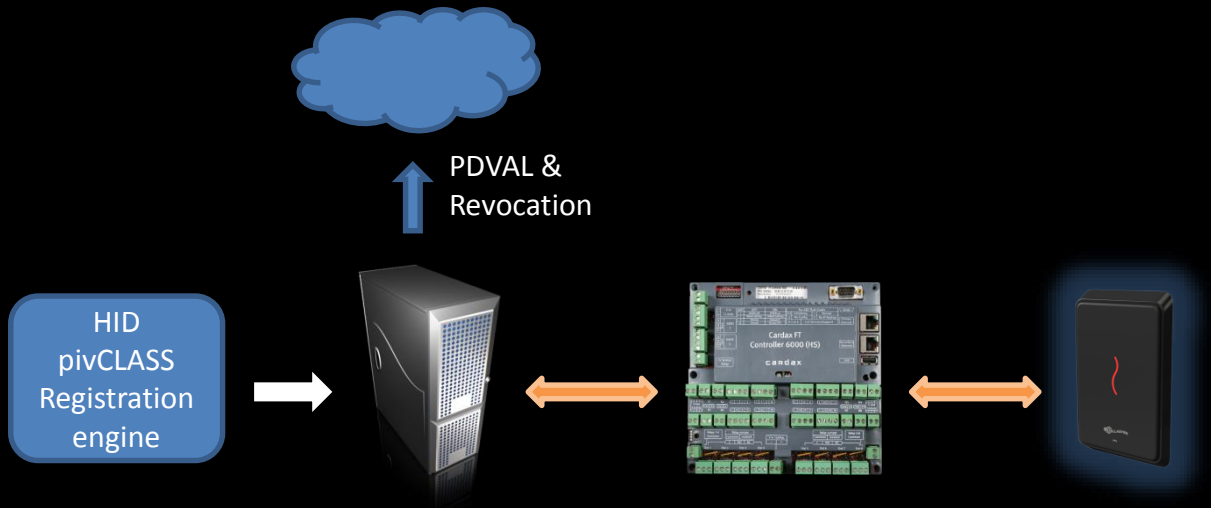
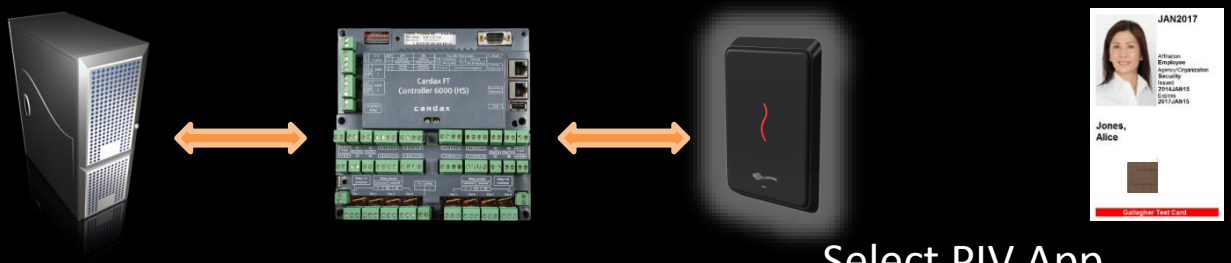
1.5 sec CAK Authentication...

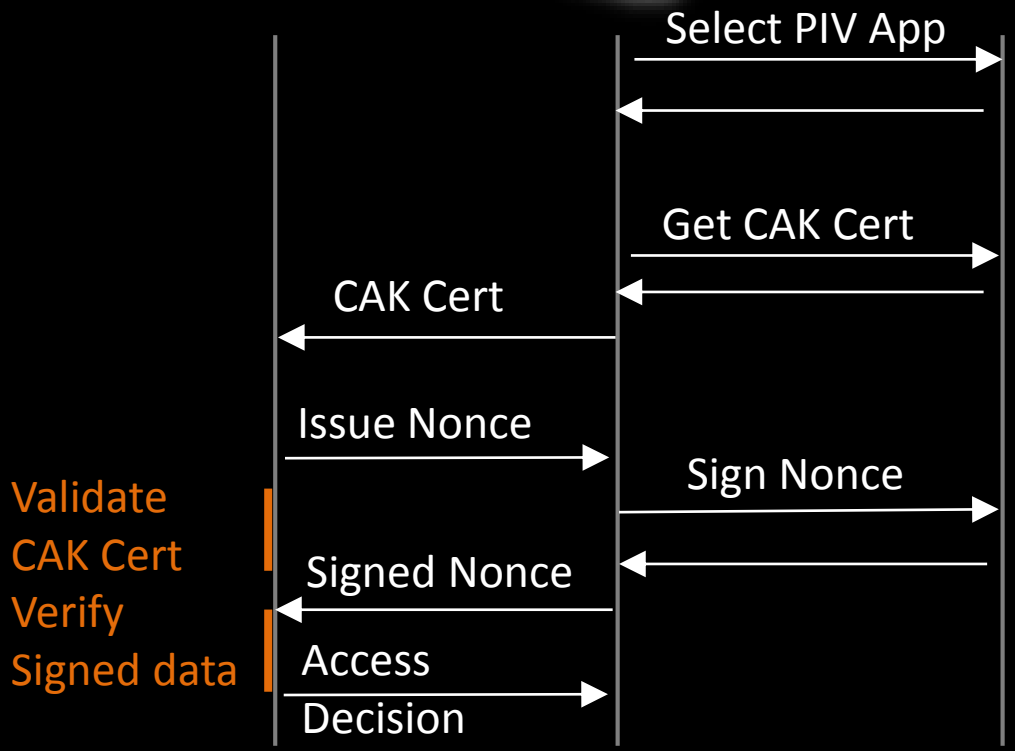
Is it possible?

