Encryption in the PCI DSS

STATE OF NORTH CAROLINA MERCHANT COMMUNITY

MARCH 22, 2016

Presenter: Joseph D. Tinucci, CTP, QSA, CISSP

Speaker Introduction

Joseph D. Tinucci, CTP, QSA, CISSP

Senior Director of Enterprise Risk & Compliance Platform

- Mr. Tinucci holds a BS Mathematics from Colorado State University. Accreditations held: Certified Treasury Professional (CTP); PCI Qualified Security Assessor (QSA); Certified Information Systems Security Professional (CISSP)
- Over 20 years of experience in payments and security, consulting, treasury management and teaching.
- Managed banking relationships for the University of Colorado as well as card security compliance programs across a large portfolio of diverse merchants.

Coalfire Regional Offices

250 professionals, 10 US offices + Canada, LAC & the UK

Agenda

- The context of the PCI DSS and Encryption
- A few definitions and concepts
- Encrypting data at rest
- Encrypting data in transit
- The SSL / TLS issue
- Tokenization
- Managing service providers' use of cryptography
- Questions?

Context of the PCI DSS and Encryption

- Encrypt wireless used to transmit cardholder data anywhere, any time (Requirement 2.1.1)
- Encrypt remote administrative access to the CDE (Requirement 2.3)
- Encrypt cardholder data at rest (Requirement 3)
  - If you store it you must encrypt it
- Encrypt cardholder data in transit over open/public networks (Requirement 4)
  - If you move it over open/public networks, you must encrypt it
- Encrypt all authentication credentials during transmission and storage (Requirement 8.2.1)

Cardholder Data Environment (CDE)

- Any system that stores, processes, transmits, or can affect cardholder data
  - Anywhere cardholder data is stored
  - Any application that processes cardholder data
  - Any communication with an application (internal or external) or third party across an open/public network to transmit or process cardholder data
  - Any other system that can affect any of the above systems

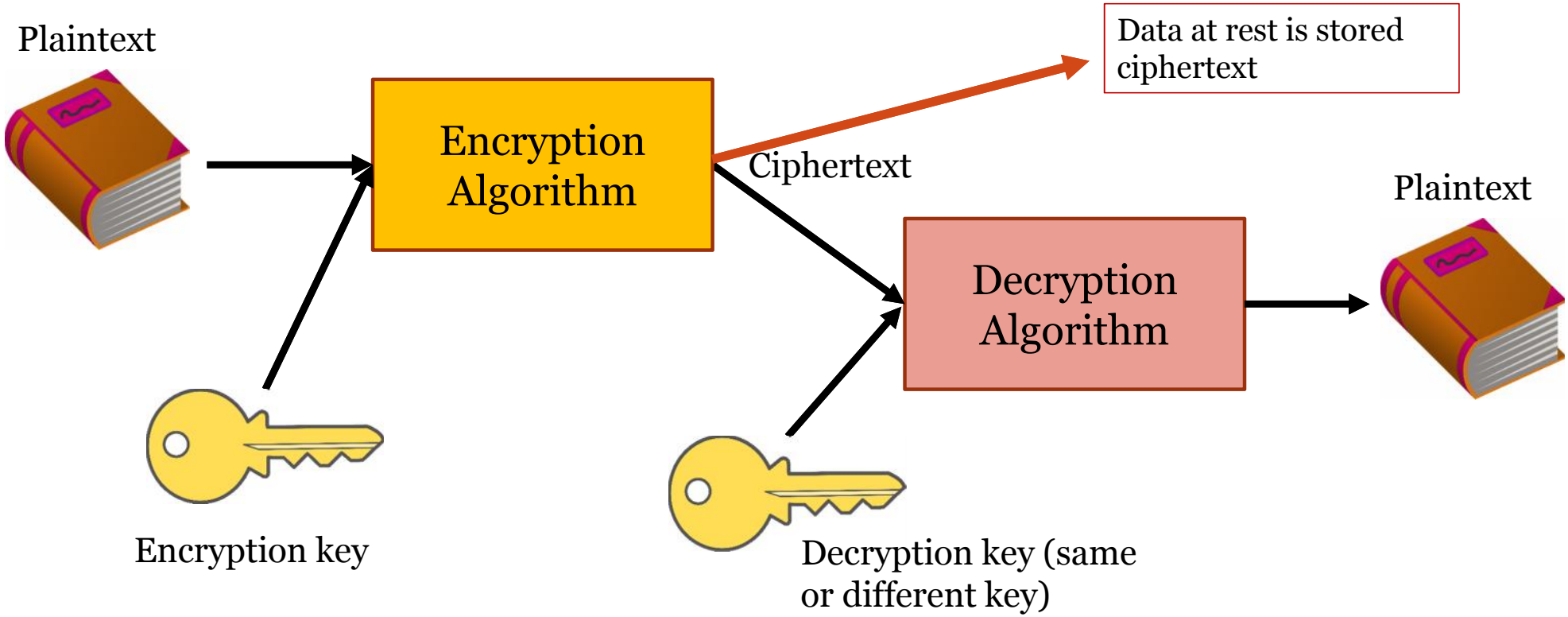
Definition of “Strong Cryptography”

