Safetech℠ Encryption and Tokenization Supplemental Guide

Published: November 13, 2018

Version: 3.2.0.0

Disclosure Statement

This document contains information which is confidential and proprietary to Paymentech, LLC, Chase Paymentech Solutions and Chase Paymentech Europe Limited (collectively “Merchant Services”) and may only be used in relation to services and products provided by Merchant Services. Merchant Services is a marketing name for the Merchant Acquiring and Payment Processing business of JPMorgan Chase & Co. (“JPMorgan Chase”). This document contain information that is confidential and/or proprietary to Merchant Services and/or JPMorgan Chase and may not be copied, used or published, in whole or in part, for any purpose other than as expressly authorized by Merchant Services. Merchant Services makes no representations as to the legal, regulatory, tax or accounting implications or suitability for any particular business of the matters referred to in this document.

All trademarks, trade names and service marks appearing herein are the property of their respective owners.

Contents

What's New or Changed in Version 3.2.0.0

The following updates have been made to this version of the guide since version 3.1.5:

- Completed annual recertification of document content (relevance and technical accuracy)
- Updated copyright and publication dates.
- Updated version number to new 4-digit requirement.
- Replaced Change Log with this section.
- Removed highlighting of text throughout document.

Overview

Chase's Safetech℠ Encryption is a point-to-point encryption technology that protects the primary account number (PAN) on a payment card from the moment it is captured at the point of sale until it reaches the processor, with no need for the merchant or vendor to process, transmit, or store unprotected card account data. This process is referred to as *PAN Encryption* or *End-to-End Encryption*.

Chase uses format-preserving encryption (FPE) technology to encrypt the card, PAN and discretionary data before it enters the merchant's payment system and keeps it secure until it reaches Chase's data center. This requires an encryption application to be installed on the merchant's payment terminal and also requires Chase data centers to decrypt the card data for subsequent processing.

The Safetech Encryption solution encrypts the available card data as follows:

- The magnetic card track data is encrypted on all swiped card transactions.
- The PAN is encrypted on each manually keyed transaction.
- Encryption also occurs on Contactless (tap) and EMV Chip transactions.

There is no distinguishable difference between clear text data and the encrypted data; therefore, data validity checks performed at the point-of-sale (POS) level (such as MOD 10 or LRC) should not be impacted.

Benefits

- Secures cardholder data for card-present transactions, including those that are swiped and manually keyed
- May lower Payment Card Industry Data Security Standard (PCI DSS) compliance costs
- Integrates with most POS systems, usually with minimal POS system impact and low disruption
- Potentially reduces the risk of harm if/when a system breach occurs, since any encrypted data obtained is useless without decryption keys
- Card format within Verifone is preserved and the first six digits and last four digits are unencrypted for proper routing and reporting.

Industries

Safetech Encryption is applicable to the card-present, retail industry only at this time. Merchants requiring tip or other authorization adjustments (as is common in the restaurant, lodging or auto rental industries) are not currently supported. Many of these industries are being evaluated for support in future enhancements and product releases.

Merchant Environment and Setup Requirements

The following are environment and setup requirements that the merchant must follow in order to use Safetech Encryption:

- Retail card-present merchant environment
- Host Capture (HCS) Processing (PNS-ISO or UTF)
- Terminal Capture (TCS) Processing (PNS-ISO or UTF)
- Merchants must settle transactions via one of the Chase Merchant Services back-end platforms.
- Merchant provided list of third-party BIN ranges to be excluded from encryption (i.e. gift, loyalty)
- Have appropriate device hardware
    - IPBC: Mx Series
    - TGK: Vx Series
    - MagTek supporting MagneSafe
    - Ingenico Telium II supporting On-Guard
- For IPBC: Have a direct network connection from the PIN pad to Chase systems (MPLS, VPN, or similar)

- For TGK: Integration requires a certification to process E-Parms data in transaction messages and must also process key management transactions using the command BIN of 111111.
- For Derived Key: PIN pad integration for encrypted transactions with host requires certification

BIN Exclusions

Once encryption is enabled, all cards are encrypted, regardless of issuer. Cards that meet ISO standards for magnetic track data (swiped), EMVco chip cards (inserted), Contactless (tapped), NFC (mobile), as well as card data that is manually keyed is encrypted before it is passed to the payment application for processing.

In some scenarios, a card may not be encrypted. These include:

- The card track is not in an ISO standard format. Some debit, gift card, and loyalty issuers may not be following the standard.
- The card is expired.
- The card was excluded from encryption within the BIN File.

Merchant Considerations:

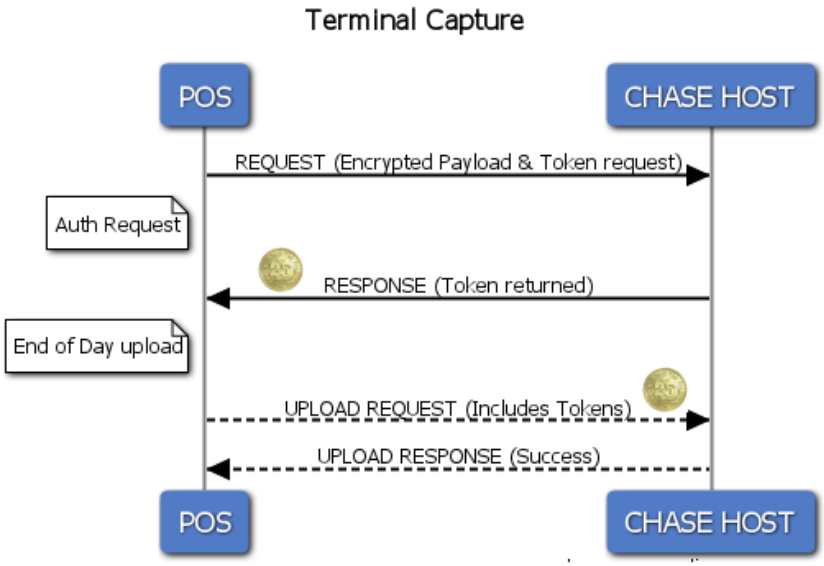
- Are any cards being accepted that do not process through Chase?
- Even though the merchant accepts non-ISO format gift/loyalty cards, is there a chance the cards are manually entered?

Updating the BIN Exclusions

- IPBC devices can be updated by creating a new BIN exclusion file, then assigning update jobs on VeriShield Key Management (VKM) to the appropriate devices, merchants, or domains. The devices connect to the Chase Decryption server during the Kmailman check (every 15 minutes) and then process the new BIN exclusions.
- Verifone Terminal Generated Keys (TGK) and Verifone IP Back Channel (IPBC) standalone devices can be updated by creating a new BIN exclusion file, having it signed by Chase and then downloaded from the Chase VeriCentre.
- Verifone Terminal Generated Keys (TGK) and Verifone IPBC integrated devices can be updated by creating a new BIN exclusion file, having it signed by Chase and pushed from the POS system to the PIN pad.
- Verifone Derived Key integrated devices can be updated by creating a new BIN exclusion file, having it signed and then pushed from the POS system to the device.
- Ingenico devices are loaded with a generic BIN exclusion list during injection.
- Magtek devices do not use a BIN exclusion file.

Terminal Capture Considerations

In the Terminal Capture environment, only Sale/Auth Only transactions and reversals are performed online to the host. Return and Prior Sale transactions, as well as transaction adjustments, occur offline at the POS. In order to receive payment, the merchant must perform a Batch Release/Upload (deposit) to move the transactions from the POS device to the host for settlement. When end-to-end encryption is enabled, the encrypted PAN is included within a 300-byte encrypted payload for each transaction. Due to space considerations, the 300-byte payload should not be stored by the POS. The token takes the place of the encrypted payload. All Terminal Capture E2EE merchants must participate in Tokenization (excluding Chase Mobile Checkout (CMC) and Orbital Gateway merchants).



Terminal Capture Requirements

1. TCS Merchants using encryption and tokenization must perform a token-only request prior to processing and transmitting offline transactions such as returns and prior sales in a TCS batch file.
2. A Safetech token cannot be requested in TCS Upload and TCS Batch (request will be ignored).
   If decryption fails during TCS upload, a detail error is logged for the transaction. If five or more transactions fail, the batch is suspended.
3. As with encryption, there is a five transaction threshold for de-tokenization errors. If five or more errors occur, the batch is suspended.

