the storage or recovery of sensitive cardholder data (i.e. full magnetic stripe data, CVV, CVV2, PIN data), irrespective of any encryption used.

Another option is to provide a token for the card data, which can be used instead of the original card data. This system is referred to as 'tokenization' and is discussed briefly above under the heading "Encryption with tokenization".

Data Field Encryption Summary

Data field encryption is a powerful tool that can be used, in conjunction with other methods, to protect cardholder data. However, the details of how the data field encryption system operates must be understood to ensure not only that sufficient protection is provided to the cardholder data, but also to understand any operational impacts such a system may have to your business. Be sure to understand these key issues when evaluating a data field encryption system:

- **Is a strong encryption algorithm being used (RSA, ECC, Triple DES, or AES)?**
 - What mode of operation is being used? Has this been ratified by an independent standards body such as NIST?
- **How are the encryption keys loaded, stored and managed?**
 - Are keys managed within a secure cryptographic device?
 - How are encryption keys loaded? Are they unique per device, or the same across all devices?
 - Are the principles of dual control and split knowledge maintained?
- **What types of transactions and cardholder data entry are supported by this system?**
 - Does it support all methods of cardholder data presentation used in your environment (magnetic stripe, contact and/or contactless smartcard, and manual entry)?
- **Do any existing systems or connections require some part of the cardholder data to remain unencrypted?**
 - Is this supported by this data field encryption system?
- **Can the encryption be switched off and, if so, what controls exist around this function?**
 - What happens when there is an error?
Are any 'debugging' features included in the data field encryption system?



Appendix A—Visa Best Practices for Data Field Encryption, Version 1.0

Point of Sale to Acquirer Domain

Security Goals

1. Limit cleartext availability of cardholder data and sensitive authentication data to the point of encryption and the point of decryption.
2. Use robust key management solutions consistent with international and/or regional standards.
3. Use key-lengths and cryptographic algorithms consistent with international and/or regional standards.
4. Protect devices used to perform cryptographic operations against physical/logical compromises.
5. For business processes, use an alternate account or transaction identifier that requires the primary account number to be utilized after authorization, such as processing of recurring payments, customer loyalty programs or fraud management.

Environment

- Visa Best Practices for Data Field Encryption pertain to systems used to acquire payment transactions in physical locations such as:
 - Payment terminals
 - Merchant point-of-sale systems
 - In-store controllers
 - Corporate transaction aggregation systems
 - Payment gateways
 - Core processing systems
 - Acquirers or acquirer-processors
- Cardholder data includes Primary Account Number (PAN), cardholder name and expiration date.
- Sensitive authentication data includes but is not limited to full contents of the magnetic-stripe data, track 1, track 2, Card Verification Value (CVV), CVV2, PIN Verification Value (PVV), and PIN/PIN block. Sensitive authentication data shall not be used for any purpose other than payment authorization.

Best Practices

The following are best practices for data field encryption to protect cardholder data and sensitive authentication data:

