Post quantum cryptography in payment transactions

Digital Payments
for a Trusted World

The viewpoint of a Payment Service Processor

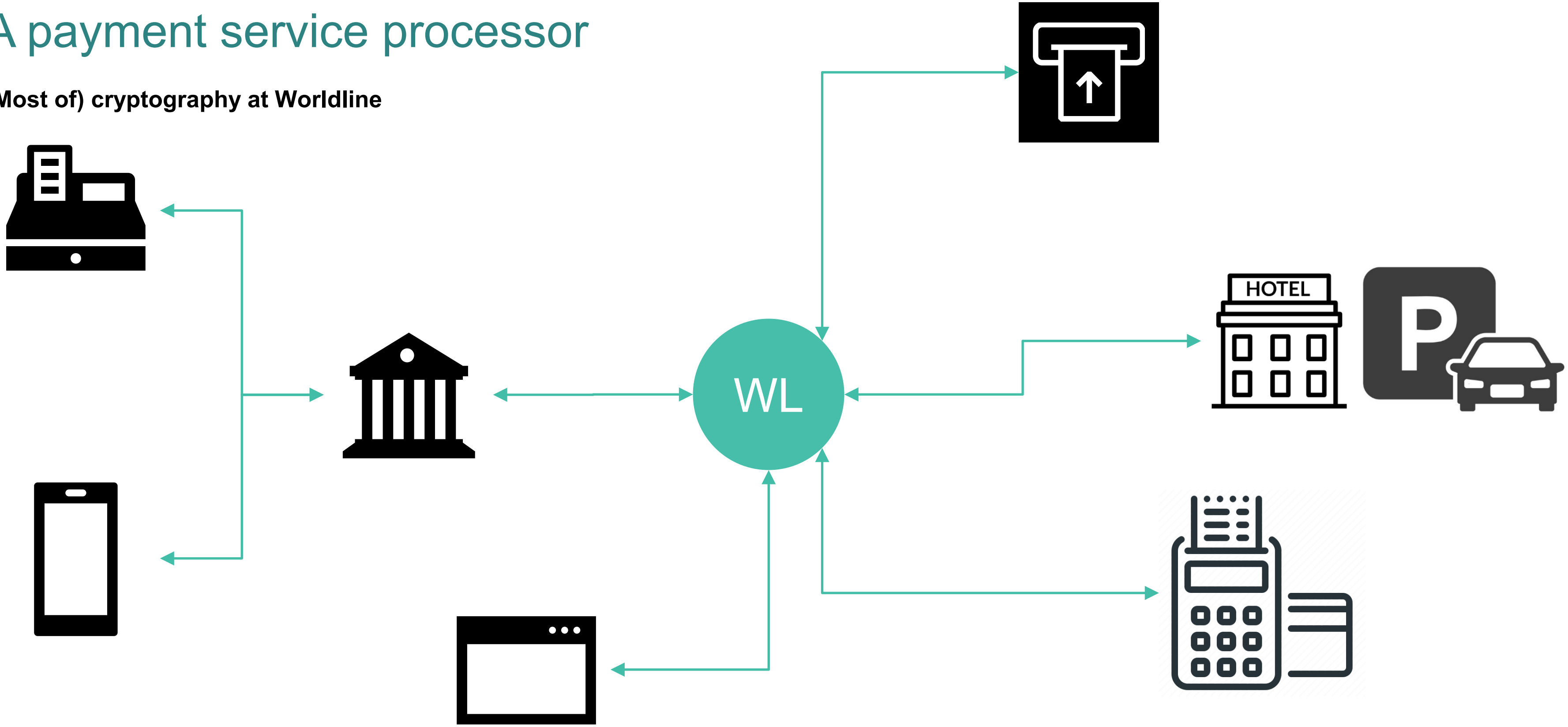
Basically, most of it is communication encryption

