IDEMIA

Post-Quantum Protocols for Banking Applications

Luk Bettale, Marco De Oliveira, Emmanuelle Dottax

IDEMIA

NIST 4th PQC Standardization Conference – Dec. 2022

Context

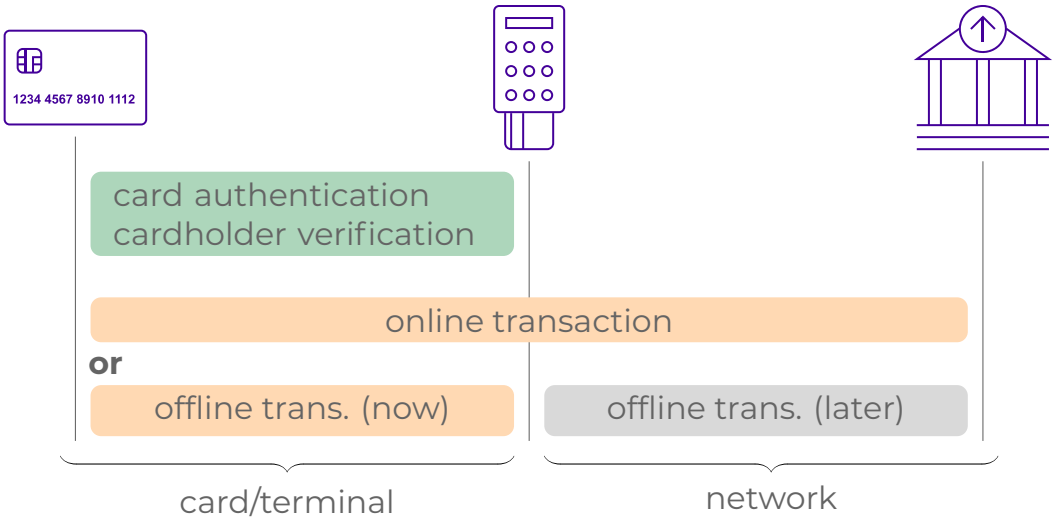
Payment Applications

- Implemented in smartcards or smartphones
- Contact or contactless payment
- Typical actors: Cardholder+Card, Merchant+Terminal, Issuer+Server

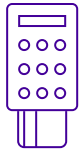
EMVCo: a “standard” for card-based payment

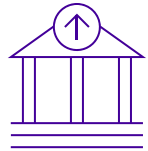
- EMVCo: global technical body formed in 1999
- Overseen by six member organizations – American Express, Discover, JCB, Mastercard, UnionPay and Visa
- Manages and evolves EMV Specifications
- More than 90% transactions worldwide

Card-Based Transactions



Common Part





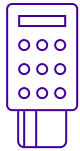
card authentication
- Verif. card's certs. (asym.)
- Dynamic challenge (asym.)

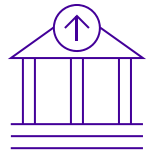
cardholder PIN verif.
- Send encrypted PIN (asym.)

card/terminal

network

Online Transaction





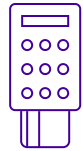
online transaction

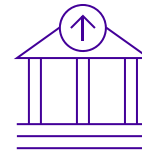
- Card generates ARQC (sym.)
- Terminal forwards to issuer for verif.
- If approved, card generates a TC (sym.)
- Terminal forwards to issuer

card/terminal

network

Offline Transaction





offline trans. (now)

- Card generates a TC (sym.)
- Terminal stores TC

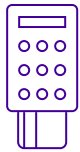
offline trans. (later)

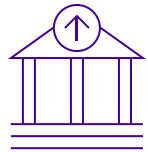
- Terminal forwards to issuer
- Issuer verifies TC

card/terminal

network

Offline Transaction





offline trans. (now)
- Card generates a TC (sym.)
- Terminal stores TC

offline trans. (later)
- Terminal forwards to issuer
- Issuer verifies TC

transaction accepted on the basis of card auth.

card/terminal

network

Quantum Threat

All current asymmetric crypto. would be broken by quantum computers.

