RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: ASEC-W04

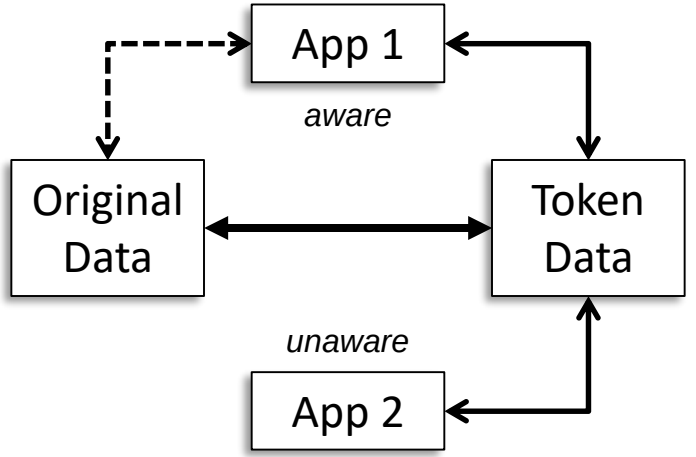
DERIVED UNIQUE TOKEN PER TRANSACTION

Jeff Stapleton

VP Security Architect
Wells Fargo
X9F4 workgroup chair

Application Security

- Solution: tokenization technology
  - Substitute sensitive data for benign data
- Control: benign data is safe
  - Data in storage
  - Data in transit
  - Data in process
- Application interoperability
  - Token aware
  - Token unaware



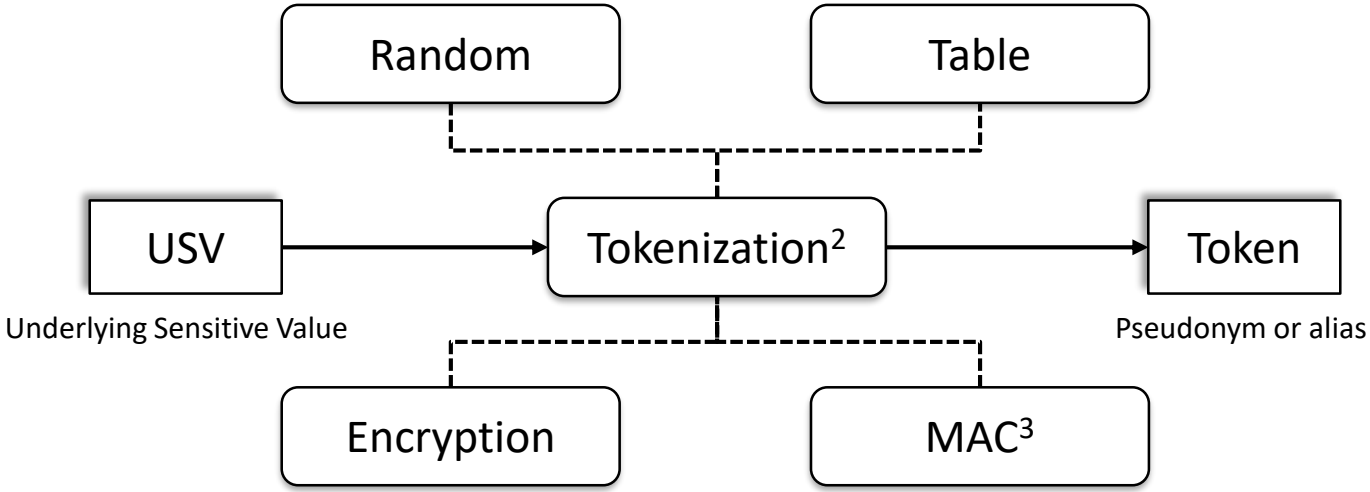
Token Problems

- Tokens data elements not well understood
  - X9 sensitive payment card data tokens www.x9.org
  - EMV payment tokens www.emvco.com
    - Apple Pay, Google pay, Samsung pay
  - PCI post-authorization tokens www.pcisecuritystandards.org
- Tokenization process not well understood
  - Tokenization versus detokenization
- Tokenization systems not well understood
  - Token vaults

WHAT IS TOKENIZATION?

