Smart Card Alliance

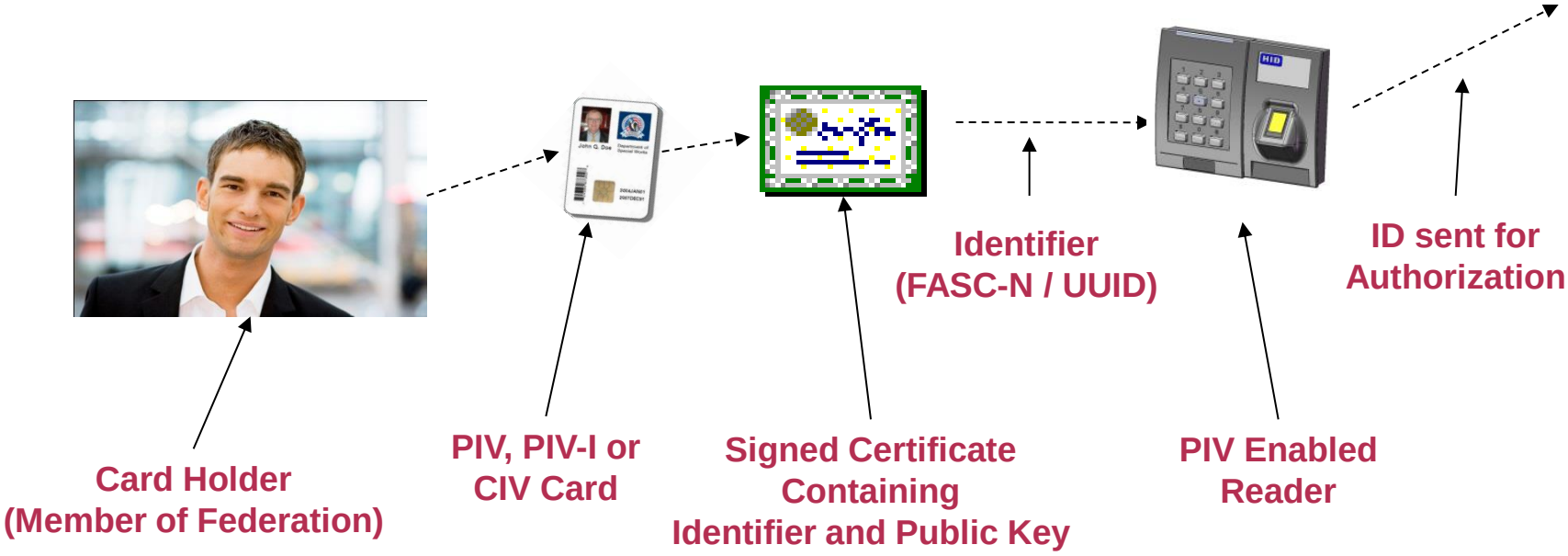
PIV Authentication Mechanisms Used in PACS

Bob Dulude, PhD
HID Global

Topics Covered

- Why Authentication is Important
- Mitigating PACS Vulnerabilities Using Authentication
- Why PKI Based Credentials; Symmetric vs Asymmetric keys
- PIV Card Data
- Understanding Digital Signatures
- PKI Certificate Validation
- Challenge Response Functions
- Comments on Performance of PIV Authentication Mechanisms

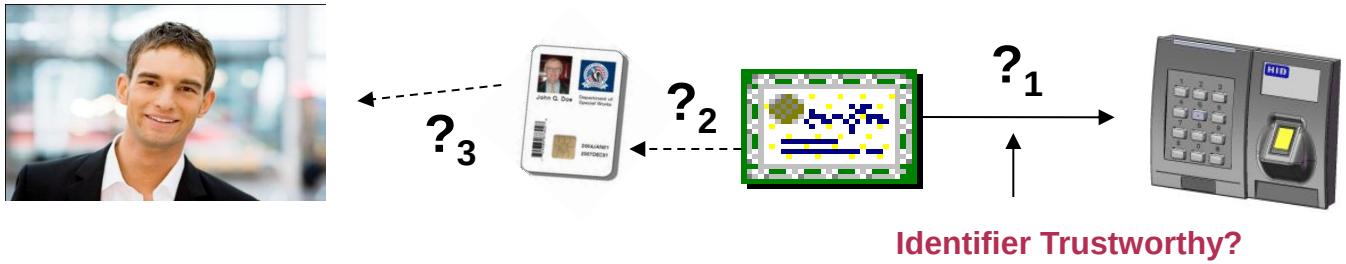
What does it mean to “authenticate”?