Online transactions

- symmetric is quantum-safe (larger keys)
- no real threat

Offline transactions

- symmetric is quantum-safe (larger keys)
- if card authentication is broken ⇝ too late!

Quantum Threat

All current asymmetric crypto. would be broken by quantum computers.

Online transactions

- symmetric is quantum-safe (larger keys)
- no real threat

Offline transactions

- symmetric is quantum-safe (larger keys)
- if card authentication is broken ⇝ too late!

Wait-and-see policy?

- no “record now, harvest later” risk
- we can still force online transactions!

Why bother now?

Offline transactions are crucial in certain cases

- high volume/high speed payment scenarios
- cost of bandwidth for small merchants
- locations with low connectivity

Forbidding them would have a significant impact on the business

Why bother now?

Offline transactions are crucial in certain cases

- high volume/high speed payment scenarios
- cost of bandwidth for small merchants
- locations with low connectivity

Forbidding them would have a significant impact on the business

Viable quantum-safe offline solution will be a long process:

- Cryptographic migration in the payment industry is long and complex
- HW changes could be required to keep fast transactions

start experiments now → identify obstacles → facilitate the effort

Outline

Contributions

- Post-Quantum, efficient versions of current protocols
- Implementation of hybrid versions using NIST PQ finalists
- Performance analysis

Outline

1 〉 Existing Protocols

2 〉 Post-Quantum Migration

3 〉 Implementation Results

Outline

1 〉 **Existing Protocols**

2 〉 **Post-Quantum Migration**

3 〉 **Implementation Results**

Existing Protocols

Already Deployed (with RSA)

- Static Data Authentication (SDA)
- Dynamic Data Authentication (DDA)
- Combined Data Authentication (CDA)

Privacy Preserving Protocol (with ECC)

- Blinded Diffie-Hellman (BDH)

Existing Protocols

Already Deployed (with RSA)

- Static Data Authentication (SDA)
- Dynamic Data Authentication (DDA)
- Combined Data Authentication (CDA)

Privacy Preserving Protocol (with ECC)

- Blinded Diffie-Hellman (BDH)

Existing Protocols

Already Deployed (with RSA)

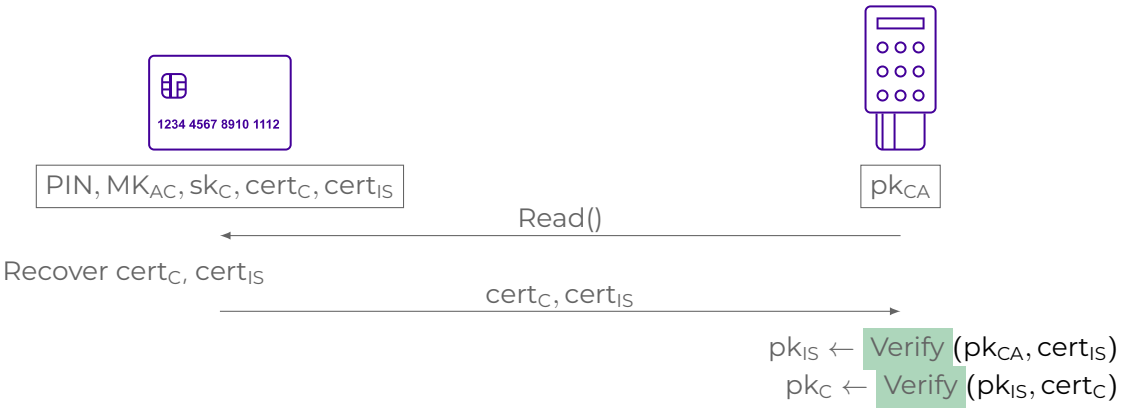
- Static Data Authentication (SDA)
- Dynamic Data Authentication (DDA)
- Combined Data Authentication (CDA)