Security Goal	Best Practice
<p>Limit cleartext availability of cardholder data and sensitive authentication data to the point of encryption and the point of decryption.</p>	<ol style="list-style-type: none"> 1. Cleartext cardholder data and sensitive authentication data shall only be available at the point of encryption and at the point of decryption. 2. All cardholder data and sensitive authentication data shall be encrypted using only ANSI X9 or ISO approved encryption algorithms (e.g. AES, TDES). 3. All cardholder data and sensitive authentication data shall be encrypted with the following exceptions: <ul style="list-style-type: none"> ▪ The first six digits of the PAN may be left in the clear for routing purposes in authorization processing. ▪ The first six and last four digits of the PAN may be displayed by the payment terminal and/or printed on the transaction receipt, in settlement reports, used for selection of account on file, etc. (This does not supersede stricter laws or regulations in place for displays of cardholder data.) 4. Sensitive authentication data must not be stored after authorization even if encrypted (per PCI DSS).
<p>Use robust key management solutions consistent with international and/or regional standards.</p>	<ol style="list-style-type: none"> 5. Keys shall be managed per ANS X9.24 (all parts) /ISO 11568 (all parts) or its equivalent within Secure Cryptographic Devices (SCD) such as a PED, HSM, etc., as defined in ANS X9.97 (all parts) /ISO 13491 (all parts) or its equivalent. 6. All keys and key components shall be generated using an approved random or pseudo-random process such as NIST SP 800-22. 7. Documentation describing the set-up and operation of the key management solution must be made available upon request for evaluation purposes. 8. Keys shall be conveyed or transmitted in a secure manner. For example, the key distribution method described in X9/TR-34 <i>Interoperable Method for Distribution of Symmetric Keys using Asymmetric Techniques, Part 1-Using Factoring-Based Public Key Cryptography Unilateral Key Transport</i> (to be published) or equivalent should be used. <ul style="list-style-type: none"> ▪ If remote key distribution is used, mutual authentication of the sending and receiving devices shall be performed. 9. Keys used in the data field encryption process: <ul style="list-style-type: none"> ▪ Must be unique per device. ▪ Must only be used to encrypt cardholder data and sensitive authentication data and cannot be used for any other purpose. ▪ Keys used for PIN encryption must never be used for data field encryption (per PCI PIN Security Requirements).

Security Goal	Best Practice												
<p>Use key-lengths and cryptographic algorithms consistent with international and/or regional standards.</p>	<p>10. Encryption keys shall have strength of at least 112 equivalent bit strength. The following table summarizes equivalent bit strengths for commonly used approved algorithms:</p> <table border="1" data-bbox="916 434 1230 676"> <thead> <tr> <th>Algorithm</th> <th>Bit Length</th> </tr> </thead> <tbody> <tr> <td>TDES</td> <td>112¹</td> </tr> <tr> <td>AES</td> <td>128²</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> </tbody> </table> <p>For details on equivalent bit strengths, see <i>ISO TR-14742 Recommendations on Cryptographic Algorithms and their Use – Technical Report</i> (to be published 2009).</p> <p>11. Any methods used to produce encrypted text of the same length and data type as the original cleartext shall be evaluated by at least one independent security evaluation organization and subjected to a peer review; such methods shall also be implemented following all guidelines of said evaluation and peer review including any recommendations for associated key management.</p>	Algorithm	Bit Length	TDES	112 ¹	AES	128 ²	RSA	2048	ECC	224	SHA	224
Algorithm	Bit Length												
TDES	112 ¹												
AES	128 ²												
RSA	2048												
ECC	224												
SHA	224												
<p>Protect devices used to perform cryptographic operations against physical/logical compromises.</p>	<p>12. Devices used to perform cryptographic operations should undergo independent assessment to ensure that the hardware and software they are using is resilient to attack.</p> <p>13. Symmetric and private keys shall be protected against physical and logical compromise. Public keys shall be protected from substitution and their integrity and authenticity shall be ensured.</p>												
<p>Use an alternate account or transaction identifier for business processes that requires the primary account number to be utilized after authorization, such as processing of recurring payments, customer loyalty programs or fraud management.</p>	<p>14. If any cardholder data (e.g., the PAN) is needed after authorization, a single-use or multi-use transaction ID or token should be used instead.</p> <ul style="list-style-type: none"> ▪ A single-use transaction ID is preferred. <ul style="list-style-type: none"> - Acceptable methods for producing a single-use transaction ID include hashing of the PAN with a transaction-unique salt value, encrypting the PAN with an approved algorithm using a transaction-unique key, or equivalent. The single-use transaction ID can be produced by other methods provided that the resulting reference data is unique per transaction and the original cardholder data (e.g. the PAN) cannot be reproduced. ▪ A multi-use transaction ID may be used if there is the need to maintain correlation (of the account) across multiple transactions. <ul style="list-style-type: none"> - Acceptable methods for producing a multi-use transaction ID include hashing of the cardholder data using a fixed (but unique per merchant) salt value, or equivalent. <p>NOTE: Irrespective of whether the transaction ID is a single use or multi-use, if a salt is used, the salt should be a minimum length of 32-bits and must be kept secret and appropriately protected.</p>												

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 transaction 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.)