Can we trust the ***identifer?***

Smart Card Alliance

What are the vulnerabilities of any PACS?



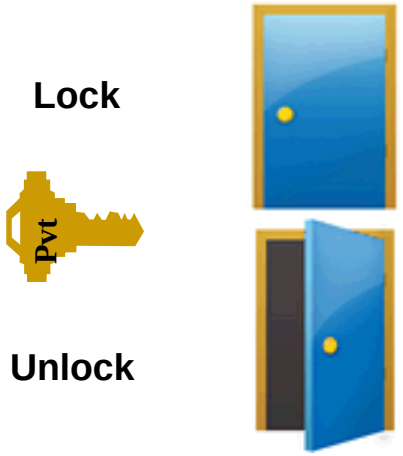
1. Counterfeit identifier
   - Digital signature by a trusted source ensures signed data object is genuine and unmodified
2. Copied or Replayed identifier (certificate only)
   - Private key challenge ensures identifier has not been copied or replayed
3. Lost/stolen/shared card – cardholder does not match identifier
   - Check binding of the identity card to individual by checking either/both
     - Something you "know" (PIN)
     - Something you "are" (biometric)
4. Revoked credential – Cardholder relationship with issuer is "broken"
   - Periodic check that credential has not been revoked

Symmetric vs Asymmetric Keys

Symmetric Key

- Locks
- Unlocks
- Must be kept private
- Same key for every user

Pro – fast for encryption
Con – one key compromised all keys compromised
Con – not generally useable for digital signing



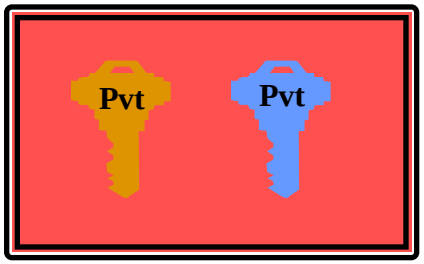
Asymmetric Key Pair

- Keys are digital
- One key encrypts; can be public
- Separate key decrypts; must be private
- Unique key pair for every PIV card holder

Pro – one key compromised, others unaffected
Pro – ideal for digital signing
Con – slower for encryption

Smart Card Alliance

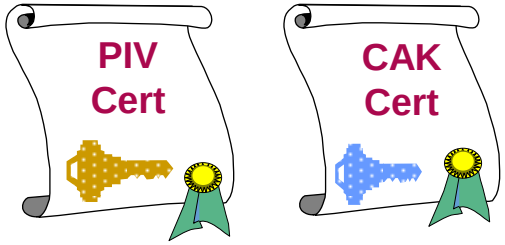
Inside the FIPS 201 Card



Private Key "Vault"

Private Keys

- Cannot be extracted from vault
- Used for digitally signing
- "PIV" key requires PIN to be used