For transactions where a token cannot be returned in the response of a transaction request (setup issue or token server issue) the Chase card-present platform still attempts the transaction request, but responds with a token error in the T4. Merchants should perform a token-only request using the original account number.

Token-Only Request

The TCS Batch specification does not support encrypted transaction payloads; therefore, any offline prior authorizations, offline refunds, or store-and-forward transactions must obtain a token prior to submitting the batch. For this reason, TCS merchants using encryption and tokenization must perform a Safetech token-only request prior to processing and transmitting offline transactions, such as returns and prior sales, in a TCS batch file. The token-only message is supported in both the PNS ISO and UTF message specifications.

Notes:

1. Token-only requests do not authorize the card on the network.
2. The request must be submitted with the following transaction codes:

- o “91” for credit cards
- o “92 for debit cards
- o “93” for EBT cards.

3. The amount must be sent as 0.00.
4. The token-only message tokenizes the account number submitted in the transaction.
5. Merchants integrating to the TCS batch must have a token for all transactions submitted.

(TCS) Offline Processing

The following table shows what offline scenarios and message types are supported for merchants that use the Verifone Encryption Solution. In the case of an offline or store-and-forward transaction where a token cannot be obtained, encrypted payloads may still be supported (see table below). This will requires the merchant to submit the 300-byte payload in the TCS upload message.

*Please note the TCS Batch does NOT support encrypted payloads so a token must always be obtained prior to submission.

Spec	Token upload support	Encrypted offline payload support	Notes
PNS ISO TCS and HCS	Yes	Yes	• Offline transactions require full encrypted track data Bit 35/45 in the 1300 upload message. • All transaction types that do not fall into the offline scenarios should use the token in the upload message.
UTF 1.97 Terminal Capture System	Yes	Yes	• Swiped Verifone offline transactions require ET token (see UTF Token Guide for supported transaction types). • All transaction types that do not fall into the offline scenarios should use the token in the upload message.
TCS Batch	Yes		• Only tokens can be submitted in the batch file. Encrypted payloads are not supported

Verifone Derived Keys

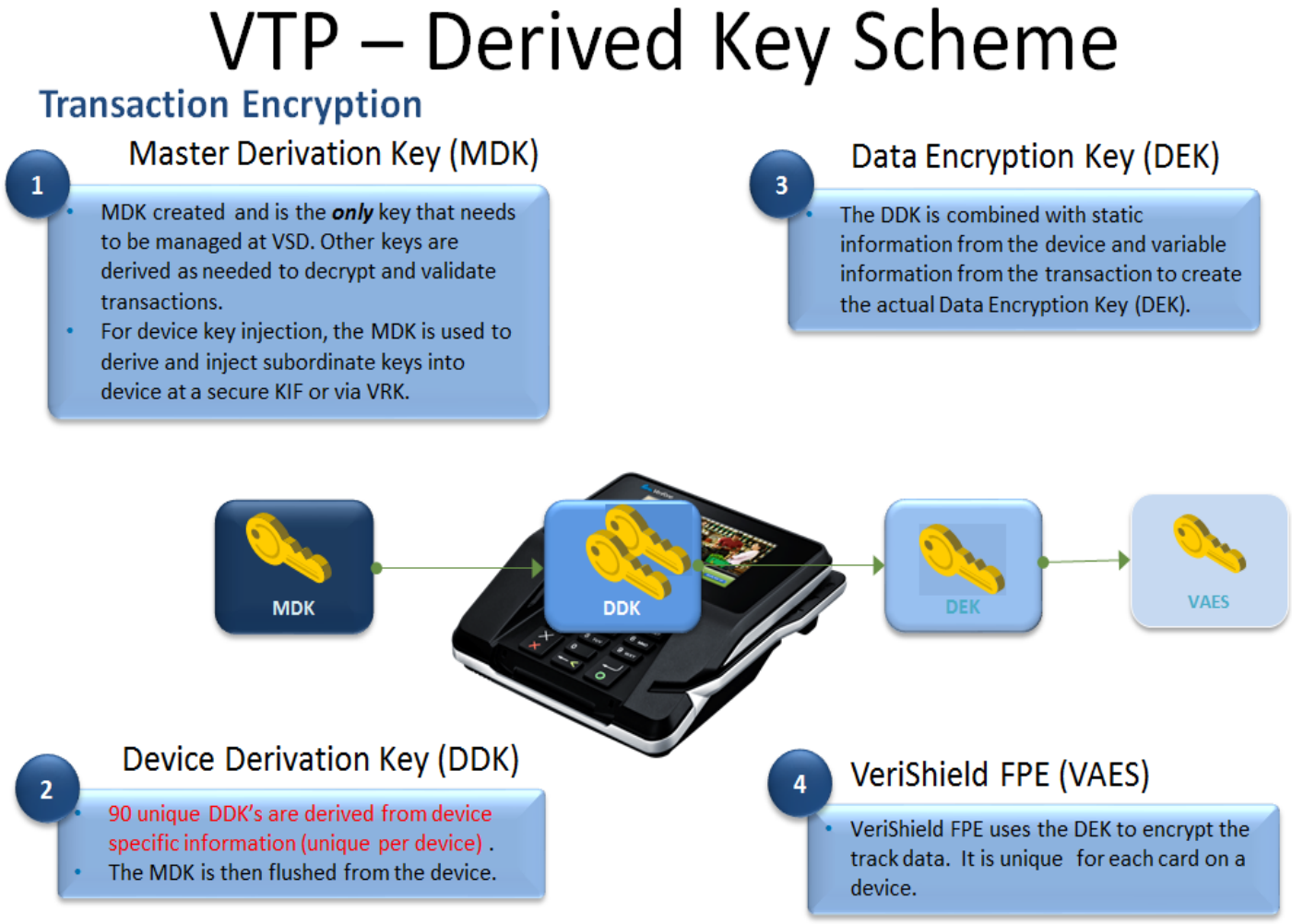
Overview

This encryption solution uses symmetric derived keys that ensure the working and decryption keys can be dynamically derived per device. Card data is preserved during the encryption process, allowing merchants to seamlessly pass data without having to modify the message format or system infrastructure. Below are some of the key terms that help explain how derived key works.

Term	Description	Location
POS	Point of Sale	Merchant Location
MDK	Master Derivation Key	Secured at Chase host Securely injected into terminal
DDK	Device Derivation Key	Created after the MDK is loaded into the device (stored on the device)
VCL	VeriShield Crypto Library (Verifone)	On the PIN pad or integrated device
Device	PIN pad or Integrated Terminal device	Merchant Location
FPE	Format Preserving Encryption	
VSD	VeriShield Decryption Service	Secured at chase host

Key Sync and Rotation

With derived key, only the Master Derivation Key (MDK) is loaded on the device. Once the MDK is loaded, 90 DDKs are derived and the MDK is securely deleted and points to the first DDK. The DDK is then combined with static information from the device and variable information from the transaction to create the actual Data Encryption Key (DEK), which results in unique keys per device.



Notes:

1. Verifone Derived Key is available to U.S. merchants.
2. Merchants must settle transactions via one of the Chase Merchant Services back-end platforms.
3. IP back channel, derived key or TGK encryption packages cannot be set up on the same Client/Division.
4. A Chase project manager should be assigned to large merchant implementations.
5. Multi-merchant configured devices are not supported.
6. Merchants must be certified to process either PNS ISO or UTF as HCS or TCS.
7. Integrators should reach out to Technical.Implementations@chase.com to obtain testing and certification information.
8. Even if the Integrator is PNS ISO- or UTF-certified, adding the E-Parms (PE Token) requires a certification to validate proper registration and key sync capability.

9. Test devices should be ordered through a Chase Relationship manager unless purchasing directly from Verifone.
10. Query the device status using the Verifone VCL API to regularly confirm encryption is enabled.
11. Do not initiate or allow a key advance with an open batch (online or offline).
12. Enable a method for merchants to initiate the Key Sync command process to manage key rotation as required by QSA.
13. The MSR swipe on the register must be disabled to avoid processing clear card transactions.