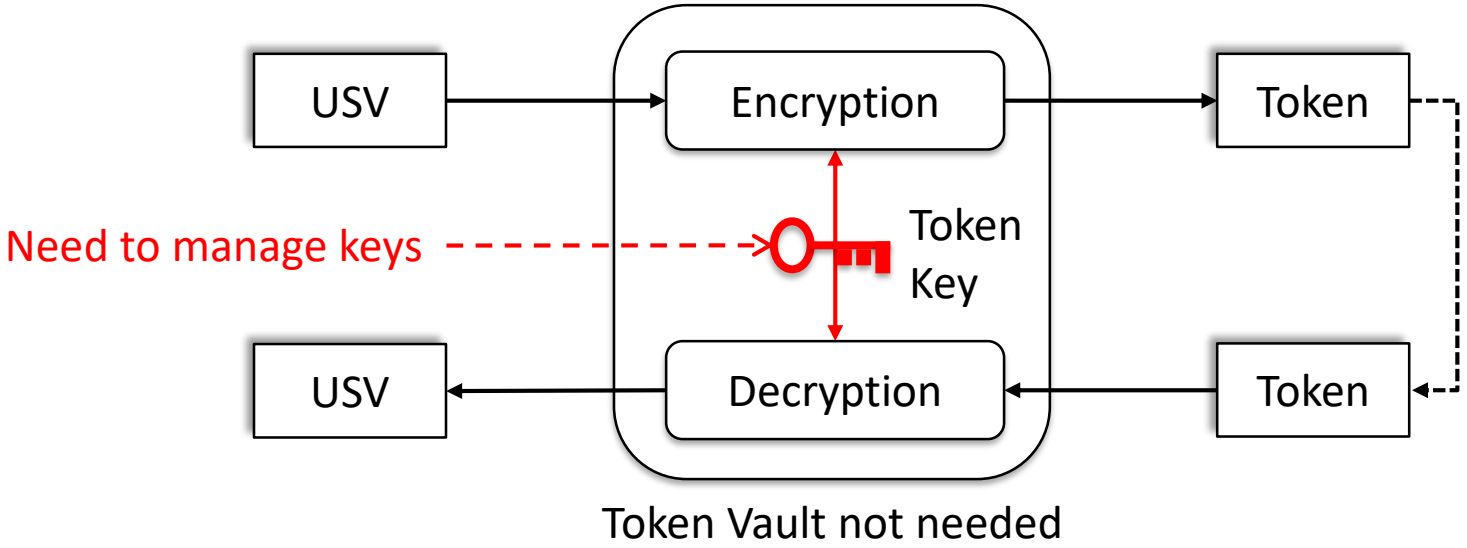
Background Information

Tokenization Defined[1]

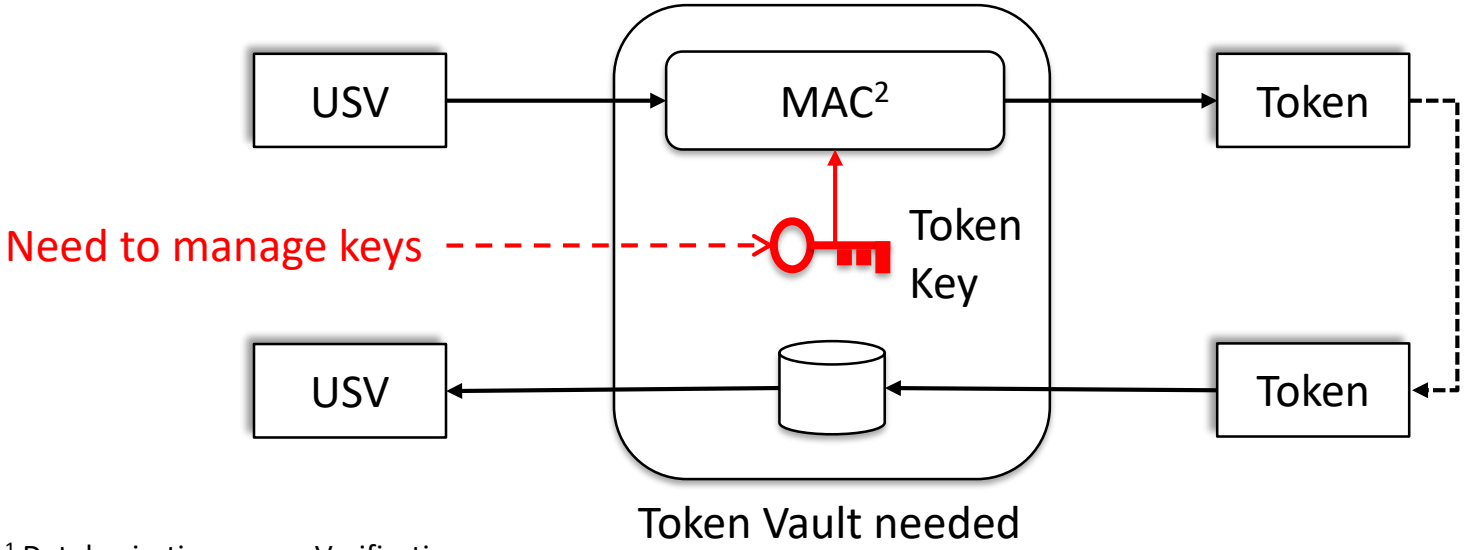


[1] RSA 2017 Conference PDAC-R02 Cybersecurity vs. Tokenization
[2] X9.119 Protection of Sensitive Payment Card Data – Part 2: Post-Authorization Tokenization Systems
[3] ISO 16609 Banking – Requirements for Message Authentication Using Symmetric Techniques