- Industry-tested and industry-accepted algorithms
  - “Roll your own encryption” not allowed
- Proper key management practices
  - NIST Special Publication 800-57, Parts 2 and 3
- Minimum of 112 bits of effective key strength for data at rest
  - NIST Special Publication 800-57 Part 1 for guidance
  - http://dx.doi.org/10.6028/NIST.SP.800-57pt1r4

How Does Encryption Work?

- Data / text / numbers (plaintext) that you want to keep secret gets converted to ones and zeros
- Plaintext plus a large number (key) fed together into an encryption algorithm
- Output is another string of ones and zeros (ciphertext)
- Output (ciphertext) plus a key is fed into a decryption algorithm
- Out comes the secret data / text / numbers (plaintext)

Encryption Process



Symmetric versus Asymmetric encryption

- Symmetric encryption:
  - One encryption key is used by all parties
  - Securely distributing that key to all users/parties is difficult as number of users grows
  - Fast to both encrypt and decrypt
- Asymmetric encryption:
  - Each party has their own key pair – 1 public and 1 private key
  - Public keys are publicly available, private keys are kept secret
  - Something encrypted with the public key can only be decrypted with the associated private key
  - Key management is much simpler as number of users grows
  - Slow to encrypt and decrypt

Encrypting Data at Rest

- **Don't store cardholder data if you don't have a compelling business need!**
- Symmetric algorithms generally used for bulk encryption of data in storage
- Rely on your vendor to provide this capability in their software
  - Industry-accepted algorithms
  - Implementation of industry-accepted key management practices
- Only use PA-DSS validated applications
  - Configured according to the vendor's Secure Configuration / Installation Guide
  - Maintained, patched, and upgraded per vendor's specifications

Encrypting Data in Transit

- Where is data in transit?
  - POI Devices to processor or to tokenizer
  - Web sites – client-to-server communication
  - Server-to-server communication
- Point Of Interaction (POI) devices should encrypt cardholder data at first read/insertion for communication to payment gateway or processor
  - If upgrading equipment, opt for PCI SSC-approved PTS devices
- Web site client-to-server communication
  - Encryption built into browser software
  - SSL/TLS issue will force customers to use modern (updated) browsers
- Server to server communication using secure communication software

Browser Encryption Model

- When "going to" (creating a session with) a secure web page, the communication goes as follows:
  - Handshake, Certificate exchange, key exchange
- Typical ecommerce browser-to-server conversation:
  - Browser says "Here are the encryption algorithms that I know how to use to talk to you securely"
  - Server says "Here are the encryption algorithms that I can use to talk to you securely", and "Here are the encryption algorithms that I decided we will use"
  - Server sends browser its TLS certificate to identify itself and provide its (asymmetric) public key
  - Browser checks certificate against list of trusted certificates
  - If certificate trusted, browser generates symmetric key, sends to server, both use agreed algorithm with symmetric key to encrypt subsequent traffic

What are Digital Certificates?