Supported Devices

- VX-Series
- MX-Series
- E-series

Command Events

Registration and Key Advance Messages

Device Registration refers to the process of enabling encryption on the device and registering the device within the Chase environment. The encrypted PAN of the registration event is 15 digits and the command BIN begins with 111111. Key registration events and key advance requests follow the same message format, but return different result codes upon successful completion. To initiate a command event, the ECR/POS integrates with Verifone's API and enables the ability to generate and send the command to the Chase host.

Note: A 30-second delay should be added after command events to ensure replication on the CHASE host has been completed.

Request Message

1. Command Events must be formatted as credit card authorization-only transactions:
    a. UTF 1.97 – Transaction Code 2
    b. PNS ISO – 1100 Message, Bit 3 = 009100
2. The first six digits of the Cardholder Account Number (BIN) must be sent as '111111'.
3. The PAN Encryption indicator must be present in the message:
    a. UTF 1.97 – PE Token
    b. PNS ISO – Bit 62, Subtag 'P4'
4. POS/VAR Capabilities Data must be submitted:
    a. Sub-element P12 must be submitted with a hex value of '80' to indicate Semtek PAN Encryption.

Response Message

1. Response message is format:

    a. UTF 1.97 – See ‘Key Registration’ in the UTF 1.97 HCS Main Specification

    b. PNS ISO – See ‘Key Update Request Process’ in the PNS ISO Specification

    - Host responds with a 1510 message

2. The encryption response code is included in the Host response:

    a. UTF 1.97 – SR Token

    - Full list of CDS result codes are in the (UTF 1.97 Token Reference Guide)
    - Semtek 909=Successful Key advance
    - Semtek 905=Successful registration

    b. PNS ISO – Bit 48, Subtag ‘S9’

    - Full list of Semtek result codes are in the (PNS ISO Specification, Appendix:A “CDS Result Codes”)
    - Semtek 909=Successful Key advance
    - Semtek 905=Successful registration

Integrator Considerations

Verifone’s VCL API supports several administrative APIs to be used by payment applications. For questions please reach out to your Verifone technical representative.

- Device registration
- Re-registration for device movement
- Advance DDK for key advance
- BIN table or settings file update

Example use cases an Integrator would work through with both Verifone and Chase.

New device installed and encryption must be turned on

1. A user initiates the administration key on the ECR/POS, which notifies the device payment application to initiate a registration function in VCL.
2. Device Payment Application initiates a registration function in VCL, which causes VCL to obtain the derivation material needed to support key management and package it up into what looks like an authorization request
3. VCL performs the registration function, turn encryption on and generates a Command Response.
4. Device Application retrieves the Command Response
5. ECR/POS obtains the Command Response from the Payment Application and sends it up through the normal authorization path.
6. Device is now registered with the VSD server on the backend and ready to begin encrypting transactions.

Device Change

1. The steps outlined in Use Case 1 could be used to simply generate a new registration event, or additional intelligence could be built into the ECR/POS as follows:

2. ECR/POS software changes are implemented to obtain and store the derivation material used by the device (e.g. serial number) and monitor changes during daily startup.
3. When changes are detected, the ECR/POS initiates administration mode which notifies the device payment application to initiate a registration function in VCL.
4. The Device Payment Application initiates a Registration function in VCL.
5. VCL performs the registration function and generates a Command Response.
6. Device Application retrieves the Command Response.
7. ECR/POS obtains the Command Response from the Payment Application and sends it up through the normal authorization path.
8. Device is now re-registered with the VSD server on the backend and can continue encrypting transactions.

Settings Change

1. ECR/POS software changes are implemented to be able to initiate specific functions in VCL through the Device Payment Application.
2. When the Merchant wants to Advance the DDK, the ECR/POS administration option for the function
3. is submitted and the Device Application is notified to initiate function in VCL.
4. Device Payment Application initiates the function in VCL.
5. VCL performs the function and generates a Command Response.
6. Device Payment Application retrieves the Command Response.
7. ECR/POS obtains the Command Response and sends it up through the normal authorization path.
8. Physical device and virtual device are now in sync with respect to which key to use for
9. decryption/derivation and/or which BINs to exclude from encryption

Verifone IP Back Channel (IPBC)

Overview

This encryption solution is sometimes referred to as “Key Pull” key management. It is facilitated through a Verifone Mx8xx or Mx9xx series PIN pad (integrated multi-lane device). There are no transaction message-level indicators related to the encryption and the integration system does not handle any of the key management traffic. Instead, each Mx device connects directly to the Chase Key Management Service through a dedicated TCP/IP connection (Frame, VPN or MPLS).

After registration and setup, the device “polls” the Key Management Server at pre-determined intervals to “pull” new key information from the server. The IP Back Channel solution is only available to U.S. Merchants.

Notes and Considerations

- IPBC is available to U.S. Retail Merchants only.
- All IPBC setups are handled as a custom setup.
- IPBC and TGK packages cannot be setup on the same Client/Division.
- Merchants must settle transactions via one of the Chase Merchant Services back-end platforms.
- For new IPBC merchants, an implementation project manager from Verifone and one from Chase (MIT) should be involved. Typical implementations can take 6-12 weeks. *(Longer if a full Class B certification is needed)*
- Devices should be set to poll every 15 minutes or more depending on network conditions. Consult with a Chase network administrator to determine network needs based on number of devices and network volume.

Mx8xx and Mx9xx devices

The multi-lane Mx devices must be integrated to a certified middleware or POS system and be updated with:

- Verifone Kmailman – Manages the IPBC traffic, key sync etc.
- Verifone VCL – Encryption firmware
- Chase Encryption Package – Keys and BIN exclusions
- IP enabled Mx cables =

The IP Back Channel Network

- Kmailman communicates to the Chase decryption server and handles device registration, key sync requests, and BIN updates.
- Jobs can be scheduled on the Chase decryption server to perform the following tasks:
    - Obtain a device status
    - Stop/re-start encryption
    - Perform a key sync
    - Update the BIN exclusions
    - Merchants need to have their dedicated private connection (MPLS/VPN) to Chase set up
        - IP address(es) and port numbers are supplied by Chase
        - ***Note***: Authorization traffic still uses the existing connection to Chase.
- The Kmailman application is a background application that checks in with the Chase decryption server. This check contains ~220 bytes of data.
- If the IP back channel connection is lost or down, transaction processing continues without interruption. The device continues to attempt the connection at each polling interval.

Integrator Considerations

- Integrate the POS to the Mx PIN pad device, typically using Verifone FormAgent application on the Mx.
- Set the environment variables on the PIN pad to properly register the device to Chase. The environment variables map the Auth IDs to the Chase Decryption Server keys.
- Send the key sync command to the Mx PIN pad – Also done by FormAgent upon updating the environment variables.
- Must be certified to process either PNS ISO or UTF as a HCS environment. Integrators should reach out to Technical.Implementations@chase.com to obtain testing info

Testing and Certification Info

- Must be certified to process either PNS ISO or UTF as a HCS environment.
    - Even if the Integrator is PNS ISO or UTF certified, adding IPBC requires a supplemental certification to validate proper registration and key sync capability.
- Test devices should be ordered through Verifone and injected with the test keys (referred to as SQA01 Keys at Verifone).
    - Devices with production keys cannot be used for testing.
    - Devices with test keys cannot be used in production.
- IPBC registration and Chase Merchant Services card-present host testing values are provided by the IRM Certification analyst.

Best Practices

- Disable all card entry on the POS, including manual entry.
- Query the device status and confirm encryption regularly; do not allow unencrypted processing.
- Do not process store-and-forward or offline transactions.

- Do not initiate or allow a key sync with an open batch (online or offline).
- Force a registration and key sync if a PIN pad is being replaced, or moved between POS lanes before any transactions are attempted.

Key Sync and Rotation

- FormAgent automatically performs a device registration and key sync when a change to the environment variables is detected.
- The PROV Key (for performing key syncs) is unique per merchant/client.
- The PAN key is unique per merchant number. Rotation requires a coordinated job setup on the Chase IPBC system and must be setup manually for each merchant number.
- The discretionary key is unique per device. Rotation requires a coordinated job setup on the Chase IPBC system and can be done across the domain/client base.
- Rotation of the keys is required every three years. Annual key rotation is considered a best practice.

