

PAN Encryption: Yes, we can standardize now

By Jason Way – ISSA member, Northern Virginia, USA Chapter

The purpose of this article is to suggest a PAN encryption solution utilizing existing technology in a manner that is both standardized and interoperable.

Abstract

Following up on an article addressing Personal Account Number (PAN) encryption, this submission expands on some of today's common practices related to payment card industry encryption techniques. While most articles are written explaining the need for PAN encryption or the challenges associated with doing so, this article draws light to existing methodologies that can be utilized to encrypt the PAN. The purpose of this article is to suggest a solution utilizing existing technology in a manner that is both standardized and interoperable. Moreover, if these techniques were employed, the industry could begin encrypting the PAN in a matter of months, not years.

Last issue, Jeff Stapleton¹ discussed numerous challenges for protecting the Personal Account Number (PAN) on credit cards. The article began to draw corollary requirements already in existence for PIN encryption in hopes of sharing lessons learned before embarking upon PAN encryption. In summation, it was suggested that the unique requirements for PAN encryption combined with the industry's complacent attitude about compliant utilization of Triple-DES (TDES) encryption methods, including key-bundling techniques, were a recipe for ubiquitous procrastination. Merchants and service providers alike are all clamoring for PAN encryption. Cardholders around the world are dumbfounded when confronted with the realization that their account numbers are transmitted in clear text. Although the technology and capability to encrypt the PAN has existed for years, cooperation and universal understanding will be required to achieve this next evolution in the payment card industry. Can we take the next step?

The payment card industry is at a crossroads where the cup is either half-empty or half-full. Either the worlds of credit and debit transactions are so uniquely challenged that end-to-end protection will never be achieved, or they are so closely aligned that with some cooperation, end-to-end encryption of all sensitive data can be achieved much more quickly than anticipated.

Most consumers do not truly appreciate or understand the differences between debit card and credit card transactions. Rather, they simply pay with a card that has a major brand logo embossed in the corner. Occasionally, a merchant will

ask if it is credit or debit, and we will respond accordingly. But, there are significant differences. In the U.S., our credit cards are not associated with personal identification numbers (PIN), but our debit cards are. Our debit PINs create an additional level of security and authentication to a transaction. In most instances, a cardholder's signature does not accompany a debit transaction, while most face-to-face credit transactions do require a cardholder signature. Our PINs are sensitive authentication data, and the debit processors require that this authentication data be protected.

The PCI DSS

While the PCI DSS² looms over merchants and service providers on an annual basis, most security or compliance managers are now aware of the infamous "dirty dozen," PCI DSS compliance requirements. We have learned to create access control measures, deploy perimeter security defense, manage our log data, segment our networks, employ a risk-based approach towards compliance, and do our very best to create a "secure network architecture." In fact, we can do all of this with the same level of effort for protecting any sensitive data within our organizations and still have absolutely no idea what occurs during a payment-card transaction. We tick our checkboxes and celebrate when the final compliance report is signed off and submitted. Most U.S. merchants have no cause to participate in or understand the security requirements associated with a debit transaction because key management requirements are outsourced and/or assumed by the service providers processing these types of transactions. As a result,

1 Jeff Stapleton, "PAN Encryption: The next evolutionary step?" *the ISSA Journal* (June 2009).

2 PCI DSS v1.2 (2008), Payment Card Industry Data Security Standard: Requirements and Security Assessment Procedures Version 1.2.

they fail to understand the differences between debit and credit transactions due to relying on these outsourced providers. Third-party service providers handle all of the heavy lifting and provide letters of compliance attestation to the merchants who employ them. In most instances, this is the extent of our merchant's awareness related to PIN security.

In fairness, there are a very small handful of line items in Requirement #3 of the PCI DSS v1.2 that delineate how to manage cryptographic keys. However, there is no mention of what types of keys these are, or what they should be used for. Some merchants will not have a single cryptographic key, while others can have hundreds if not thousands.

