EMV Payment Security
A Brief Overview
Card-based Payments Environment

Typical Point of Sale Card Transaction

(A) Cardholder
(B) Merchant
(C) Acquirer (merchant’s bank)
(D) Issuer (cardholder’s bank)

Image from http://www.sec.gov/Archives/edgar/data/1141391/000119312508034694/d10k.htm
Card-based Payments Environment

Cardholder Goals
- Receive goods, services
- Keep personal payment credentials secure

Image from http://www.sec.gov/Archives/edgar/data/1141391/000119312508034694/d10k.htm
Card-based Payments Environment

Merchant Goals
- Profit from the sale of goods, services
- Rest assured that regardless of the form of customer payment, will receive $
Card-based Payments Environment

Acquirer Goals
- Profit from offering payment processing services to merchants
- Limit fraud losses

Image from http://www.sec.gov/Archives/edgar/data/1141391/000119312508034694/d10k.htm
Card-based Payments Environment

Card Association / Payment Network Goals
- Profit from movement of money everywhere (interchange fees)
- Limit fraud losses

Image from http://www.sec.gov/Archives/edgar/data/1141391/000119312508034694/d10k.htm
Card-based Payments Environment

Issuer Goals
- Profit from offering a variety of buyer-side banking services to individuals, corporations
- Limit fraud losses

Image from http://www.sec.gov/Archives/edgar/data/1141391/000119312508034694/d10k.htm
Card-based Payments Environment

Fraudster Goals
- Profit

More than one type of card...

In the U.S., magnetic-stripe readers by default
Plaintext account data stored magnetically on the card
Most MSR information also displayed on the card
CVV2 = 2FA for magnetic stripe “card not present” txns

Rest of world largely uses “EMV” chip cards
Based on the Europay Mastercard Visa (EMV) consortium,
ISO 7816 physical definitions

International standards govern terminal, card security
Payment Card Industry (PCI), EMV, Common Criteria (CC)

It’s 1996: enter EMV and “liability shift”

What shifts where?
Financial responsibility for fraud losses shifts from issuers to whichever party (issuer or acquirer/merchant) failed to deploy an EMV solution

Industry arguments:
1: ‘Unclonable’ chip cards that can compute ‘cryptograms’ for card authenticity attestation
2: Personal Identification Number (PIN) for cardholder verification
3: Issuers can configure chip card transaction parameters

Now at scale:
~1 billion active EMV cards, ~15 million terminals

References: [1]
How prevalent is EMV?  
And where?

<table>
<thead>
<tr>
<th>Region</th>
<th>EMV Cards</th>
<th>Adoption Rate</th>
<th>EMV Terminals</th>
<th>Adoption Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada, Latin America and the Caribbean</td>
<td>207,715,356</td>
<td>31.2%</td>
<td>3,900,000</td>
<td>76.5%</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>336,602,681</td>
<td>27.9%</td>
<td>3,480,000</td>
<td>43.0%</td>
</tr>
<tr>
<td>Africa and the Middle East</td>
<td>233,003,747</td>
<td>17.6%</td>
<td>345,000</td>
<td>60.7%</td>
</tr>
<tr>
<td>Europe Zone 1 (SEPA countries)</td>
<td>645,472,323</td>
<td>73.9%</td>
<td>10,500,000</td>
<td>89.0%</td>
</tr>
<tr>
<td>Europe Zone 2</td>
<td>27,516,286</td>
<td>12.7%</td>
<td>513,600</td>
<td>65.4%</td>
</tr>
<tr>
<td>United States</td>
<td>Not reported</td>
<td>Not reported</td>
<td>Not reported</td>
<td>Not reported</td>
</tr>
<tr>
<td>Totals</td>
<td>1,240,310,393</td>
<td>40.1%</td>
<td>18,738,600</td>
<td>71.1%</td>
</tr>
</tbody>
</table>

Note: Figures reported in Q1 2011 and represent the latest statistics from American Express, JCB, MasterCard and Visa, as reported by their member financial institutions globally. Figures do not include data from the United States.  
Source: EMVCo, LLC

What about fraud rates?

(they meant millions, not billions)
What properties to verify during a transaction?

Authenticity of payment card
Attestation that card is legitimate

Presence of payment card
More on this later

Cardholder presence, intent
Attestation that account owner intends to conduct txn

Availability of funds
Confirmation that account funds or credit line sufficient

Managed risk
Assurance that behavior is approved by issuer
What makes verification difficult?

**Cost**
Merchants must purchase terminals
Issuers must provide millions of cards
These are large expenditures

**Power and performance**
Not just terminals that need to run crypto - cards too

**Size**
Mobile Point-of-Sale systems increasingly common, impose additional requirements on designers

**User experience**
Anti-fraud mechanisms can degrade usability

**Network distribution and access**
Cards, terminals widely, globally distributed, long roll-out periods
Attackers can easily obtain terminals and cards for vulnerability discovery, often have physical access in exploit scenarios
Offline Data Authentication:
Static, Dynamic, or Combined Data Authentication (SDA, DDA, CDA)

Cardholder Verification:
“Enciphered” PIN incorporated into online mode as well as one offline mode

Card Action Analysis:
Card signs transaction information to be sent to issuer, issuer responds with signed data

Application Cryptogram:
Card cryptographically certifies its decision on the transaction (both accept and decline)
Verifying Card Authenticity
Static Data Authentication (SDA)

Card maintains list of Certificate Authority Public Keys
These CAPKs are used to authenticate cards’ issuer certificates. Some are still 1024-bit RSA keys.