Disabling Encryption

Chase can setup an IPBC job to disable (or re-enable) encryption on one or all devices in a domain/client base.

Verifone Terminal Generated Keys

Overview

Terminal Generated Key (TGK) encryption solution is sometimes referred to as “key push” key management. It is facilitated through the Vx820 PIN pad or a standalone device (Vx510, Vx520, or Vx680 3G). Transaction message-level indicators (E-Parms – PE Token) must be included in all transactions and the device uses existing communication paths for transaction processing to perform key management functions. Integration to a POS system is available using the Vx820 PIN pad.

- TGK is available to both U.S. and Canadian retail merchants.
- Merchants must settle transactions via one of the Chase Merchant Services back-end platforms.
- IPBC and TGK packages cannot both be set up on the same client/division.
- A Chase project manager should be assigned for large merchant implementations.
- Multi-merchant configured devices are not supported
- A 30-second delay should be added after command events to ensure replication on the Chase host has been completed.

Standalone Devices – Vx520, Vx510-DC, and Vx680-3G

The standalone Vx devices need to be Chase-signed and downloaded with:

- **Verifone EMA**† – Manages the device updates and key rotations. EMA performs automated key rotations annually by default. Rotations can also be done manually.
- **Verifone TGK**† – Application that performs key generation.
- **Verifone XEPH**† – Current Chase Class A SoftPay HCS application.
- **Verifone VCL**‡ – Encryption firmware.
- **Chase Encryption Package**‡ – Keys and BIN exclusions.

Integrated Device – Vx820 PIN pad

The Vx820 PP device needs to be Chase igned and be downloaded with:

- **Verifone TGK**‡ – Application that performs key generation.

- **Verifone XPI**‡ – Current Vx PIN pad Integration Software.
- **Verifone VCL**‡ – Encryption firmware.
- **Chase Encryption Package**‡ – Keys and BIN exclusions.
- Integrator needs to process TGK command transactions… BIN 111111.
- Integrator needs to be certified for E-Parms (PE Token) data within Auth messages (*see UTF or PNS-ISO specifications).

Testing and Certification Info

- Must be certified to process either PNS ISO or UTF as a HCS environment.
    - Integrators should reach out to Technical.Implementations@chase.com to obtain testing and certification information.
    - Even if the Integrator is PNS ISO or UTF certified, adding the E-Parms (PE Token) requires a certification to validate proper registration and key sync capability.
- Test devices should be ordered through Verifone, TASQ, or Maxwell and injected with test keys.
    - Devices with production keys cannot be used for testing.
    - Devices with test keys cannot be used in production.
- TGK and Chase Merchant Services card-present host testing values are provided by the certification analyst.

Best Practices

- Disable all card entry on the POS including, manual entry.
- Query the device status and confirm encryption regularly; do not allow unencrypted processing.
- Do not process store-and-forward or offline transactions.
- Do not initiate or allow a key sync with an open batch (online or offline).
- Force a TGK sync if a PIN pad is being replaced, or moved between POS lanes before any transactions are attempted.
- Enable a method for Merchants to initiate the TGK Key Sync command process to manage key rotations annually to all devices.

Semi-integrated devices – Vx520, Vx510-DC

Semi-integrated Vx devices need to be Chase signed and be downloaded with:

- Verifone EMA† – Manages the device updates and key rotations. EMA performs automated key rotations annually by default. Rotations can also be done manually.
- Verifone TGK† – Application that performs key generation.
- Verifone XEPH† – Current Chase Class A SoftPay application.
- Verifone ECRi† – Interface application between POS system and Vx device.
- Verifone VCL‡ – Encryption firmware.
- Chase Encryption Package‡ – Keys and BIN exclusions.

  Note: The semi-integrated ECRi solution is limited to a basic set of command functions that can be sent from the POS system (sale, refund, etc) *See the ECRi Integration Guide on Developer Center for more details.

†- Downloaded via Chase VeriCentre
‡ - Injection performed at TASQ or Verifone

Key Sync and Rotation

- On stand-alone devices, EMA automatically performs a device registration and key sync upon first boot-up when keys are first injected.

- The PROV Key (for performing Key Sync's) is unique per merchant/client.
- The PAN Key is unique per Merchant number.Rotation is managed by an EMA Parameter on stand-alone devices (monthly/quarterly/annually).
- The Discretionary Key is unique per device. Rotation is managed by an EMA Parameter on stand-alone devices (monthly/quarterly/annually).
- Rotation of the Keys is required every three years. Annual key rotation is considered a best practice.

Verifone TGK/XPI Key Sync Process

1. The controller triggers the key event and sends the E04 command to the XPI.
2. The XPI communicates with the VCL/TGK module to get new key material. The VCL/TGK Module generates new PAN and discretionary keys.
3. The XPI passes all three keys back to the controller.
4. The controller creates and sends the first authorization request with the first key part to the Chase host.
5. The host responds to the controller with a batch inquiry response. The controller interrogates the encryption response code to determine the result. If successful, the response = 941.
6. The controller sends the next authorization request with the second key part to the Chase Host.
7. The host responds to the controller with a batch inquiry response. The controller interrogates the encryption response code to determine the result. If successful, the response = 942.
8. The controller sends the next authorization request with the third key part to the Chase Host.
9. The host responds to the controller with a batch inquiry response. The controller interrogates the encryption response code to determine the result. If successful, the response = 943.

Note: It is up to the controller to manage the response received from the Chase host. If the key update is unsuccessful (response code = 940), the controller should respond with a new TGK update request until the key management task is successful.

TGK Update Request and Response Message Formats

The following request and response messages are required between the controller and the Chase Host. Integrators should use the Chase Technical Specifications, the VX 820 XPI specification and the Integrator XPI Handbook to make necessary changes to their payment application.

Three requests and responses are required to complete the TGK Update Request. Once all three requests have been completed successfully, encryption is enabled and the new keys are in use.

Request Message

1. Key update requests must be formatted as credit card authorization-only transactions:
    a. UTF 1.97 – Transaction Code 2
    b. PNS ISO – 1100 Message, Bit 3 = 009100
2. The first six digits of the Cardholder Account Number (BIN) must be sent as '111111'.
3. The transaction sequence flag must be sent as '2' for multi-part messages.
4. The PAN Encryption indicator must be present in the message:
    a. UTF 1.97 – PE Token
    b. PNS ISO – Bit 62, Subtag 'P4'
5. POS/VAR Capabilities Data must be submitted:
    a. Sub element P12 must be submitted with a hex value of '80' to indicate Semtek PAN Encryption.

Response Message

1. TGK update response message is formatted as a batch inquiry response:
    a. UTF 1.97 – See 'Terminal Generated Key Registration' in the UTF 1.97 HCS Main Specification

b. PNS ISO – See 'Key Update Request Process' in the PNS ISO Specification

2. The encryption response code is included in the Host response:
   a. UTF 1.97 – SR Token
   b. PNS ISO – Bit 48, Subtag 'S9'

Disabling Encryption

Chase can set up a VeriCentre file download that clears the key from the standalone Vx devices.

Note: Encryption cannot be re-enabled once it has been disabled.

Ingenico On-Guard

This encryption solution is supported on the Ingencio iPP320 PIN pad (baseline version CPX 904 or later) or any other Telium 2 device that supports On-Guard. This implementation features 2 Key Triple DES DUKPT key management along with full card number, and/or Track 2 encryption via an algorithm that produces format-preserving encryption (FPE). All data is numeric and is the same length as standard Track 2 data. Message-level indicators are required for successful decryption of encrypted transactions.