Privacy Preserving Protocol (with ECC)

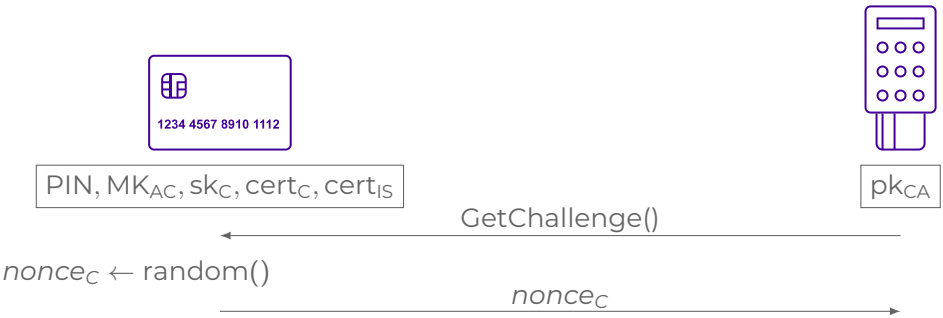
- Blinded Diffie-Hellman (BDH)

In this presentation: we only detail CDA migration.

EMV CDA protocol (1/4)

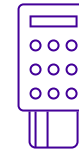


EMV CDA protocol (2/4)



EMV CDA protocol (3/4)

$PIN, MK_{AC}, sk_C, cert_C, cert_{IS}$



pk_{CA}

Get PIN from user: *uPIN*

$c \leftarrow$ RSA.Enc $(pk_C, uPIN|nonce_C)$

VerifyPIN(c)

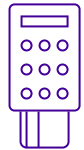
$uPIN|nonce \leftarrow$ RSA.Dec (sk_C, c)

Check *uPIN* and *nonce*

Success/Failed

EMV CDA protocol (4/4)

$PIN, MK_{AC}, sk_C, cert_C, cert_{IS}$



pk_{CA}

$nonce_T \leftarrow random()$

GenerateAC($nonce_T$)

$TC \leftarrow$ MAC $(MK_{AC}, transInfo)$

$data = nonce_C | TC | nonce_T$

$SDAD \leftarrow$ RSA.Sign $(sk_C, data)$

$TC, SDAD$

RSA.Verify $(pk_C, SDAD)$

If ok, transaction is accepted and *TC* stored.

Outline

1 〉Existing Protocols

2 〉**Post-Quantum Migration**

3 〉Implementation Results

Post-Quantum Migration

How to proceed?

- Smooth transition $\rightsquigarrow$ minimum amount of change in the transaction flow
- The lazy way: replace by equivalent primitives (enc/sign)
- The efficient way: use key encapsulation mechanism

Post-Quantum Migration

How to proceed?

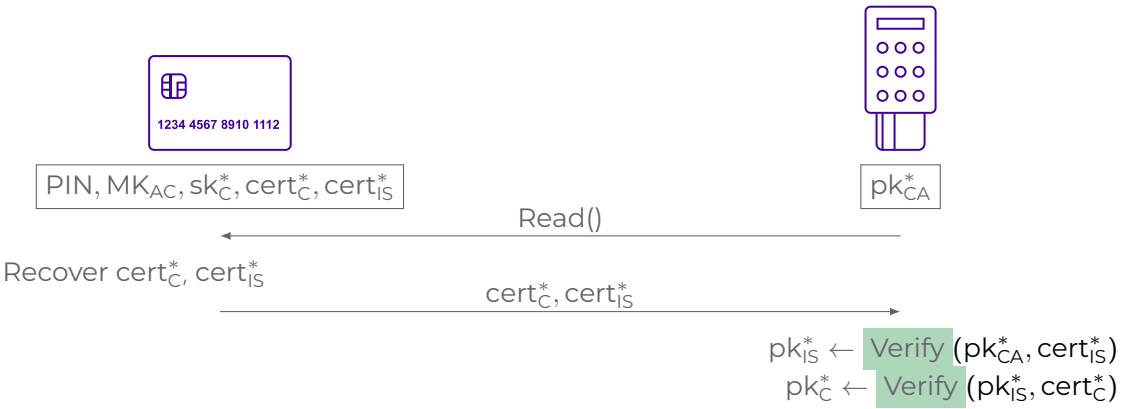
- Smooth transition ⇝ minimum amount of change in the transaction flow
- The lazy way: replace by equivalent primitives (enc/sign)
- The efficient way: use key encapsulation mechanism