Detokenization: Encryption Method



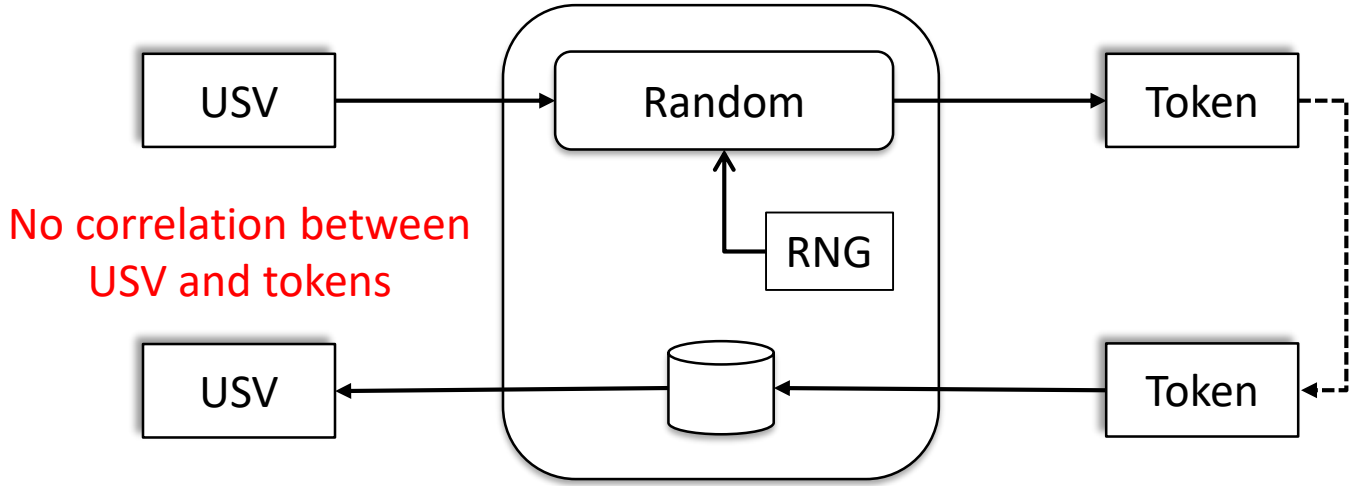
Detokenization: MAC[1]



[1] Detokenization versus Verification

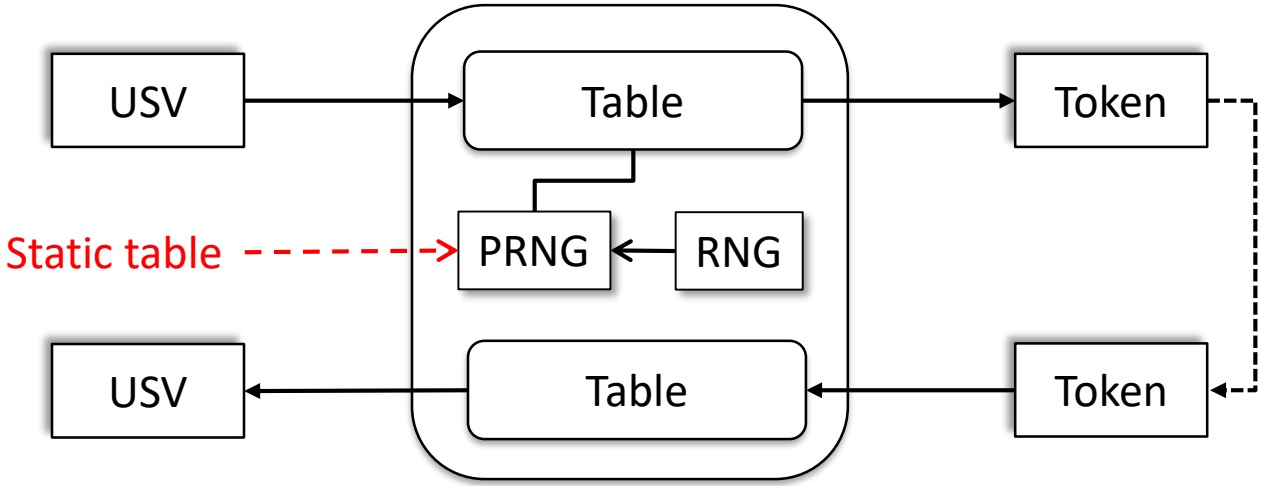
[2] ISO 16609 Message Authentication Using Symmetric Techniques (includes MAC and HMAC)