Merchant Environment and Setup Requirements

- U.S.- or Canadian-based retail card-present merchant environment
- Host Capture (HCS) Processing (PNS-ISO or UTF)
- Process via the Chase Merchant Services card-present front-end platform.
- Merchants must settle transactions via one of the Chase Merchant Services back-end platforms.
- Transmit transactions through any supported communication method (TCP/IP, NetConnect or Dial).
- Merchant provided list of third-party BIN ranges to be excluded from encryption (i.e. gift, loyalty) if needed
- Have appropriate device hardware
    - iPP320 or any other Ingenico Telium 2 device that supports On-Guard
    - PIN pad must be loaded with the required software and injected with the E2EE DUKPT key
    - DUKPT Key Management for PAN encryption/decryption
        - Key rotations are not required in a DUKPT environment

BIN Exclusions

Once encryption is enabled, all cards regardless of the issuer are encrypted. Cards which meet ISO standards for magnetic track data (Swiped), EMVco chip cards (Inserted), Contactless (Tapped), NFC (Mobile), as well as manually keyed cards are encrypted ***before*** being passed to the payment application for processing.

BIN Exclusion Management

1. Chase provides a generic BIN Exclusion file that houses known ISO gift card BINs.
   BIN EXCLUSION File (E2EE2BIN) is separate from the SECURETXTPROMPT File.
   a. Merchant integrator has the ability add additional BIN ranges to this file.
   b. A project request is required to ensure that the new BIN ranges are captured, reviewed and finalized.
      I. Identify added BIN Ranges
      II. Ideniify card entry method on PED (chip, swiped, manual)
2. Upon approval of the project, Chase creates a MAC value for each line that is modified in the file.
3. Chase provides the MAC value upon completion of project.
4. VAR is responsible for testing updated BIN Exclusion file to confirm BIN Exclusion does not impact merchant processing.

Integrator Considerations

- Integrate the POS to the IPP320 Pad device using Ingenico Telium CPX E2E Specification and other relevant Ingenico Telium Specifications. Contact your Ingenico representative to obtain these specifications.

- Utilize the E2EE Activate Command and CPX configuration settings to enable encryption. The E2EE configuration file “e2ecfg” is a Chase signed file and CPX only uses the validated version.
- Certify to process either PNS ISO or UTF as a Host Capture (HCS) environment. Integrators should reach out to Technical.Implementations@chase.com to get testing and certification info
- Confirm the CPX version is able to support Onguard Encryption.
- Manage updates and distribution of the Chase signed BIN Exclusion file for merchants. The integrator is responsible for working with the merchant to identify whether the merchant requires:
    - BIN Exclusion as part of their encryption solution,
    - Additions to the standard BIN Exclusion file supplied by Chase.

Testing and Certification Info

- Certify to process either PNS ISO or UTF as a Host Capture (HCS) environment. Integrators should reach out to Technical.Implementations@chase.com to obtain testing and certification info.
- Order testing devices through Ingenico and inject them with the Chase DUKPT Test Key (referred to as Chase Test DUKPT Key at Ingenico)
    - Production devices cannot be used for testing.
    - Test devices cannot be used in production.
- Register the test PIN pad’s serial numbers with IRM.
- Chase card-present host testing values are provided by the IRM Certification analyst.

Best Practices

- Disable all card entry on the POS including manual entry.
- Query the device status and confirm encryption regularly; do not allow unencrypted processing.
- Processing store-and-forward and offline transactions is not recommended when encryption is enabled. If performed, all fraud risk falls on the merchant.
- Ensure encryption is enabled whenever a PIN pad is replaced or moved between POS lanes and before any transactions are processed.

Disabling Encryption

- The E2EE configuration file can be downloaded by switching the E2EE feature from disabled to enabled. The feature remembers being enabled such that downloading a configuration file to disable E2EE is ignored.
- For some devices (e.g., iSMP family of devices), E2EE cannot be disabled. CPX displays the message "E2EE Error /E2EE Not Enabled" if the E2EE mode is set to disabled and CPX does not run.

Merchant Considerations:

- Are any cards being accepted that do not process through Chase?
- Even though the merchant is accepting non-ISO format gift/loyalty cards, is there a chance they manually enter those cards?
- By default, Chase excludes Petroleum and Private Label BINs listed in our Processing and Interchange Guidelines document.

Telium Terminal configuration file contains four components:

1. E2EE mode
    - ‘2’ encryption mode enabled
    - ‘D’ encryption mode disabled
2. Format Type - ‘B’ E2EE utilizing DUKPT key.
3. Type of Key - ‘D’ E2EE DUKPT
4. Key Number – ‘5’ specifies E2EE injected into slot 4 using Key Pattern 4

Encryption Output to Host

The IngeCrypt module in the device provides output as a 'raw' KSN. This KSN must be passed, along with additional data to the PNS Host for decryption.

Example of *KSN* and *extended data* in the payload:
FFFF9876543210E000070114

This extended data is appended to the KSN in a 'custom field', and passes information about the encryption parameters to the decryption appliance. Appending this information is optional if the default configuration parameters of the decryption appliance match the ones used in the terminal. The only instance where appending the data is mandatory is when padding digits have been generated when the standalone PAN is too short.

MagTek Magnesafe

This encryption solution is supported on any MagTek PINPad that supports MagneSafe. This implementation features 2 Key Triple DES DUKPT key management, along with full card number and/or Track encryption via an algorithm that uses format preserving encryption (FPE) or non-format preserving. The PNS host supports both methods; however, the method type must be identified in the incoming request. M1 is non-format preserving, and M2 is format preserving. All data is numeric and is the same length as a standard track. Message level indicators are required for successful decryption of encrypted transactions.

Merchant Environment and Setup Requirements

- U.S. based retail card-present merchant environment
- Process in a card-present, retail environment through the Paymentech Network Services (PNS) front-end platform.
- Host Capture (HCS) Processing (PNS-ISO or UTF)
- Settlement to North American (NAP) or Global backend
- Transmit transactions through any supported communication method (TCP/IP, NetConnect or Dial).
- Have appropriate device hardware
    - MagTek uDynamo, DynaPro, DynaPro Mini or any other MagTek device supporting MagneSafe
    - PIN pad must be loaded with the required software and injected with the E2EE DUKPT key
    - DUKPT Key Management for PAN encryption/decryption
        - Key rotations are not required in a DUKPT environment

BIN Exclusions

No BIN exclusions are currently supported with MagneSafe devices.

Encryption Output to Host

The device provides output as a raw unformatted or format preserving. Below are examples of both.

Non format preserving example (M1):
KSN=9010010B2485A8000344
Track1.Masked=%B6011020004005098^DISCOVER PREPAID/CPS TEST ^20120000000000000000?
Track2.Masked=;6011020004005098=20120000000000000000?
Track1.Encrypted=8D8CDC173FD06471174D33F305C7980053448A0C5B9EF3990871FCC552389C533797394D8B2DCFCA3820461AEE2B608089E4F13DDDAF1F29CC2A80596156D245AE157E0EB03AEB9F
Track2.Encrypted=CD7A29514DDD378E7B76C7D737513BDCDAE841A6AF9596731412549B7A7005AEFE8A9725C23F932C

Notes:
1. Masked data is used to populate PNS ISO Bit 2 – Primary Account Number and Bits 35 – Track 2 Data and 45 – Track 1 Data.
2. Encrypted data is used to PNS ISO Bit 62, Subtag P4 – PAN Encryption.

Format Preserving example (M2):
%B5490500066748828^SMITH/WILLIAM^10101010000000000000000000000000732001000?;5490500066748828=10101010000073200100?