Advantages

- Key decaps. faster than signature with PQ algorithms
- Only **one asym.** operation to perform on card
- Ciphertexts *usually* smaller than signatures

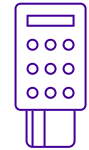
- Use **AES-256** for all symmetric primitives
- Use authenticated encryption for both confidentiality **and** authenticity

PQ analog of CDA protocol (1/4)



PQ analog of CDA protocol (2/4)

$PIN, MK_{AC}, sk^*_C, cert^*_C, cert^*_{IS}$



pk^*_{CA}

$ss, ct \leftarrow$ Encaps (pk^*_C)

GetChallengeAndEstablishKey(ct)

$ss \leftarrow$ Decaps (sk^*_C, ct)

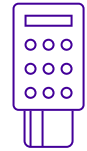
$nonce_C \leftarrow$ random()

$K \leftarrow$ KDeriv($ss|nonce_C$)

$nonce_C$

PQ analog of CDA protocol (3/4)

$PIN, MK_{AC}, sk^*_C, cert^*_C, cert^*_{IS}$



pk^*_{CA}

$K \leftarrow \text{KDeriv}(ss|nonce_C)$

Get PIN from user: $uPIN$

$c \leftarrow \text{AEnc}(K, uPIN)$

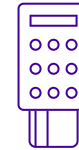
VerifyPIN(c)

$uPIN \leftarrow \text{ADec}(K, c)$

check $uPIN = \text{PIN}$

Success/Failed

PQ analog of CDA protocol (4/4)



$\text{PIN}, \text{MK}_{\text{AC}}, \text{sk}^*_{\text{C}}, \text{cert}^*_{\text{C}}, \text{cert}^*_{\text{IS}}$

pk^*_{CA}

GenerateAC()

$TC \leftarrow \text{MAC}(\text{MK}_{\text{AC}}, transInfo)$

$SDAD \leftarrow \text{AEnc}(K, TC)$

$SDAD$

$TC \leftarrow \text{ADec}(K, SDAD)$

If ok, transaction is accepted and *TC* stored.

Hybrid Protocol

Hybrid Cryptography

Combine classical and post quantum cryptography.

Why Hybrid Cryptography?

- Proposed PQ scheme could still be broken (e.g. Rainbow, SIKE)
- Take advantage of mature algorithms
- Solution promoted by several agencies

Hybrid Protocol

Hybrid Cryptography

Combine classical and post quantum cryptography.

Why Hybrid Cryptography?

- Proposed PQ scheme could still be broken (e.g. Rainbow, SIKE)
- Take advantage of mature algorithms
- Solution promoted by several agencies

Our strategy

- Store both classic and PQ certificates on card
- Concatenate signatures (e.g. $\text{RSA.Sign}(\text{sk}_C, data) \| \text{AEnc}(K, data)$)
- Stack encryption (e.g. $\text{RSA.Enc}\left(\text{pk}_C, \text{AEnc}(K, PIN)\right)$)

Outline

1 〉 Existing Protocols

2 〉 Post-Quantum Migration

3 〉 Implementation Results

Choice of PQ algorithms

NIST Standardization Process

- Selection not finished when this work started
- 4 finalists for KEM
- 3 finalists for Signature