Detokenization: Random



RNG: Random Number Generator

Detokenization: Table



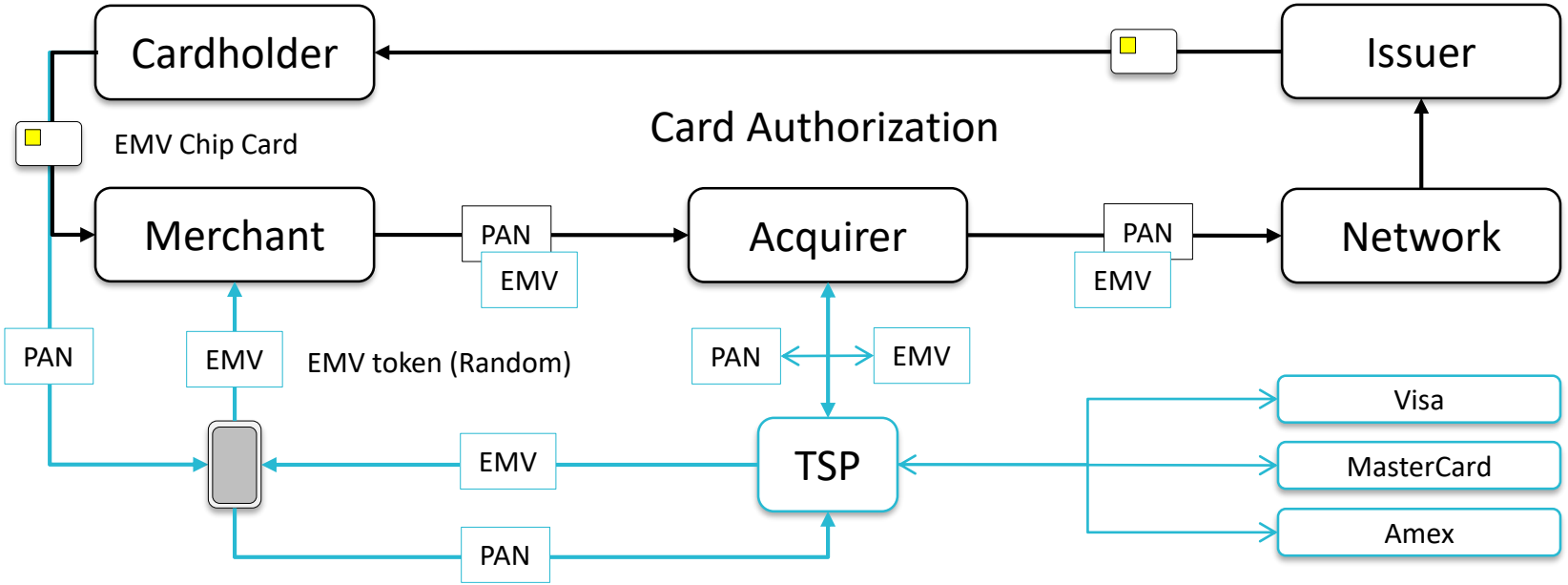
RNG: Random Number Generator
PRNG: Pseudo RNG

Comparison of Methods

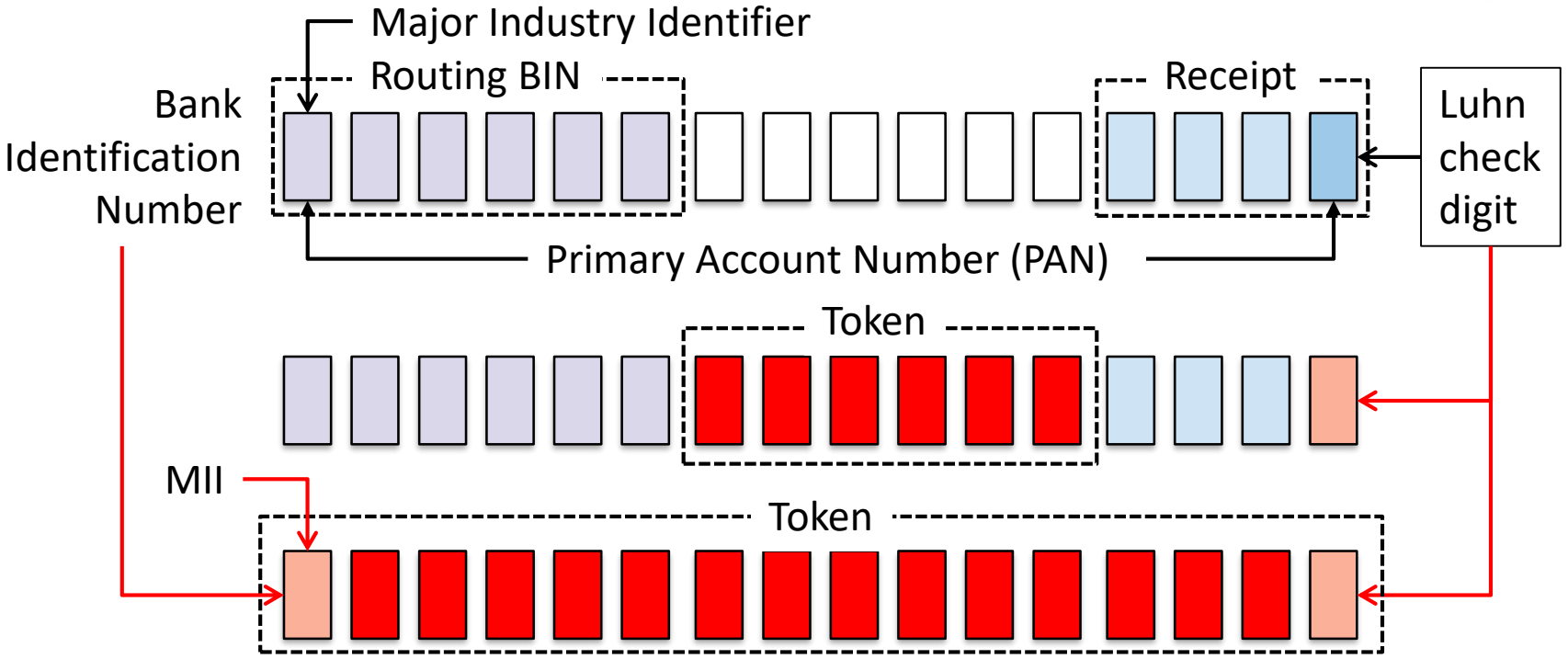
- **Encryption Method**
  - Vulnerable to key compromise
  - Key management
- **MAC Method**
  - Vulnerable to key compromise
  - Key management
  - Vulnerable to vault attack
- **Table Method**
  - Vulnerable to table compromise
  - Table management
- **Random Method**
  - Vulnerable to RNG compromise
  - Entropy management
  - Vulnerable to vault attack