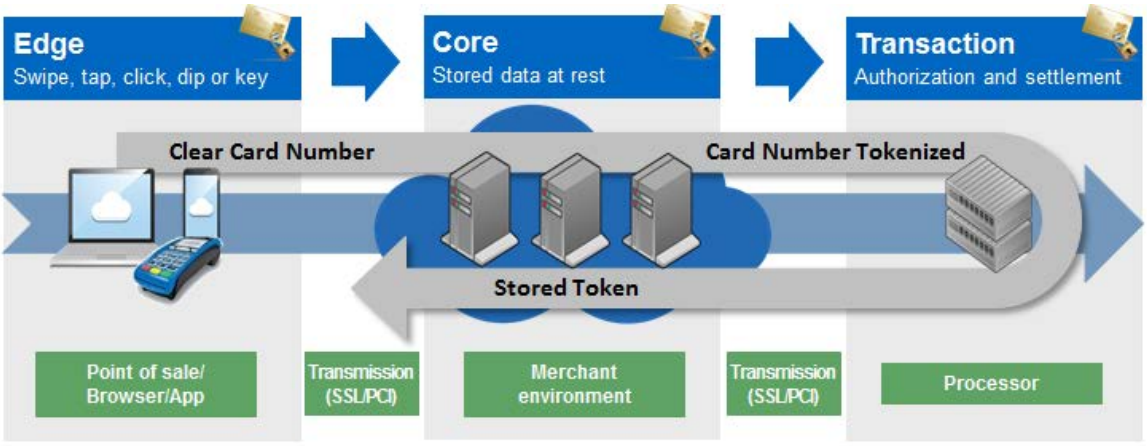
Safetech Tokenization

Overview

To address cardholder data security, Chase offers the Safetech Tokenization solution on both the PNS and card-not-present platforms. The solution entails returning a token in the transaction response message in lieu of clear-text card data. By returning tokens to merchants, no cardholder data is passed in subsequent transactions as the token serves as a proxy for the original account number (see transaction flow below). Merchants can use tokens in lieu of clear card numbers in their internal systems as well.

Safetech Tokenization can be used as a standalone solution for both PNS and card-not-present merchants. For added security, encryption can also be combined with Safetech Tokenization. For PNS merchants, any of the encryption products mentioned in the Safetech Encryption section can be paired with Safetech Tokenization. For card-not-present merchants, Safetech Page Encryption can be paired with Safetech Tokenization.

Note: Safetech Page Encryption is not offered without Safetech Tokenization.



Benefits

- Reduces risk if a merchant's system is breached
- Potentially reduce your Payment Card Industry Data Security Standards (PCI DSS) compliance scope.
- Secures card account data (chip, contactless, swiped, or manually keyed)
- Integrates this solution with current Ingenico, Magtek, and Verifone terminals
- Continues accepting payments as usual, with no change in the customer's experience

Merchant Environment and Setup Requirements

PNS

Safetech Tokenization is offered for all industry segments supported on the PNS platform for U.S. and Canadian merchants. Both PNS ISO and UTF message formats support Safetech Tokenization in HCS processing only on the following card types:

- Visa

- Mastercard
- American Express
- Discover, Diners
- JCB
- FSA/IIAS
- U.S. and Canadian Debit
- International Maestro.

Safetech Tokenization supports all transaction types where a Primary Account Number (PAN) is passed between a merchant and PNS, as long as it does not interfere with card brand requirements. Please refer to the PNS ISO or UTF specifications for a list of these transaction types.

Merchants should consider what token format best fits their business needs prior to enrolling in Safetech Tokenization. For example, a merchant may need to preserve a portion of the PAN or may need the token to meet MOD10 checks. Safetech Tokenization offers 16 different token formats to choose from. Please speak with a Chase account representative to determine which token format best meets a merchant's business needs.

Safetech Tokenization Formats with Examples

Note: The formats below have been upgraded to the new Secure Stateless Tokenization (SST2) format. Legacy token formats will remain valid for existing Chase merchants that are already configured for Safetech Tokenization. New Safetech Tokenization merchant implementations must use the new SST2 format.

PAN Digits to Preserve	Alpha Token Indicator	MOD 10 Checksum	SST2 Format (new)	Example Token	Example MOD 10
First 6 and Last 4	None	Ignore MOD 10	NI64	5454541002111234	Fail/ignore
First 6	None	Ignore MOD 10	NI60	5454541002112001	Fail/ignore
Last 4	None	Ignore MOD 10	NI04	1806551002111234	Fail/ignore
None	None	Ignore MOD 10	NI00	1806551002112001	Fail/ignore
First 6 and Last 4	None	Preserve MOD 10	NP64	5454541002801234	40
First 6	None	Preserve MOD 10	NP60	5454541002482001	40
Last 4	None	Preserve MOD 10	NP04	1806551001011234	40
None	None	Preserve MOD 10	NP00	1806551004312001	40
First 6 and Last 4	None	Change MOD 10	NC64	5454541002981234	50
First 6	None	Change MOD 10	NC60	5454541019982001	50
Last 4	None	Change MOD 10	NC04	1806551002671234	50
None	None	Change MOD 10	NC00	1806551008782001	50
First 6 and Last 4	Alpha Token	Ignore MOD 10	AI64	5454541wrtp11234	Fail/ignore
First 6	Alpha Token	Ignore MOD 10	AI60	5454541wrtp12001	Fail/ignore
Last 4	Alpha Token	Ignore MOD 10	AI04	1806551wrtp11234	Fail/ignore
None	Alpha Token	Ignore MOD 10	AI00	1806551wrtp12001	Fail/ignore

Applicable Message Types

Safetech Token-Only request can only be submitted using the 1100 message type in HCS only. The Safetech Token will be included on the 1110 response.

Refer to Chase UTF and PTI Specifcations for specific tokenization transaction codes.

Consumer Digital Payment Tokens

Merchants requesting Safetech Tokens for cards provisioned in digital wallets will receive different Safetech Tokens than for cards that are swiped, tapped, dipped, or manually entered on the terminal. Cards provisioned in a digital wallet receive a digital payment token by the wallet provider, known as a DPAN. Safetech Tokens are generated off of either the PAN or DPAN presented to the terminal, and results in different Safetech Tokens being generated for the same underlying FPAN

Token Only Requests

Best Practices

- Verify that all merchant systems and processes can handle the selected token format. Once the token format has been selected and enabled at Chase the format cannot be changed.
- Safetech Tokenization is enabled on the Chase host, but a tokens must be requested during an authorization or a token only request in order to receive a token in the response message.
- If a token is requested but not received in original transaction, merchants should perform a token only request to receive a token.
- A class B certification is required when implementing tokenization.

Glossary of Relevant Terms

Acronym	Term	Detail
BDK	Base Derivation Key	Master key used in 3DES DUKPT
BIN File	BIN Exclusion File	File which lists all BIN ranges to NOT be encrypted.
CPX, UIA, RBA	Ingenico	Ingenico PIN pad integration software that supports Encryption.
CTLS	Contactless	OS Firmware running on the Vx device which manages an integrated contactless reader.
DPAN	Digital Primary Account Number	
DA	Decryption Appliance	Generic name for the decryption server stack. – See VSD
Triple DES	Cipher Algorithm	Triple Data Encryption Stanard (DES Cipher applied 3 times)
DUKPT	Key Management Scheme	Derived Unique Key Per Transaction
ECRi	Electronic Cash Register Interface	Application running on the Vx device which enables a POS system to send commands to the Softpay application

Acronym	Term	Detail
		(aka: semi-integration)
EMA	Estate Management Application	EMA is the application running on the Vx device which drives the TGK application and triggers SoftPay to connect to the host.
FormAgent	Mx PIN pad software	Application used on a PIN pad to communicate to a connected a POS system.
FPAN	Funding Primary Account Number	
FPE	Format Preserving Encryption	Generic term used to descript E2E encryption that does not require changing the Track data format on the card.
HSM	Hardware Security Module	Secure system which houses/protects encryption keys.
IngeCrypt / OnGuard	Ingenico OnGuard	Ingenico branded end to end encryption product.
IPBC	IP Back Channel	Mx Device management connection to the decryption server.
KEK	Key Encryption Key	Standard term to describe a key that is used to encrypt another key.
KIF	Key Injection Facility	Used interchangeably to describe the deployment/injection facility itself or the specific KIF cards/injection key.
Kmailman	Application forIP Back Channel communication	Kmailman communicates over the back channel to the VMB interface and to the VKM interface.
MSR	Mag Stripe Reader	Card reading device on a PIN pad or terminal.
Mx Device	Multi-Lane PIN pad	Used in conjunction with a POS system. Common examples: Mx830, Mx915, Mx925.
PROV	Provisioner Key	Key used to encrypt and transmit the PAN encryption keys between the device and the decryption server during a key sync.
Root	Root Key	Root Key would be used to update the PROV key should it be compromised.
Softpay	Stand alone countertop payment software (Vx)	Common example: XEPH410
Telium 2	Telium 2 Devices	Ingenico Telium 2 hardware platform brand name. Examples iPP320 and iCT250