Cardholder Certificates

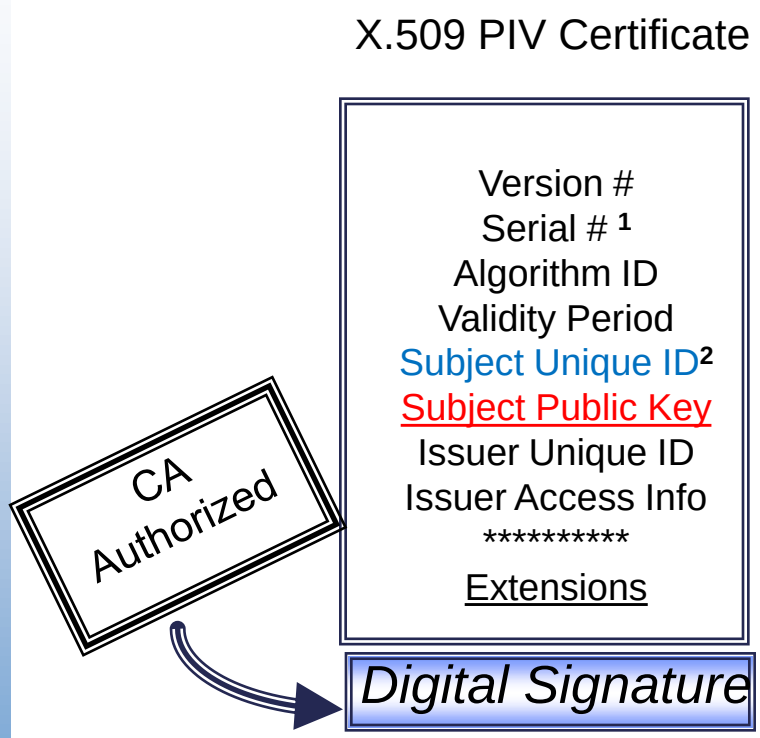
Digitally Signed Data Objects

FIPS 201 Card Data

Smart Card Alliance

What is a Digital Certificate?

Identity information authenticated by a trusted third party; Certification Authority



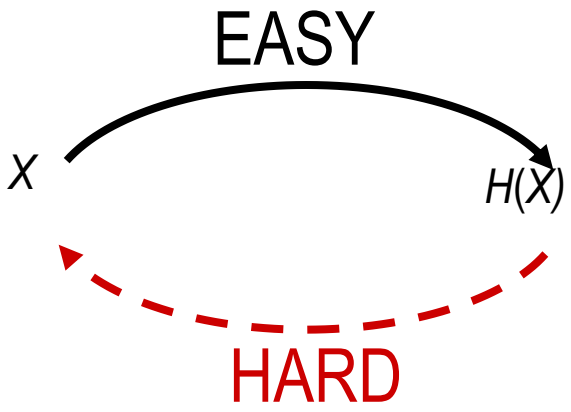
- Signed by CA's private key
- Verified by CA's public key
- Signed data, can't be altered
- Public information, not secret
- Binds identity to public key
- **Provides for secure distribution of subject's public key**

1) Unique Integer
2) E.g., FASC-N or UUID

Smart Card Alliance

Hashing Function

- A hashing algorithm converts one input value X into another *H(X)*
- The input X can be any length
- The output *H(X)* is fixed
- *H(X)* is easy to compute
  - 10,000 times faster than signature



- Hashing is a "one-way" function
  - It is hard to extract X from H(X)
    - If H(X) is 20 bytes = 160 bits, then there are 2^{160} input values to try
    - Even at 1 trillion hashes/sec, takes 10^{28} years to try all >> lifetime of universe
- *H(X)* is unique for each X
  - Probability of two X values that will give the same *H(X) ~ zero*