NIST Selected KEMs

- CRYSTALS-Kyber
- NTRU
- Saber
- Classic McEliece

NIST Selected Signatures

- CRYSTALS-Dilithium
- Falcon
- Rainbow

- Only a KEM is required on smartcard: signature/verification is performed outside.
- Signature choice impacts only the size of the certificates

Choice of PQ algorithms

NIST Standardization Process

- Selection not finished when this work started
- 4 finalists for KEM
- 3 finalists for Signature

NIST Selected KEMs

- CRYSTALS-Kyber
- NTRU
- Saber
- Classic McEliece

NIST Selected Signatures

- CRYSTALS-Dilithium
- Falcon
- ~~Rainbow~~

- Only a KEM is required on smartcard: signature/verification is performed outside.
- Signature choice impacts only the size of the certificates

Choice of PQ algorithms

NIST Standardization Process

- Selection not finished when this work started
- 4 finalists for KEM
- 3 finalists for Signature

NIST Selected KEMs

- CRYSTALS-Kyber
- ~~NTRU~~
- ~~Saber~~
- ~~Classic McEliece~~

NIST Selected Signatures

- CRYSTALS-Dilithium
- Falcon
- ~~Rainbow~~

- Only a KEM is required on smartcard: signature/verification is performed outside.
- Signature choice impacts only the size of the certificates

Personalization overhead

Figures for an ARM Cortex-M3 device running at 100 MHz, with `kyber-512`.

Algorithm	Overhead (bytes)		Increase (%)
	Theoretical	Measured	
Falcon-512	4661	5120	39.48
Falcon-1024	6785	7168	47.73
Dilithium-2	8584	9216	54.01
Dilithium-3	10 970	11 264	58.93
Dilithium-5	14 214	N/A	N/A

Personalization overhead

Figures for an ARM Cortex-M3 device running at 100 MHz, with `kyber-512`.

Algorithm	Overhead (bytes)		Increase (%)
	Theoretical	Measured	
Falcon-512	4661	5120	39.48
Falcon-1024	6785	7168	47.73
Dilithium-2	8584	9216	54.01
Dilithium-3	10 970	11 264	58.93
Dilithium-5	14 214	N/A	N/A

Remarks

- Same overhead for hybrid BDH variant
- Difference theoretical/measured due to pagination
- More personalization data ⇝ slower perso = less cards produced: huge impact

Full transaction timings

Figures for an ARM Cortex-M3 device running at 100 MHz, with `kyber-512`.

Protocol	Certificate Algo.	Timings (ms)			Ratio
		Comm.	Card	Total	
Hybrid CDA	Falcon-512	5267	376	5643	3.36
	Falcon-1024	7302	376	7678	4.58
	Dilithium-2	9018	384	9402	5.61
	Dilithium-3	11 297	402	11 699	6.98

Full transaction timings

Figures for an ARM Cortex-M3 device running at 100 MHz, with `kyber-512`.

Protocol	Certificate Algo.	Timings (ms)			Ratio
		Comm.	Card	Total	
Hybrid CDA	Falcon-512	5267	376	5643	3.36
	Falcon-1024	7302	376	7678	4.58
	Dilithium-2	9018	384	9402	5.61
	Dilithium-3	11 297	402	11 699	6.98

Remarks

- Ratio w.r.t. classic protocol
- Most of the time spent in communication!
- Slightly worse for hybrid BDH variant
- Very similar results with `ntruhps2048509`

Conclusion

Summary

- Using KEM for authentication is more efficient
- Communication cost is significantly higher than computation one
- In this context, short signatures should be preferred
- Same conclusions apply equally to CRYSTALS-Kyber and NTRU
- See the paper for BDH variant

Future Work

- Formal proof of the protocols
- More studies of post-quantum migration in other domains

Questions?

IDEMIA

Join us on

www.idemia.com