Acronym	Term	Detail
TGK	Terminal Generated Keys	TGK Process: Keys created at the device and sent to the host then passed to the decryption server. TGK Application: application running on the Vx device which creates the new keys.
VCL	VeriShield Crypto-Library	Encryption OS firmware which runs on the Mx or Vx device.
VFI	Verifone	Abbreviation for Verifone Inc.
VKM	VeriShield Key Management	Web application used to manage IP back channel devices. Turn off encryption, update BIN exclusions, and update keys.
VMB	VeriShield Merchant Boarding	Web application used to import new Keys, board merchants from Tandem and export config packages
VMC	VeriShield Monitoring and Compliance	Web application that provides reports of encrypting devices and transaction activity.
Voltage	Voltage Encryption	Voltage is an encryption/tokenization brand and methodology.
VRK	Verifone Remote Key	Verifone's remote Key injection process capable of injecting keys to devices remotely – Debit and E2E.
VSD	VeriShield Decryption	Generic name for the decryption server stack. – See DA
VSP	VeriShield Protect	The Verifone product brand name.
Vx Device	Stand alone countertop Terminal or PIN pad	Common examples: Vx520, Vx820 PP
VxCI	Vx Contactless Interface	Application running on the Vx device which manages the contactless data received from the CTLS firmware and sends it to the SoftPay application.
XPI	External PIN pad Interface (Vx)	Application used on a PIN pad to communicate to a connected terminal (Vx520) or to a POS system.

Appendix A – Verifone IPBC Kmailman Configuration

Example of a Kmailconfig File	Description
kserver1name=206.253.180.179	**kserver1name** – The IP Address of the Chase Decryption Server
kserver1port=8443	**kserver1port** – The port number of the Chase Decryption Server
kserver1useSSL=1	**kserver1useSSL** – Enable or disable SSL.
kserver1urlroot=/RKDS3Web.asmx	**kserver1urlroot** – Web Services page being accessed on the Chase Decryption Server
checkmail_interval=900	**checkmail_interval** – Frequency of the Kmailman check in seconds
wait_request=5	**wait_request** – Seconds before network connection time out
wait_kmailout=30	**wait_kmailout** – Seconds before server response time out
ssl_match_hostname=0	**ssl_match_hostname** – Validate the URL matches the SSL certificate value
ssl_check_dates=0	**ssl_check_dates** – Validate the server SSL certificate date value
ssl_allow_selfsigned=1	**ssl_allow_selfsigned** – Allow SSL certs not signed by a 3rd party Root CA (i.e. verisign)
ssl_depth=3	**ssl_depth** – Max number of intermediate cert issuers
DOMAIN=GET_FROM_ENV_VALUE:*pos_dom	**DOMAIN** – Environment variable Domain
MERCHANT=GET_FROM_ENV_VALUE:*pos_mer	**MERCHANT** – Environment variable Merchant ID
store=GET_FROM_ENV_VALUE:*pos_store	**store** – Environment variable Store ID (Bank or PNS MID)
terminal=GET_FROM_ENV_VALUE:*pos_lane	**terminal** – Environment variable Lane ID (PNS TID)
device=GET_FROM_ENV_VALUE:*pos_dev	**device** – Environment variable Device ID (either pump/device ID field or wildcard [*]

Appendix B – Verifone Hierarchy Environment Variables

Variable	API Variable	Description	Example	Auth Message Value
Domain	*pos_dom	Domain is assigned from the Client/Div during boarding and E2E enablement.	VAR1_0002_VFMXE	N/A
Merchant	*pos_mer	This is the PNS Merchant ID from Tandem	700000001234	PNS MID *700000001234*
Store	*pos_store	The PNS Merchant ID from Tandem is often used. However, the Bank MID could be used.	700000001234	N/A
Terminal	*pos_lane	This is the PNS Terminal ID from Tandem which needs to include a prefixed 0 to the three digit TID. Also called the Lane.	0001	PNS TID *001*
Device	*pos_dev	This is a value for the actual device. In a controller environment this is used similar to a Controller/Pump ID, though it is often not used. The default is *	*	N/A *or Lane/Pump ID (4 digits)*

Appendix C – Verifone Semtek CDS Result Codes

Result Code	Result Description	Notes
1xx	Success result code.	The VeriShield Decryption Service perceived the data as encrypted, found keys, and decrypted the data.
2xx	Unencrypted result code.	The transaction was not decrypted by the VSD because the source data was not encrypted as expected.
3xx	Transaction Error result code.	The transaction was not decrypted by the VSD due to an error in the track data or missing lookup keys in the database.
8xx	RSA Command result code.	The transaction was processed as a RSA command
9XX	VSD Command result code.	The transaction was processed as a VSD command

VSD Command result Codes 9xx

Result Code	Result Description	Notes / Resolution
901	Start Encryption Fail (DK) Server	Indicates VSD was unable to process a derived key registration command response received from the device Confirm VCL loaded, Retry,
902	Start Encryption Fail (SRED) Server	Indicates VSD was unable to process a derived key registration command response received from the device in SRED mode
904	New Keys Posted	Indicates that the device generated Discretionary key was posted to the VSD database table
905	Start Encryption Success (DK)	A derived key device registered with VSD and enabled encryption
906	Insert BIN Record	A new BIN table record was added to the device's configuration
907	Delete BIN Record	A BIN table record was deleted from the device's configuration
908	Reset BIN Table	The BIN table configuration for a device was reset
909	Advance DDK Success	A derived key device rotated to a new DDK
910	Set terminal options	A device has received updated configuration settings
911	Exhaustive PAN Max Exceeded	The device exceeded the maximum allowed number of operations for a given time window Came with SRED module, indicates over 100 swipes in a 60 second period, used to indicate possible fraud. Client response is to wait 10-20 seconds and retry.
912	Device Replacement Success	A new device registered in place of another at an existing set of locator values
913	Upgrade to Unique Key Success	A device was upgraded and is now using derived key with device unique keys
914	Advance DDK Fail Server	VSD was unable to rotate to the next DDK for that device Registart
915	Upgrade to Unique Key Fail Device	A device failed to upgrade to derived key with unique keys for a device
916	Upgrade to Unique Key Fail Server	VSD was unable to switch to derived key with unique keys for a device
919	Advance DDK Fail Device	A derived key device was unable to rotate to a new DDK
920	BIN Table Rejected By Device	BIN Replace Rejected by Device
921	BIN Table Replace Server Error	BIN Replace Unknown Server Failure

922	BIN Table Replaced	BIN Replace Success
923	Settings Update Fail Device	A derived key device failed to process a delivered settings file update
924	Settings Update Fail Server	VSD failed to apply the settings file updates that were successfully applied to the device
926	Settings Update Success	A derived key device and VSD successfully applied a settings file update
931	Encryption Start	Encryption enabled on a device using Classic Mode firmware
932	Encryption Stop	Encryption disabled on a device using Classic Mode firmware
933	Stop Encryption Fail	A device was unable to stop encryption
934	Start Encryption Fail	A derived key device was unable to start encryption
935	Start Encryption Success (SRED)	A derived key device registered with VSD and enabled encryption in SRED mode
936	Start Encryption Fail (SRED) Device	A derived key device was unable to start encryption in SRED mode
940	Key Replace Failure – TGK	VSD was unable to save the Device Generated keys delivered in multiple authorization calls
941	TGK – Transaction One	VSD saved the first of 3 key components
942	TGK – Transaction Two	VSD saved the second of 3 key components
943	TGK – Transaction Three	VSD received the last of 3 key components
944	Replace BIN Table Via TCP/IP	A device BIN table was updated via TCP/IP
948	TCP/IP Advance DDK Fail Server	VSD was unable to rotate to the next DDK for that device
949	Create BIN Mask Range	A new BIN Mask Range was created
950	TCP/IP Register Derived Fail Server	VSD was unable to start encryption for that device
951	TCP/IP Get Status	A device is reporting its status information
952	TCP/IP Start Encryption	A device enabled encryption
953	TCP/IP Stop Encryption	A device disabled encryption
954	Activate Via ECR	Activate Via ECR
956	TCP/IP Replace PAN and DISC Key	A device replaced its PAN and DISC key
957	TCP/IP Device Key Sync	A device completed key synchronization
958	Register Device	A derived key device registered successfully
959	Replace Discretionary Key	A device replaced its DISC key
960	Terminal moved	A device/terminal is reporting new locator information
972	Missing Serial Number	An eParms enabled device has not registered its serial number with VSD
973	Missing Store PAN key	There is no PAN key associated with the Store
974	TCP/IP Register Derived Key	A derived key device registered with VSD
975	TCP/IP Advance DDK Success	A derived key device rotated to a new DDK
976	Settings Update Success	A settings update was successfully applied to a device
977	TCP/IP Advance DDK Fail	A derived key device was unable to rotate to a new DDK