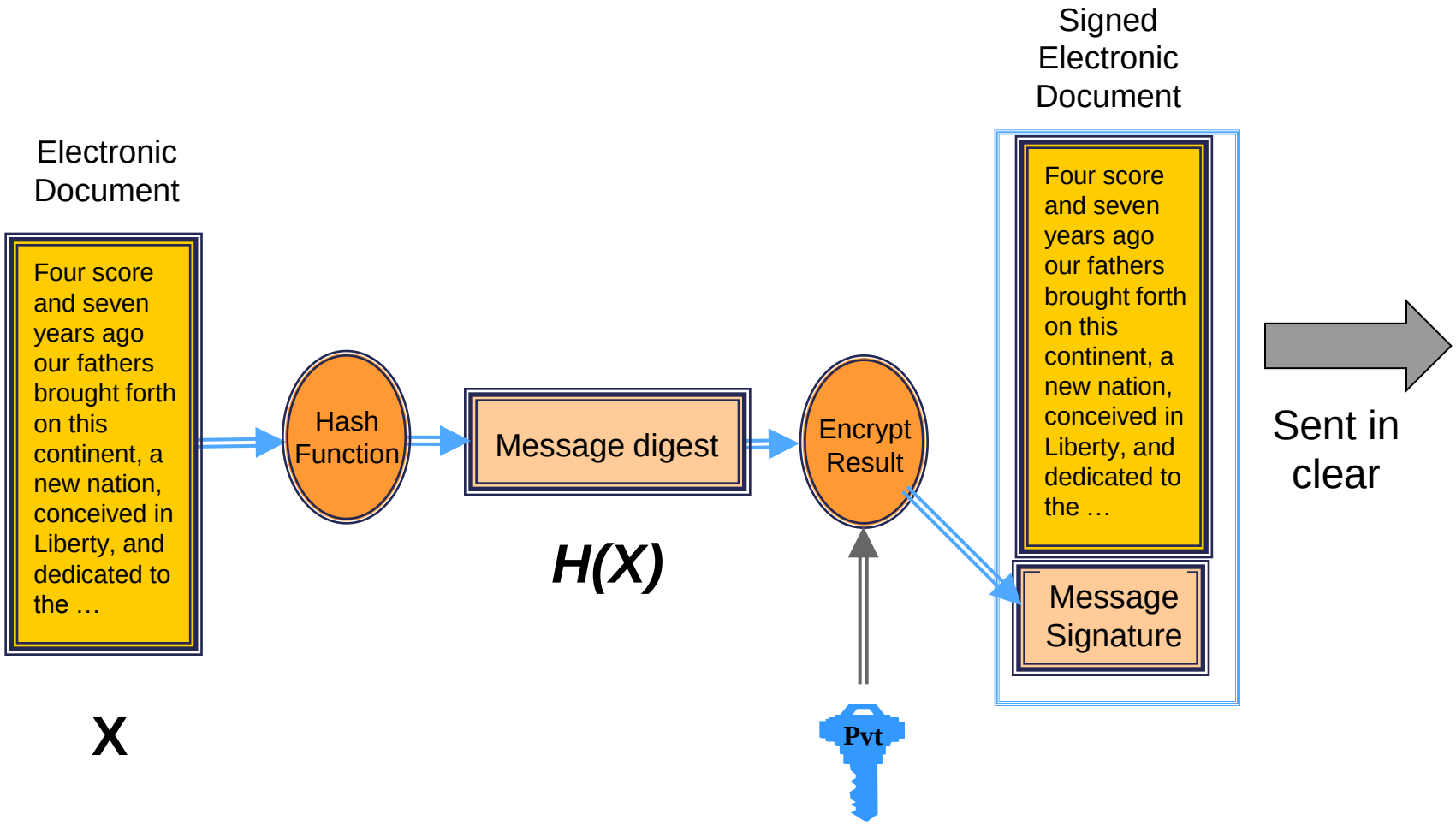
Hashing Examples

Input	MD5 Hash (128-bit output)	SHA-1 Hash (160-bit output)
World Clocks are Nice	d41d8cd98f00b204e9800998ecf8427e	4c2731850527f7ba97e0c31026bdbbecd8801545
World Clocks Are Nice	3b7a7b4bb91ce3a25adab34a9d2533a4	F60f3155885566bd2c5b54069f6dc018a6406710
5	e4da3b7fbbce2345d7772b0674a318d5	Ac3478d69a3c81fa62e60f5c3696165a4e5e6ac4
MyPassword9$	F696d1859228db7f8f54743f649d9810	B7b98b3c68696c316dd69a0942c2246a2d0b01a5
Contract.ppt 7,234 KB	95b3554a88160fed8834214fff502ff1	A4dwon34DASD976ojajdcBud7928a89ql3lo90no

SHA-2 is set of hashing functions; SHA-224, SHA-256, SHA-512, etc.