But what about **EMV Tokenization**?

EMV Tokenization

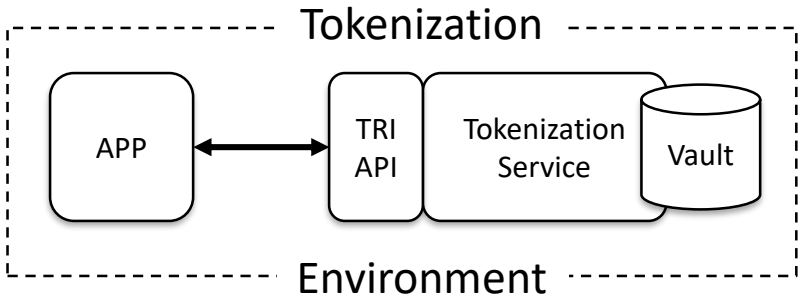


PCI Tokenization



Tokenization Issues

- Token replaces USV to protect it from disclosure or misuse
  - Static tokens have value so might be misused
  - Static tokens might be detokenized
- Token vaults are prime targets
  - Application access controls
  - Network segmentation controls
- Tokenization services
  - Who can get a token, who cannot
  - Who can detokenize, who cannot



So What if?

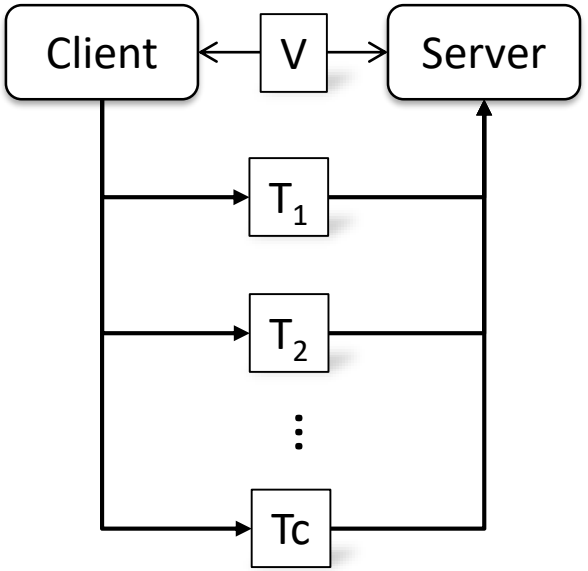
- Each token is used only once
  - No static tokens
  - No detokenization
  - No token vault
- Capabilities
  - Unique token per transaction
  - No residual data
  - Ability to verify token
  - Cryptography based

DERIVED UNIQUE TOKEN PER TRANSACTION (DUTPT)