	Device	
978	VCL Device Upgrade Success	A device was successfully upgraded
979	TCP/IP Settings Update Fail Device	A settings update could not be applied to a device
980	TCP/IP Error	An operation failed
981	TCP/IP Settings Update Fail Server	VSD was unable to process a device settings update
982	VCL Device Upgrade Fail Device	A device was unable to process a device upgrade command
983	VCL Device Upgrade Fail Server	VSD was unable to process a device upgrade
984	TCP/IP Derived Key Device Replaced	A new device registered with VSD replacing an existing device in the same location
985	TCP/IP Register Derived Key Fail Device	A derived key device failed to process a register command
986	TCP/IP Start Encryption Fail	A derived key device was unable to start encryption
987	TCP/IP Stop Encryption Fail	A derived key device was unable to stop encryption
988	TCP/IP Start Encryption Success (SRED)	A derived key device registered with VSD and enabled encryption in SRED mode
989	TCP/IP Start Encryption Fail (SRED)	A derived key device was unable to start encryption in SRED mode
990	Get Status Derived Key	A derived key device is reporting its status information
991	Command Error	VSD encountered an error processing the command
992	Command Unknown Function	VSD was unable to process a command response due to an unknown or incorrect function code
993	Command Not Supported	VSD no longer supports the command response function
994	Set DDK Success	A derived key device has been set to use a new DDK
995	Set DDK Fail Device	A derived key device was unable to set a new DDK
996	Set DDK Fail Server	VSD was unable to set a new DDK for that device

Success Codes 1xx

Result Code	Result Description	Notes / Resolution
101	Success with old key set (This reason code can only occur if eParms is being used)	No action needed
102	Success Track 1	No action needed
103	Success Track 2	No action needed
104	Success Manual PAN	No action needed
105	Success Track 1 and Track 2	No action needed
106	Success Track 1 and Manual PAN	No action needed
107	Success Track 2 and Manual PAN	No action needed
108	Success Not MIV validated	No action needed
180	Success (PKI decryption / tokenization)	No action needed

181	Success (PKI decryption / tokenization) – encryption key will expire soon	No action needed
182	Success (PKI decryption / tokenization) – encryption key has expired	No action needed

Unencrypted Transaction 2xx – transaction continued to process

Result Code	Result Description	Notes / Resolution
201	Card data not encrypted	Turn encryption on, registart device
202	Expiry date too high to encrypt	
203	Card is expired	Ask for another card
204	BIN Excluded from encryption	
205	Invalid Track 1	Re-swipe, hand key or ask for another card
206	Invalid Track 2	Re-swipe, hand key or ask for another card
207	Invalid PAN	Re-swipe, hand key or ask for another card
208	Invalid Expiration Date	Re-swipe, hand key or ask for another card
211	PAN Luhn failed Mod 10 Check	Re-swipe, hand key or ask for another card
212	PAN Too Short	Re-swipe, hand key or ask for another card
298	Unknown Multiple Tracks	Re-swipe, hand key or ask for another card
299	Unknown Reason – No Action	

Transaction Error 3xx – transaction did not process

Result Code	Result Description	Notes / Resolution
301	Invalid Requestor	Setup Error, mis-match between Terminal and Host setups Escalate to Decryption host
302	Invalid Transaction ID	Transaction ID does not meet format requirements. Most likely this would be found during system testing.
303	Invalid Transaction Type	Transaction Type does not meet format requirements. Most likely this would be found during system testing.
304	Invalid Amount	Amount input does not meet format requirements. Most likely this would be found during system testing.
305	Invalid Domain Code	Setup Error, mis-match between Terminal and Host setups
306	Invalid Merchant Code	Setup Error, mis-match between Terminal and Host setups
307	Invalid Store Code	Setup Error, mis-match between Terminal and Host setups
308	Invalid Terminal Code	Setup Error, mis-match between Terminal and Host setups
309	Invalid Device Code	Setup Error, mis-match between Terminal and Host setups
310	Missing Payment Card Data	Merchant should try reswiping, Ask for a different card, or Key in the data. If these actions do not work, then escalate
311	Domain Code not found	Setup Error, mis-match between Terminal and Host setups
312	Merchant Code not found	Setup Error, mis-match between Terminal and Host setups
313	Store Code not found	Setup Error, mis-match between Terminal and Host setups

316	Device not added	Setup Error, mis-match between Terminal and Host setups
317	Missing PROV key	Dogwood Error - Key mis-match between Terminal and Host
318	Missing PAN or DISC key	ELM Error - Key mis-match between Terminal and Host
320	Encrypted PAN Mod 10 Failure	Merchant should ask for a different card, or Key in the data. If these actions do not work, then escalate
321	Invalid eParms Data	E-Parms not being used, this response should not be seen
322	Key Service is Not Reachable	Key mis-match between Terminal and Host
323	Key Sync ID not found	
324	Missing eParms Data	Key not in Verifone Host Registart
325	Missing MDK	Missing data in Verifone Host Registart
326	Missing Derivation Data	Key mis-match between Terminal and Host Registart
327	Derived Key Derivation Error	Could be one of two root causes: 1) Missing data in Verifone Host - Could be cleared up by Registart OR 2) Key Mis-match between Terminal and Host
328	MAC Error Track 1 PAN	Dogwood Error - during Decryption, errors are found Dogwood Activate
329	MAC Error Track 1 DISC	Dogwood Error - during Decryption, errors are found Dogwood Activate
330	MAC Error Track 2 PAN	Dogwood Error - during Decryption, errors are found Dogwood Activate
331	MAC Error Track 2 DISC	Dogwood Error - during Decryption, errors are found Dogwood Activate
332	MAC Error Manual PAN	Dogwood Error - during Decryption, errors are found Dogwood Activate
333	MIV Error Track 1 PAN	ELM Error - during Decryption, errors are found Registart
334	MIV Error Track 2 PAN	ELM Error - during Decryption, errors are found Registart
339	Virtual Device Conflict Detected	
380	RSA KeyID Not Found	
381	RSA Decryption Error	
382	RSA Invalid Blob	
383	RSA Configuration Name already exists	
384	RSA Configuration Name cannot be found	
385	RSA returned no data	
386	RSA - Locator does not support VTP	

387	RSA Web Service Initialization Error	
388	RSA Token Creation Error	
389	RSA Tokenization Error	
390	RSA Detokenization Error	
391	RSA Result Ambiguous	
392	Invalid Parameter [Parameter Name]	A mis-match in the Terimal vs. Host Setup
393	Host Access Denied – [Host Address]	ELM has the capability to restrict traffic by IPs if needed (ex. block Fraud) This indicates the transaction came from a blocked IP, could indicate VSP setup issue if the IP is valid
397	HSM Error	HSM communication error
398	Decryption Error Multiple Tracks	
399	Decryption Failure – Unknown Error	

RSA Command result codes 8xx

Result Code	**Result Description**	**Notes / Resolution**
801	RSA Key Generated Success	
802	RSA Key Provided Success	
803	RSA Certificate Error	
804	RSA Deactivate Key Success	
805	RSA Deactivate Key Error	
806	RSA Revoke Certificate Error	
807	RSA Key Pair Generation Error	
808	RSA Create Configuration Success	
810	RSA Certificate Generation Error	
811	RSA Issue Certificate Error	
812	RSA Store Certificate Error	
813	RSA Unable to Get Key ID	
814	RSA Unable to Create Token Type	