Gallagher PIV Solution Architecture



Gallagher CAK Transaction





SP800-73-3 B.1.5

Choice of Crypto

ECC 1.2 seconds

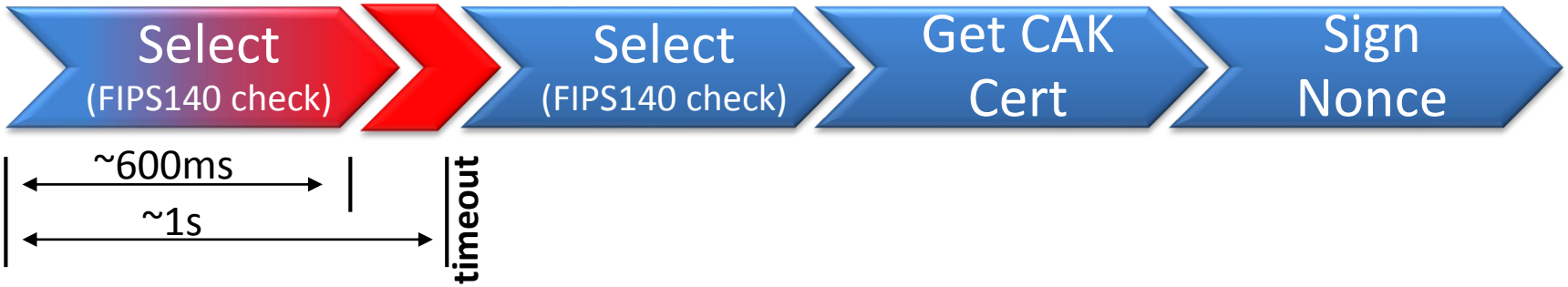
RSA 2.2 seconds

The importance of good RF

Bad RF coupling 3.2
second transaction

Good RF coupling 2.2
second transaction

What went wrong?



Insufficient power in card to complete the fips140 start-up checks

The importance of good RF

- Readers must supply sufficient energy
- Ideally the Antennas should be about the size of the card for good coupling and power transfer
- FICAM test requirements of 3cm is a fair test

How do we train the users to present the card at the optimum orientations when "prox" cards responded to a wave?

The importance of good RF

Optimizing Communication to the Card

Card Data Rate

Card Data Rate	Card Vendor 1 RSA 2048 (ms)		Card Vendor 2 RSA 2048 (ms)	
106 kbps	2387		2860	
212 kbps	2320	3%	2797	2%
424 kbps	2270	5%	2760	3%

At the higher data rates there is a higher chance of failed reads

Conclusion: stick to the base rate

Optimizing Communication to the Card

Extended ADPU's

CHUID read option	Transaction Time (ms)	Improvement
Standard -- command chaining	1336	
Optimized -- Extended length APDUs	1150	14%

For this transaction close to 200ms improvement

Conclusion: Use Extended ADPU's wherever the card supports the capability

Compress the Certificates?

Using NIST Test Cards

Card 10: RSA2048 uncompressed Certificates

Card 11: RSA 2048 Gzip compressed Certificates

Average of 5 reads (s)	Uncompressed	Compressed
T10 Reader	2.11	2.09
T11 Reader	2.11	2.14
T20 Reader	2.18	2.15

Conclusion: For these particular cards no significant improvement

Controller to Reader Communications

HBUS – Gallagher's RS485 protocol

Designed prior to PIV project to meet our other access control and sensor needs

- **1M bps** traditional RS485 protocols run up to 38K4 bps some now using 115K2 bps

- **Not Polled** traditional multi-dropped RS485 2 wire protocols are polled e.g. if polling each device at 5 polls per second then average queueing delay is 100ms

Where does the time go?

	RSA2048	ECC P256
Card start-up	28%	35%
Read CAK certificate	15%	22%
Dispatch CAK to Controller, receive challenge	8%	9%
Card challenge & response	47%	24%
Controller verify challenge & grant access	3%	10%

Note: these percentages were measured in 2012 and vary between card vendors.

Is a 1.5 second ECC CAK transaction realistic?

Yes

But…