Background Information

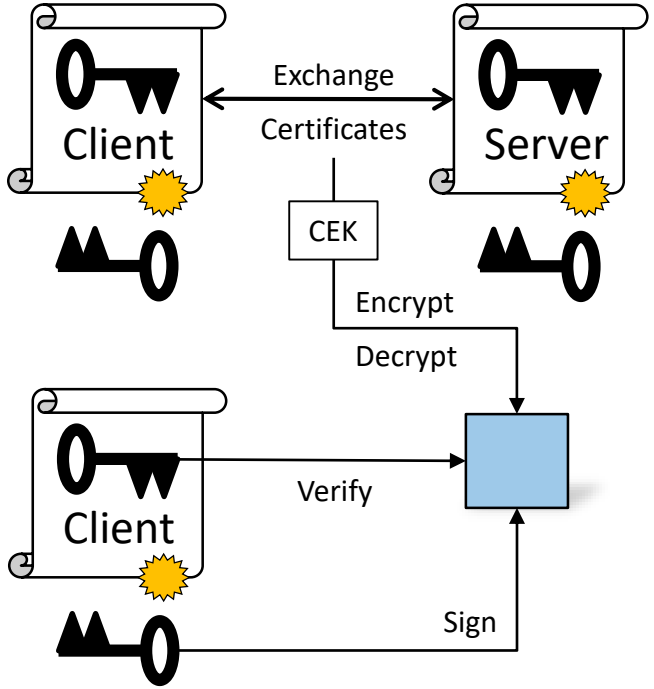
DUTPT Parameters

- Two different one-way functions F(x) ≠ G(x)
- Transaction (c = 1, 2, 3, … max) counter
- Value (V) to be tokenized
- PKI with X.509 certificates
  - CMS-based digital signatures
  - CMS-based encrypted data
- Client has unique identifier (ID)
  - Uses value (V) once then destroys



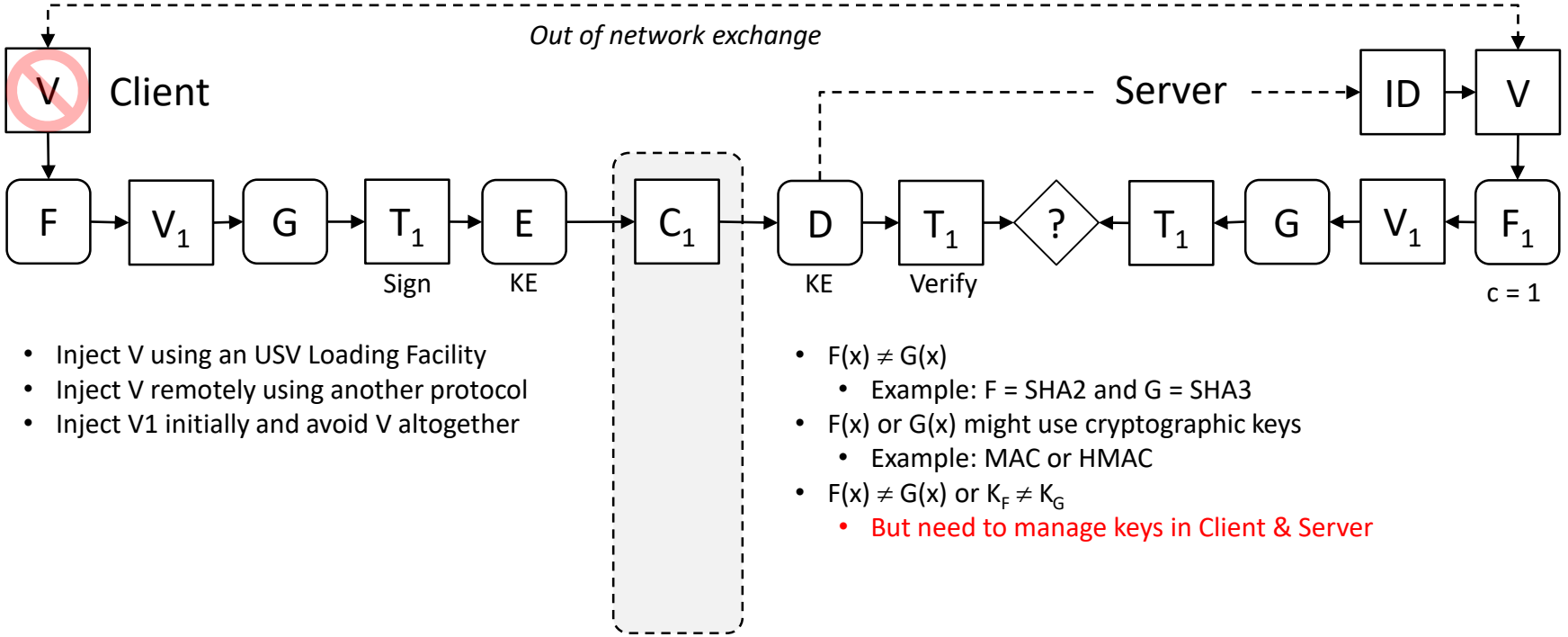
Cryptographic Message Syntax[1] (CMS)

- Signed Data
  - Certificates, Signer Info
- Enveloped or Encrypted Data
  - Recipient Info, Encrypted Content Info
  - Encrypted Content Info
- SignCrypted Data
  - Certificates, Signcrypters