Smart Card Alliance

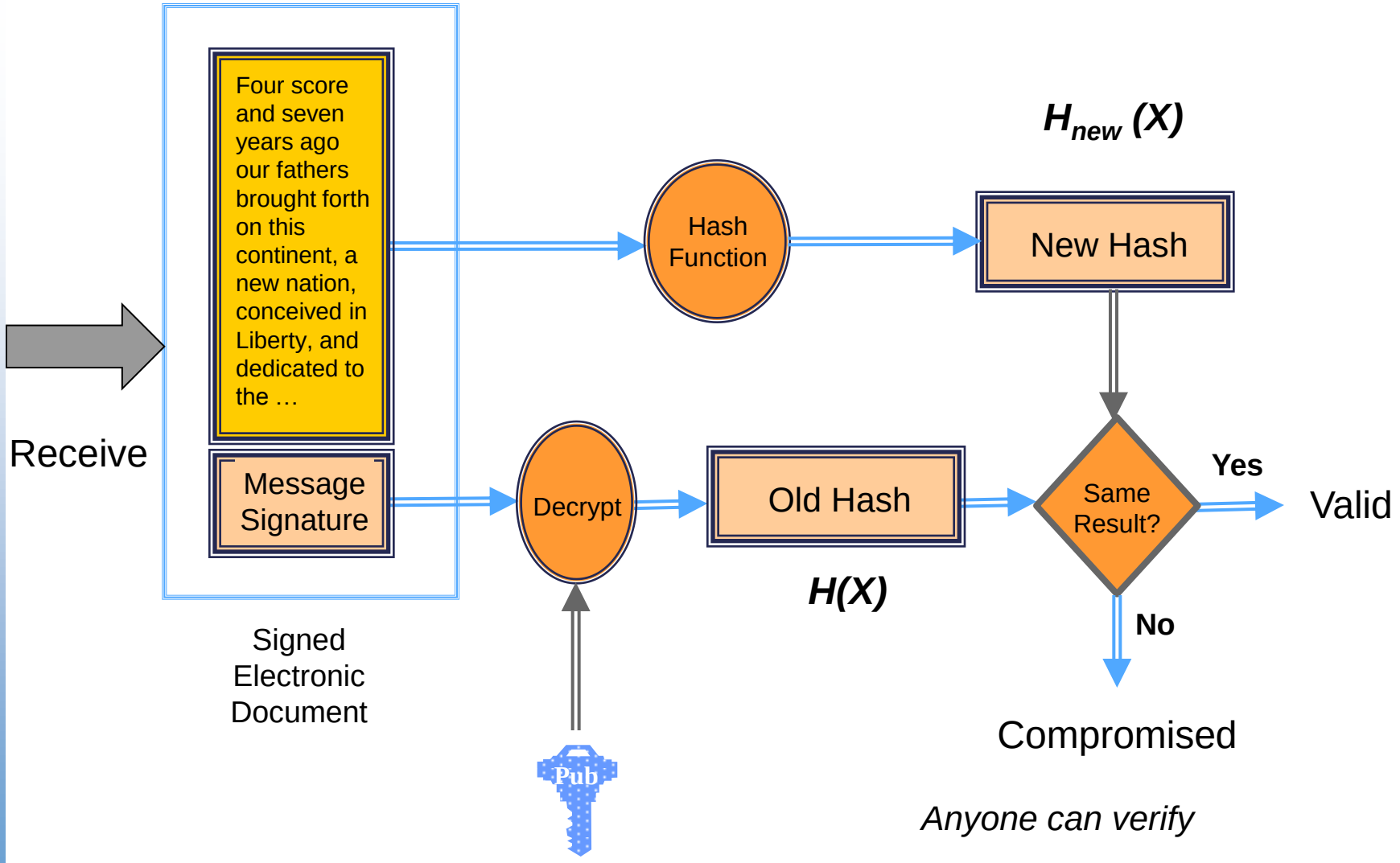
Creating a Digital Signature



Only Private key holder can sign

Smart Card Alliance

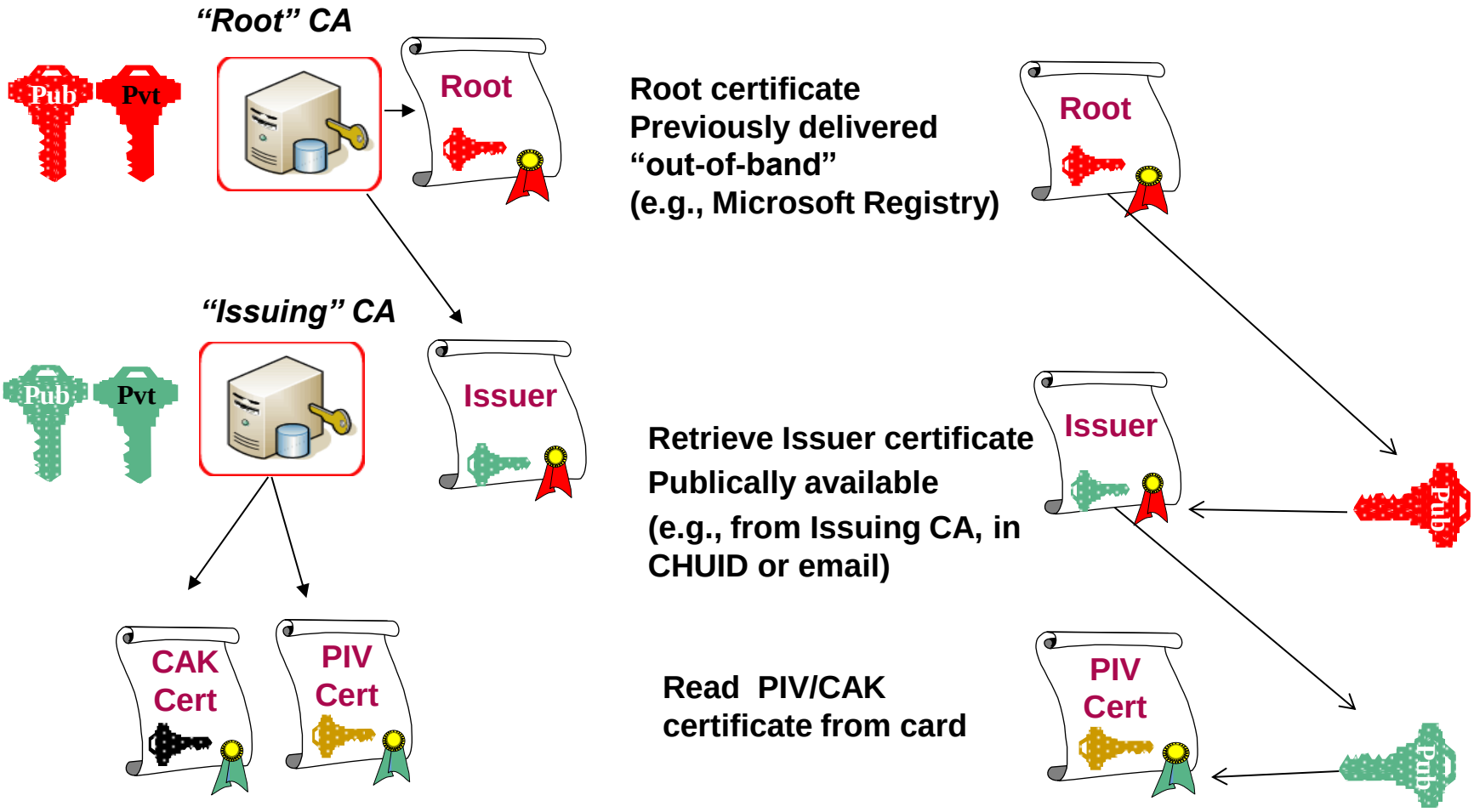
Verifying a Digital Signature



Note – if symmetric key is used then decryption key is same as encryption key

Smart Card Alliance