- A digital certificate is an attestation of identity from a known party
- A Certification Authority issues these attestations of identity to those it examines and for whom it is willing to vouch for their identity
  - Online merchant / storefront, web server, specific application owned by an organization, individual
- Certificate contains the subject's public key
- You can see the trusted certificates within your own browser by examining its certificate store (database of trusted root certificates)

The SSL/TLS Issue

- Secure Sockets Layer (SSL) was the original browser encryption solution (20+ years old now); superseded by Transport Layer Security (TLS)
  - All versions of SSL (2.0, 3.0) are simply not secure, TLS 1.0 is also considered not secure
  - TLS 1.1 is running out on its useful life, TLS 1.2 is current version
  - TLS 1.3 is in the Request for Comment stage
- Hash algorithms are used in conjunction with digital certificates (as part of an cipher suite) to ensure authentication of server/browser
  - Some hash algorithms are simply no longer secure
  - Some are subject to “collisions” that allow forgery of certificate

Why is SSL/Early TLS a Problem?

- The https process allows browsers to request weaker encryption suites than the server would like
  - The server "falls back" to older, weaker algorithms/keys
  - Older encryption algorithms use smaller keys and are easier to "brute force" all possible key combinations quickly
  - Allows bad guys to impersonate the server
  - Allows bad guys to sit in the middle of, and read, all encrypted communications
- Many attacks exist against SSL/Early TLS
  - New attacks discovered every month
  - Important in mitigation plan

SSL/Early TLS Deadlines

- All Acquirers, Processor, Gateways, and Service Providers must provide a TLS 1.1 or greater service offering by June 2016
- All new implementations must be enabled with TLS 1.1 or higher, with TLS 1.2 recommended
  - **You cannot consider consumer web browser limitations as pre-existing infrastructure that needs to be supported**
- All entities must cutover to use only a secure version of TLS (as defined by NIST) effective June 30, 2018
- POI terminals using SSL/early TLS that can be verified as not being susceptible to all known exploits for SSL/early TLS, with no demonstrative risk, can continue to be used beyond June 2018

Migration Plan

- If you are using SSL/early TLS, you need to have a plan to migrate to TLS 1.1/1.2 by June 30, 2018 (sooner is better)
- The plan could include:
  - Upgrading to current, secure version of TLS that is securely implemented
  - Encrypting data with strong encryption BEFORE sending over SSL/early TLS
  - Setting up strongly encrypted tunnel (IPSEC), then sending data using SSL/early TLS through tunnel
- No exception for small merchants except for POI terminals (discussed later)

Mitigation Plan

- If you must use SSL/early TLS, you need to implement (compensating) controls around its use to mitigate the potential impact of compromises
- Description of how vulnerable controls are used
- **Risk assessment of environment and systems and risk reduction controls in place**
- Description of processes in place to monitor for new vulnerabilities and redo risk assessment as appropriate
- Description of change control processes in place to ensure no new SSL/early TLS implemented into new environments
- Overview of migration plan

Implications for ASV Scans

- ASV scans are required to identify SSL/early TLS as a vulnerability
- May be able to pass ASV scan if compensating / mitigating controls in place
- Have a discussion with your ASV scan team
- Need to do your own risk assessment homework
  - Migration plan
  - Mitigation plan

Using Tokenization

- Third party service provider converts card number into “token”
  - Generally, at point of card/chip read in POI device
  - Token used in merchant system in place of actual card number, removes card number from system scope
  - Vendor stores card number for merchant, translates back to card number if/when needed
- Vendor should provide a Tokenization Installation Guide (TIG) for secure deployment, installation and use of tokenization product
- *Tokenization Product Security Guidelines* from PCI SSC is excellent resource

Managing Vendors' Use of Cryptography

- Contracts must mention the PCI DSS and compliance thereto
- Vendor responsibility matrix of who does what
- PA-DSS application vendors MUST provide Secure Implementation Guide
  - Application not necessarily secure out-of-box
  - Must be tuned and maintained according to Secure Implementation Guide
- Ask your POI (terminal) device vendor:
  - Does this device use SSL/early TLS?
  - Is it susceptible to any known vulnerabilities?
  - Can you attest that this device is not susceptible to any known vulnerabilities? **(Must be done every time a new vulnerability in SSL/early TLS is discovered/announced.)**

Questions

For More Information

- Coalfire Systems, Inc.
  - www.Coalfire.com
  - http://www.coalfire.com/Contact
- Joseph D. Tinucci, CTP, QSA, CISSP
  - Joseph.Tinucci@Coalfire.com
  - 303-554-6333 x 7105