When the PCI DSS was established, the main charter was to protect the PAN and ensure that no track data is ever stored. The PCI DSS provides little or no consideration for the security of the PIN. All the while, we knew our most sensitive piece of information, the PAN, was being transmitted in clear text. We live with this phenomenon today mostly resulting from sins of the past. When the first credit card was created, there was no approved format for all to follow. Routing these types of transactions was accomplished on a proprietary basis. As client bases grew, it became financially inadvisable to wipe the slate clean and start over. Decades later, we have recognized the perils and the major brands suggested joining forces to standardize the process. Thus the PCI DSS was formed. PCI tells us that we may retain the PAN, but if we do so, we must render it unreadable. However, in no way shape or form are we permitted to retain any other track data. The PCI Security Standards Council is acutely aware that other standards bodies oversee and provide security for the PIN.

PCI and debit security

Most networks that process debit transactions enforce a requirement known as TG-3.³ TG-3 is for the debit world, what the PCI DSS is for credit. TG-3 is a publication of ANSI ASC X9 and is the de facto certification guideline for PIN security compliance. Consequently, the PCI DSS relies on these other types of compliance mandates for providing that level of protection.

Due to the fact that most merchant organizations have chosen to outsource debit functions and thereby outsource debit compliance, they have not had the opportunity to learn requirements and standard practices associated with encrypting transaction data much in the same way they have learned requirements for credit. In fact, most organizations would probably elect to outsource credit compliance if given the option. In a few short years, merchants and service providers, large and small, have learned how to create secure network architectures to protect cardholder data. The PCI Security Standards Council has created every vehicle imaginable in order to educate the merchant and service provider community with skills and resources to secure any environment where cardholder data may exist. Yet, by relying on these

other standards bodies⁴ for PIN security, these communities have been prevented from learning how to use what are essentially basic fundamentals of card-based transaction cryptography. Now, we are left wondering how we can protect the most commonly abused piece of sensitive information in the payment card industry, the PAN.

Why isn't the PAN being encrypted today?

No one has yet to provide a legitimate argument that would suggest encrypting the PAN would have a negative outcome. Quite to the contrary, numerous articles are written suggesting that the PAN should be encrypted. Major manufacturers are already working on proprietary methods for offering this level of security. Lest we forget, due to the absence of interoperable consideration by early payment card entrepreneurs, the industry as a whole continues to struggle with these mistakes of the past. Accordingly, proprietary methods for PAN encryption will be akin to throwing another log on a fire we are already trying to put out.

A solution that works within the confines of existing formats and existing technologies is the direction the industry needs to pursue. Upon studying such formats and technologies already in place for debit processing, we can locate all that is required in order to overcome the identified challenges of PAN encryption.⁵

Encrypting the PAN

The requirements for PAN encryption are complex as Stapleton noted. Though, when you allow for X9 formats to address the technical minutia, these requirements can be simplified as follows:

1. Need for interoperability
2. Need to isolate certain digits within the PAN for routing purposes
3. Need to utilize an X9 or ISO key management technique
4. Need for key diversification and rotation

Several technologies already in use for debit transactions can facilitate these aforementioned requirements in order to encrypt the PAN. There are two specific practices utilized today that can overcome these challenges. By combining DUKPT with TR-31,⁶ we can eliminate these challenges and succeed in encrypting the PAN.

DUKPT

Every point of sale terminal capable of processing a debit transaction in the U.S. makes use of a methodology known as Derived Unique Key Per Transaction (DUKPT). This methodology allows for a PIN to be protected by a unique

3 TG-3: Guideline for Financial Services TG-3 2006, Retail Financial Services Compliance Guideline, Online PIN Security and Key Management.