[1] X9.73 Cryptographic Message Syntax – ASN.1 and XML

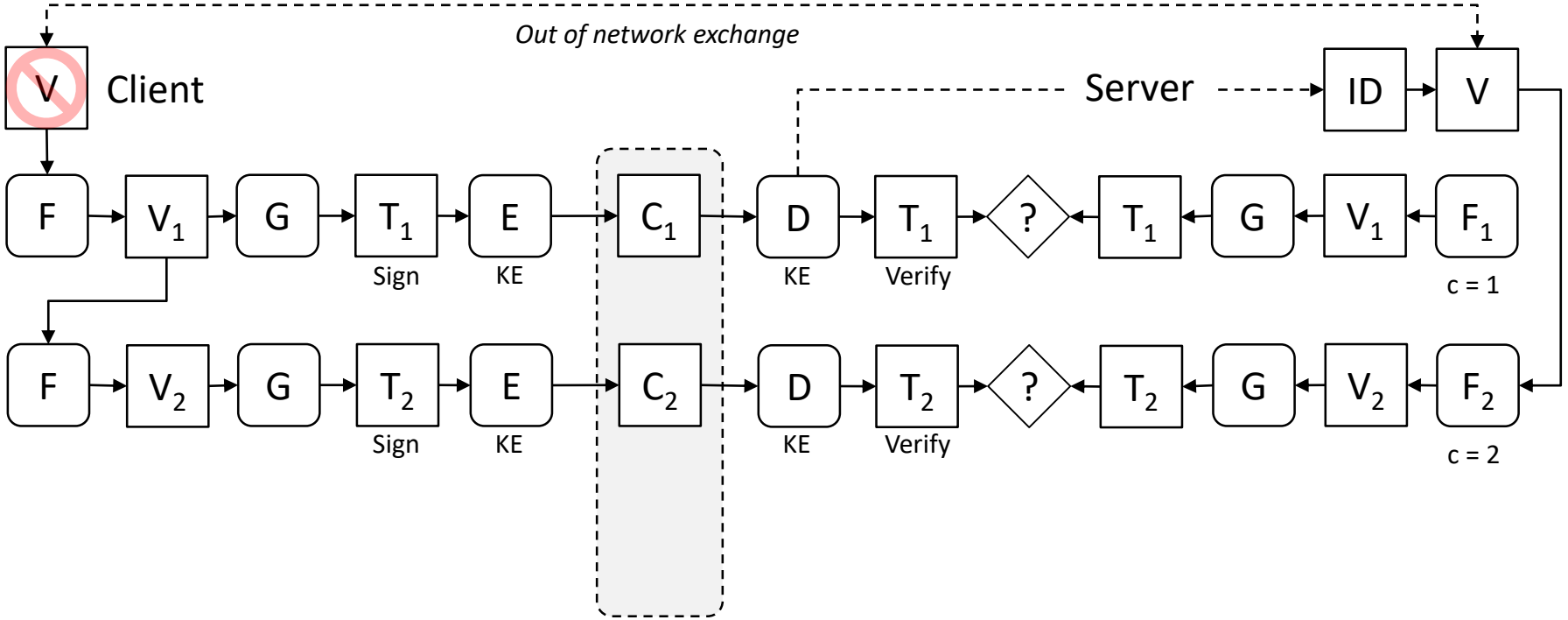
DUTPT Process: x = 1



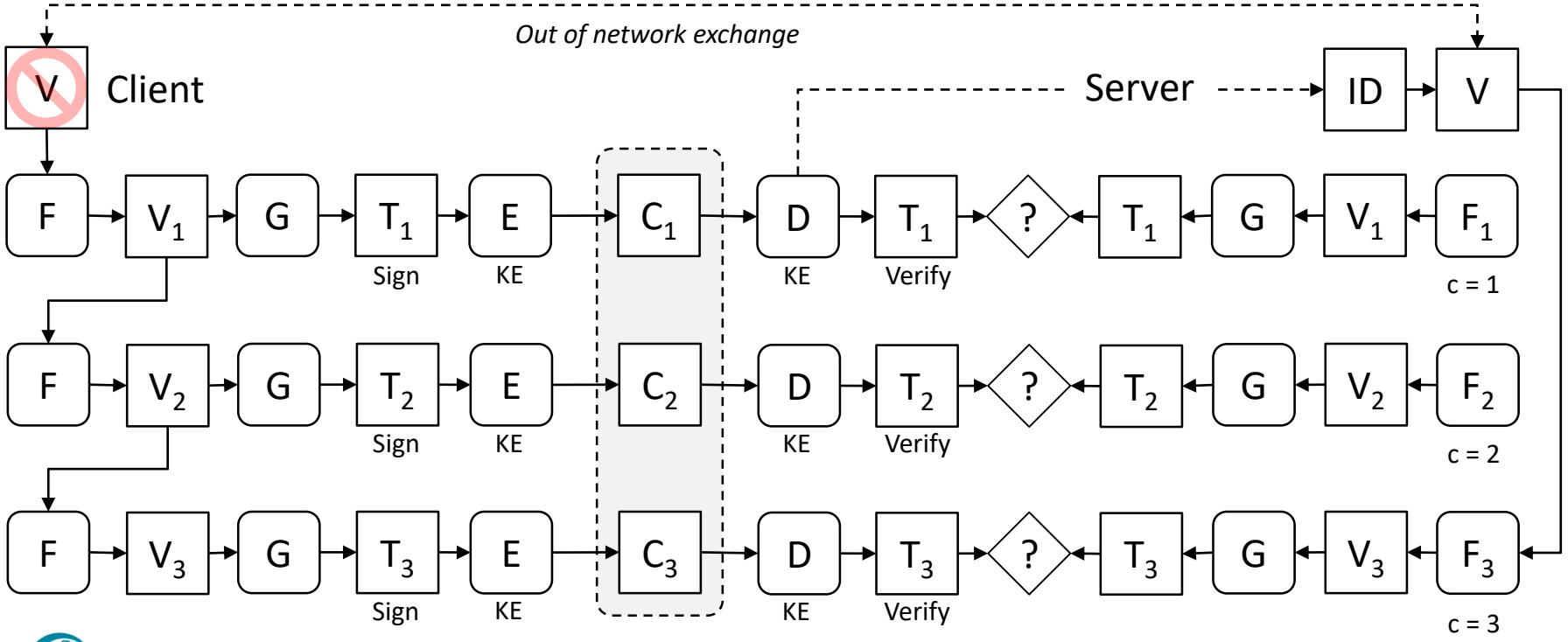
- Inject V using an USV Loading Facility
- Inject V remotely using another protocol
- Inject V1 initially and avoid V altogether

- $F(x) \neq G(x)$
  - Example: F = SHA2 and G = SHA3
- F(x) or G(x) might use cryptographic keys
  - Example: MAC or HMAC
- $F(x) \neq G(x)$ or $K_F \neq K_G$
  - But need to manage keys in Client & Server

DUTPT Process: x = 2



DUTPT Process: x = 3



DUTPT Benefits

- Unique token per transaction
  - Each token (Tc) used only once
  - Next token (Tc+1) not derivable from current token (Tc)
- Derived token using cryptographically sound functions
  - Hash (SHA2, SHA3), MAC or HMAC but $F(x) \neq G(x)$
- Client does not retain value (V) only next value (Vc)
  - Value (V) not recoverable from intermittent values (Vc)
- Suitable for mobile, IoT or *other* remote devices

Conclusions

- Derived Unique Token Per Transaction
  - Conceptual design schema
    - No standards or specification at this time
  - Thinking about application opportunities
    - No software implementations at this time
  - Solution looking for a problem
- Audience questions or comments?
  - Any thoughts, ideas, or interest?

Appendix: References

- International Standards Organization www.iso.org
- American National Standards Institute www.ansi.org
- Accredited Standards Committee X9 www.x9.org
- National Institute of Standards and Technology www.nist.gov
  - Cryptographic Algorithm Validation Program (CAVP)
  - Cryptographic Module Validation Program (CMVP)