SDA provides a static verification mechanism
Terminal can verify:
- Card’s issuer certificate is signed by an unrevoked, legitimate CAPK
- Card’s static data blob is signed by the issuer

No replay protection
An attacker who observes this data once can “clone” the SDA capability over the card
Verifying Card Authenticity
Dynamic Data Authentication (DDA)

This time in addition to issuer certificate, card-specific key verified
This certificate is signed by the issuer

Terminal chooses an ‘Unpredictable Number’ (UN)
32 bits in length. This is added to other data in a Data Objects List (DOL), sent to the card

Card hashes data with SHA1, signs hash using private RSA key
Terminal verifies this to complete the authentication

Why a signature scheme like this?
Think about how to represent a long message..
Cardholder Verification Methods

Offline Enciphered PIN (Card verifies PIN)

Card has separate PIN encipherment certificate
Verified through issuer-CA chain, as before

This time, card generates a random nonce
64 bits in length, sent to terminal

Terminal generates its own random, pads message, encrypts with card's RSA public key
Rest of the message is header, PIN, card's nonce

Card decrypts, checks nonce is the same
Then, can verify the PIN against internal storage
Cardholder Verification Methods

Online Enciphered PIN (Issuer verifies PIN)

Terminal can send entered PIN to acquirer
Encrypted with 2-key Triple-DES, in ISO PIN block format

But it’s not that simple
How does the terminal know the acquirer’s TDES key?
Could the terminal share a key with the issuer?
If not, how are keys established between acquirer and issuer?
Are the keys static?

Solution: extensive use of HSMs (e.g. ‘payshield 9000’)
Physically-secure, tamper-detecting module use for key storage and cryptographic operations

$ Application Cryptogram $
Hardware Security Modules
And the difficulty of a flexible-yet-secure API

HSMs need to perform a wide range of functions
Cryptogram generation, PIN block translation, key export...
Key export example:
- card and issuer HSM currently share key $K_i$
- want to roll to $K_{i+1}$

APIs sometimes do terrible things in the name of flexibility [2]
IBM Common Cryptographic Architecture key export also allowed key **extraction** by a third party with access to API

Cashing Out
Acronym soup: ARQC, ARPC, TC ...

Authorization Request Cryptogram (ARQC)
Generated when online authorization required
Card computes TDES-based MAC on transaction data

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount, Authorised (Numeric)</td>
<td>Terminal</td>
</tr>
<tr>
<td>Amount, Other (Numeric)</td>
<td>Terminal</td>
</tr>
<tr>
<td>Terminal Country Code</td>
<td>Terminal</td>
</tr>
<tr>
<td>Terminal Verification Results</td>
<td>Terminal</td>
</tr>
<tr>
<td>Transaction Currency Code</td>
<td>Terminal</td>
</tr>
<tr>
<td>Transaction Date</td>
<td>Terminal</td>
</tr>
<tr>
<td>Transaction Type</td>
<td>Terminal</td>
</tr>
<tr>
<td>Unpredictable Number</td>
<td>Terminal</td>
</tr>
<tr>
<td>Application Interchange Profile</td>
<td>ICC</td>
</tr>
<tr>
<td>Application Transaction Counter</td>
<td>ICC</td>
</tr>
</tbody>
</table>

Table 26: Recommended Minimum Set of Data Elements for Application Cryptogram Generation

Application Cryptogram

Image from [1]
Cashing Out

Acronym soup: ARQC, ARPC, TC ...

Authorization Response Cryptogram (ARPC)
Sent by issuer when online authorization requested
TDES-based MAC, but authentication data opaque to terminal

Transaction Certificate (TC)
Generated by card, effectively a card-signed (RSA) log of transaction
Needed by acquirer to collect $!
Untrusted Intermediary
What happens if it's between ICC and terminal?


Why might a cardholder care?
How is the transaction amount communicated to the card?

Can cards authenticate terminals?
What are the challenges involved?

EMVCo discussing proposed ECC-based key-establishment between card and terminal [3]
Blinded Diffie-Hellman. Why the blinding factor?
Relay Attacks
Humans are usually the weakest link

Image from [3]
Pre-play Attacks

What if a weak RNG is used to generate the Unpredictable Number?

What about a REALLY weak RNG?
Details in [5], let’s discuss on whiteboard
securing $ with crypto, subject to real-world constraints

= real-world problems
External References


[5] Mike Bond; Omar Choudhary; Steven J. Murdoch; Sergei Skorobogatov; Ross Anderson. “Chip and Skim: Cloning EMV cards with the pre-play attack.” 2012.