4 X9 Accredited Standards Committee X9 Incorporated.

5 Jeff Stapleton does an excellent job of enumerating the requirements and challenges.

6 X9 TR-31 (2005), Interoperable Secure Key Exchange Key Block Specification for Symmetric Algorithms.

Derived Unique Key Per Transaction

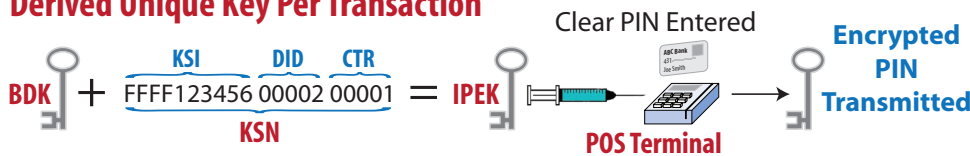


Figure 1 – DUKPT

key every time a PIN-based transaction is processed from a point-of-sale (POS) terminal. Here’s how it works (Figure 1).

BDK – Base Derivation Key: 32 hexadecimal character {0-9, A-F} key, which is randomly generated

KSN – Key Serial Number: 20 hexadecimal characters comprised of three unique parts: Key Set Identifier, Device Identification, and Transaction Counter

- **KSI – Key Set Identifier:** First 10 characters of KSN, which serves as a BDK identifier in payment application systems
- **DID – Device Identification:** Next 5 characters of KSN, which enables 524,287 individual POS devices to be injected with a single key
- **CTR – Transaction Counter:** Last 5 characters of KSN, enabling 1, 048, 575 transactions to occur from each POS device

IPEK – Initial PIN Encryption Key: Cryptographically combining the BDK and KSN will yield the IPEK, which is the key actually injected into a POS terminal. This key is then utilized to derive future “Transaction Keys” – this same sequence can be used for PAN encryption utilizing an interoperable key exchange format

Every time a PIN is entered into a POS terminal, a key is incremented from the base-derived IPEK, and a transaction counter accompanies the transaction back to the host so that the host may understand how much iteration has occurred. Then, the host may logically utilize this information for the purpose of decrypting the encrypted PIN block with the Base Derivation Key for ultimate payment routing and final authorization. The X9.24⁷ DUKPT implementation provides for the creation of an initial derived key that is injected into a point of sale terminal. Cryptographically combining the BDK with a portion of the KSN yields this derivation key (IPEK). Subsequent PIN encryption keys or session keys are then derived for every ensuing transaction such that the transaction key is derived from the previous transaction key and the counter. The PIN encryption key is just a variant of the current transaction key. The PAN encrypting key can be just another variant.

This methodology fulfills both the need to use an X9 key management technique, as defined in ANSI X9.24 part 1, as well as the need for key diversification and rotation. DUKPT can be the method for encrypting the PAN, and based on its complete adoption in the US, we would be utilizing a format

that is already omnipresent, time tested, and proven.

TR-31

DUKPT only solves some of the challenges for PAN encryption. We still need to exchange the

PAN in an interoperable format and account for specific data, which needs to be in the clear for routing purposes. TR-31 was designed for utilizing TDES encryption in a more secure manner. TR-31 is another X9 Report, which demonstrates a commonly accepted guideline for interoperable secure key exchange. TR-31 can be utilized to fulfill remaining requirements for PAN encryption.

TR-31 works like this:

Header	Header (Optional)	Key Length	Key	Padding	MAC
Encrypted Value					XXXXXXXX
MAC					

Header:

Byte #	Field Name	Byte #	Field Name
0	Key Block Version ID	9-10	Key Version Number
1-4	Key Block Length	11	Exportability
5-6	Key Usage	12-13	Number of Optional Blocks
7	Algorithm	14-15	Reserved for future use
8	Mode of Use		

Header (Optional): Can be used to extract required routing data of PAN:

Byte #	Field Name
16-17	First Optional Block ID
18-19	Optional Block 1 Length
20-n	Optional Block 1 Data (Routing Data)
n-m	Additional Optional Blocks, if present