Establishing Trust in a Certificate

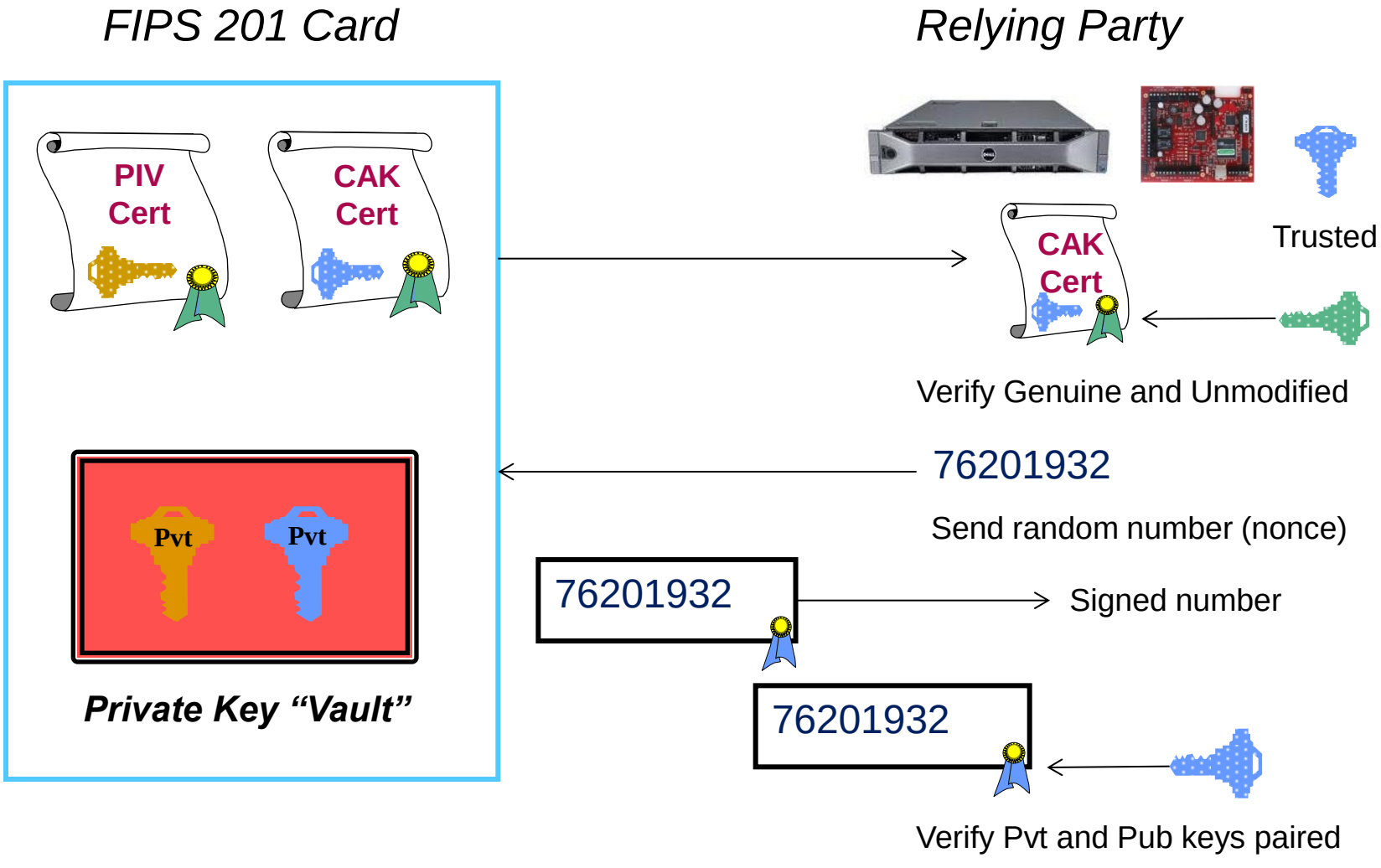


Certificate Hierarchy

Where do the Certs Come From?

How is Trust Established?

Executing Private Key Challenge



Was this certificate issued to this card?
PIV key challenge requires PIN, otherwise same

Authentication Modes & Assurance Levels

Auth Modes ↓	Revoked	Counterfeit or Altered	Copied or Replay	Lost or Stolen	Shared	Auth Factors	SP 800-116 Security Area Definitions
Legacy & FASCN readers	Local Only					0	Uncontrolled (No Auth)
~~CHUID+VIS~~	**via Cert**	✓				1	Controlled (Affiliated)
CAK	✓	✓	✓			1	Controlled (Affiliated)
PIV+PIN	✓	✓	✓	✓		2	Limited (Role)
PIV+PIN+BIO	✓	✓	✓	✓	✓	3	Exclusion (Individual)