- National Information Assurance Partnership www.niap-ccevs.org
  - Common Criteria Evaluation and Validation Scheme (CCEVS)

Appendix: Standards

- Accredited Standards Committee X9 www.x9.org
  - ANSI X9.73 Cryptographic Message Syntax (CMS) – ASN.1 and XML
  - ANSI X9.82 Random Number Generation (RNG) – *multiple parts*
  - ANSI X9.119 Requirements for Protection of Sensitive Payment Card Data – Part 2: Post-Authorization Tokenization Systems
- International Standards Organization www.iso.org
  - ISO/IEC 7812 Identification cards -- Identification of issuers -- Part 1: Numbering system
  - ISO 16609 Message Authentication Using Symmetric Techniques
- Europay-MasterCard-Visa Company (EMVCo) www.emvco.com
  - EMVCo Payment Tokenisation Specification Technical Framework v1.0 March 2014

Appendix: Reading

- Code Breakers: Story of Secret Writing by David Kahn (1967)
- Code Book: Science of Secrecy by Simon Singh (2000)
- Handbook of Applied Cryptography (HAC) by Menezes, van Oorshot, and Vanstone (1997)
- Security without Obscurity by Jeff Stapleton
  - A Guide to Confidentiality, Authentication, and Integrity (2014)
  - A Guide to Public Key Infrastructure (PKI) Operation (2016)
  - A Guide to Cryptographic Architectures (June 2018)

How to apply this session

- One week
  - Determine if your organization uses, or plans to use, tokenization
- Three months
  - Determine your tokenization regime (e.g. EMV, PCI, X9, other)
  - Determine the tokenization method (Encryption, MAC, Random, or Table)
  - Determine if *static* tokens or *dynamic* tokens are useful
- Six months
  - Determine if derived unique token per transaction (DUTPT) makes sense