Encrypted value: Would include the following encrypted values

- **Key length:** Can be used to signify exact character count of the PAN
- **Key:** Can be the encrypted PAN as well as the PIN block. This solves both encryption problems and if the encrypted value is a multiple of 32, we can easily migrate to AES as the encrypting algorithm
- **Padding:** Can be used to make a 12-19 character PAN into a 32-character plain text key, which becomes encrypted for transmission by utilizing a DUKPT methodology

MAC: The entire message would be MAC’d for message integrity (MAC is a hash algorithm, which is used to verify the integrity of the entire message; thus, the utilization of the X9

7 ANSI X9.24 Part 1 (2004), Retail Financial Services Symmetric Key Management, Part 1: Using Symmetric Techniques.

TR-31 format allows for interoperability and isolation of the digits required for routing purposes)

Key management challenges

Since the inception of encryption, one ideal has remained the same. The more complex the encryption, the more stringent the requirements are for managing cryptographic capabilities. Compliant key management is a challenge for merchants and service providers alike. Even the most experienced entities struggle with effective controls and procedures for constantly evolving technologies.

Compliant key management is based on principles of dual control and split knowledge. Though, once an organization can make effective use of these disciplines, additional challenges present themselves which are not as easily addressed. Organizations must ensure that keys are used for singular purposes; keys need to be random and unique; key inventories need to be maintained and security parameters need to be established for how different types of keys are to be handled.

TR-31 was developed with these challenges in mind. Currently, keys are constructed, stored, and transmitted in seemingly random blocks of sixteen hexadecimal characters. However, there is nothing attached to a key that dictates its intended usage, or assignment of any security parameter to the key itself. The TR-31 format provides a framework, by way of the message header, that informs a handler of the exact usage and parameters for any given key.

Can debit and credit align?

The worlds of debit and credit transactions exist in dramatically different environments, but they share similar pain points and can leverage one another's experience if a universal understanding would be embraced. Everything from the format of the transaction right down to the people involved is different. Recent headlines and personal inconveniences, however, have finally created enough motivation for the industry to seek a solution. Yet, the concept of split knowledge and dual control has prevented these two worlds from joining forces. The answers to these problems have already been thought through. Education, training, awareness, and advocacy are all that are required to solve this dilemma.

Summary

Everyone is clamoring for PAN encryption. Several manufacturers of payment-related technology have even made strides to provide point solutions without consideration for interoperability. Manufacturers are attempting to create proprietary silos for securing an enterprise. Effective security for a global discipline like buying goods and services with a payment card is something that should be standardized and available to all, regardless of chosen infrastructure. Utilizing the concepts of DUKPT and TR-31, the ability to allow for PAN encryption is upon us. Point-of-sale manufacturers would be required to enable an additional key slot for this encryption, and many have it already, though the formats being used are proprietary and without standardization. A standardized format in combination with point-of-sale manufacturer cooperation will allow for PAN encryption today. PAN encryption is the next evolutionary step for this industry, and the question remains: can we get there?

The purpose of this article was to show that we can. Pressure needs to be applied and solutions need to be demanded from a body capable of enforcing them. Neither developers nor consumers are willing to spend money on stronger security for the common good. Yet, once a standard is in place, where we are all confident that others are doing the same thing (interoperable), we will achieve PAN encryption.

About the Author

Jason Way, CISSP, PCI-QSA, CTGA, is the director of product services for GEOBRIDGE Corporation and is an experienced information security professional and cryptographic key management expert. Jason has worked in a variety of information security capacities relative to the payment card industry and is the GEOBRIDGE representative to ASC X9F6 where subject matter consists of cardholder authentication and integrated chip cards. He provides consultative services on the proper use of DUKPT and EMV protocols. Jason also develops custom command specifications for cryptographic processing functions for host security modules. Contact him at JWay@GEOBRIDGE.net.