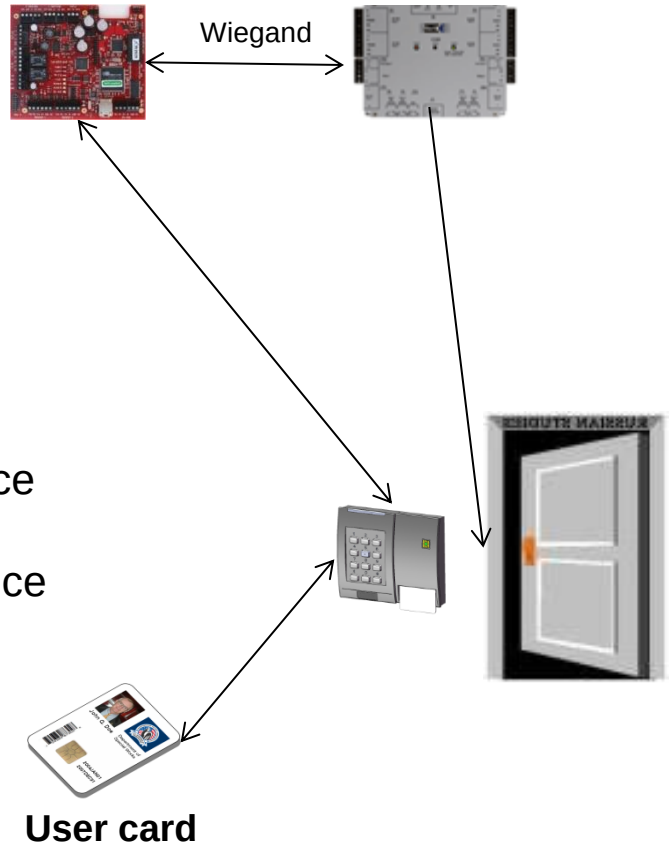
Smart Card Alliance

PKI Authentication Takes Time ~ 4 sec

Why does it take so long?

Many Contributing Factors

1. Functionality spread over multiple devices
2. A lot of data is transferred between devices
3. Crypto processing times proportional to key sizes
4. Not all PIV cards provide same level of performance
5. Not all controllers provide same level of performance
6. User impatience



Smart Card Alliance

Key Factors Affecting Performance

- Crypto processing times increase with increasing
  - Key length
  - Certificate size
- Transfer times proportional to size of data elements
  - RSA Certificate >= 1856 bytes (from 800-73-4)
  - Identifier = 200 bits
    - Wiegand transfer speed limited by standard
      - 400 ms to send 200 bit identifier
- Component manufacturer
  - PIV card performance can vary by 600 to 900 ms by manufacturer
  - Controller performance can vary by up to a full second by manufacturer
- User impatience
  - Process restarts if cardholder removes card prior to completion
- Card Crypto Self-Check
  - ~ 400 ms

How to Improve Performance

- Use faster cards
  - Best card is ~ 600 ms faster than next best card
- Move the authentication functionality into the controller
  - Will save 400 ms by eliminating the Wiegand transfer process
- Replace RSA with ECC algorithms and keys
  - Key size decreases from 2048 to 256 bits
- Optimize the card's crypto self-check
- Use OPACITY's Zero Key Management (ZKM)
  - Certificate size decreases from X.509's 1856 bytes to CVC's 601 bytes
  - Data processing times decrease from ~2600 ms to ~ 800 ms
  - Expected total transaction time < 2 seconds

Potential Improvement Using ECDSA

- Measured card processing times (using 3rd party card)
  - RSA 2048 bit signature execution time = ~ 1130 ms
  - ECDSA 256 bit signature execution time = ~ 470 ms
- Card processing time savings using ECDSA = ~ 660 ms
- Assume signature verification savings is about same; ~ 660 ms
- Estimated total time saved moving to ECDSA ~ 1.3 seconds
- Expected ECDSA total transaction time ~ 2.7 seconds

Bob Dulude Ph.D.
Director HID Federal Identity Initiative

Mobile: +1 781 710-0436
Office: +1 781 591-0913
Email: bdulude@hidglobal.com