Digital Payments
for a Trusted World

WORLDLINE

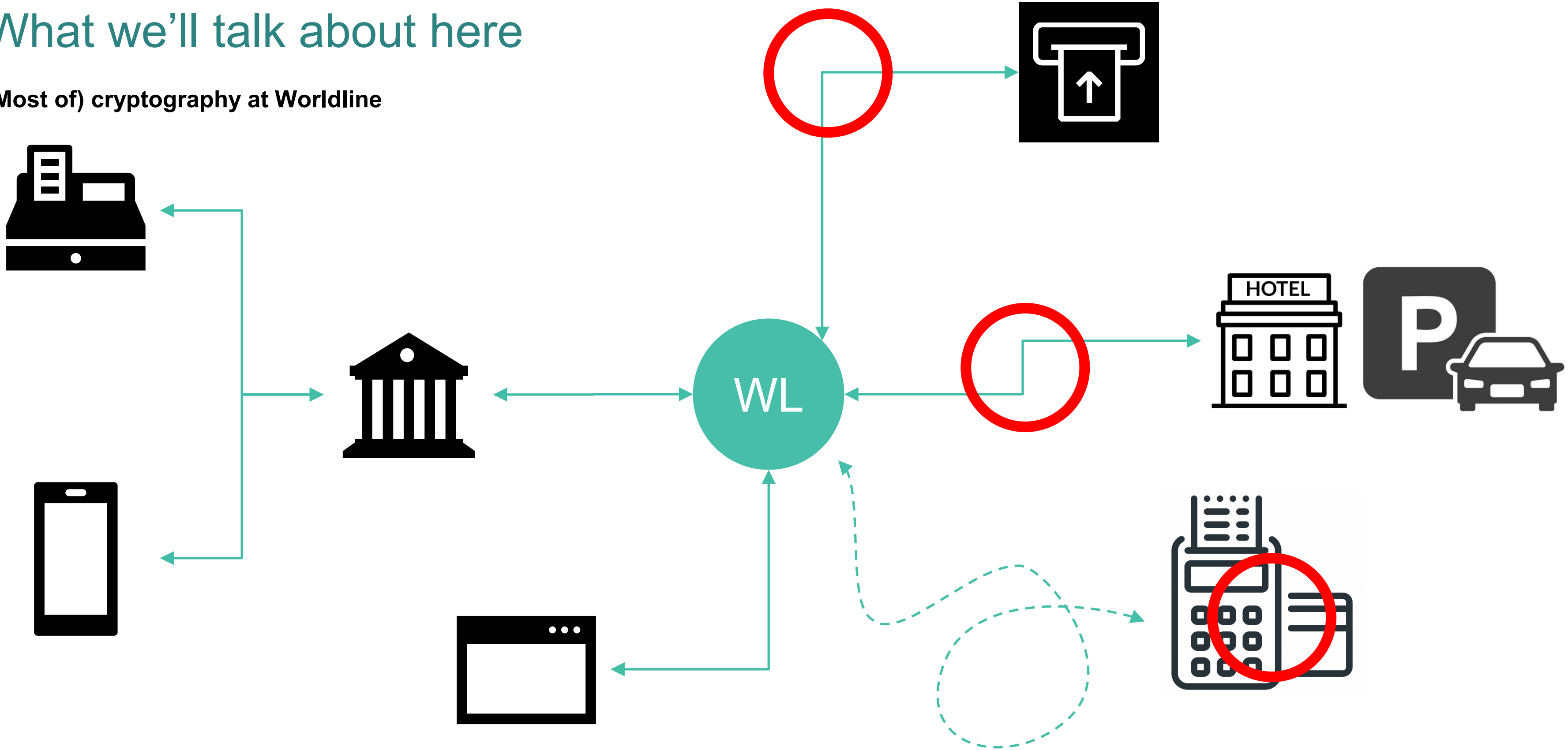
A payment service processor

(Most of) cryptography at Worldline



What we'll talk about here

(Most of) cryptography at Worldline



Generic requirements for crypto for payment

What we sell it trust after all

What the payment world sells, is trust

All requirements follow from there

Trust is complicated: easy to lose but hard to win.
Basically:

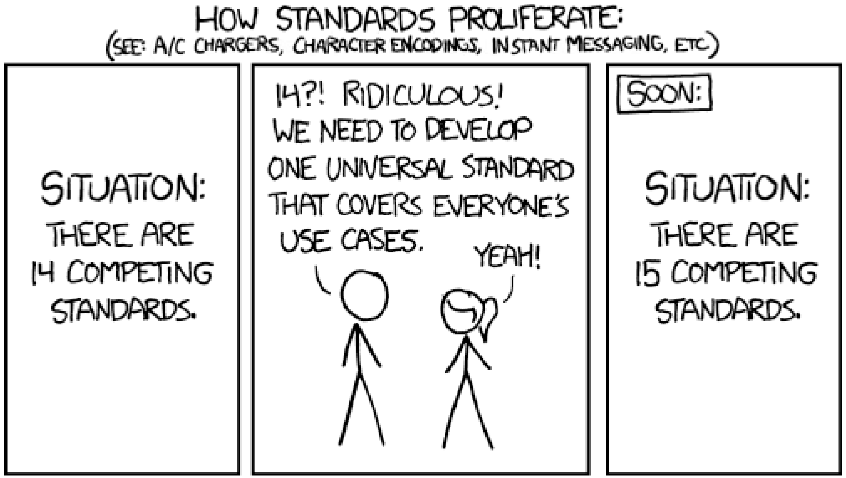
- Things have to be secure
- People have to think it's secure

Use of multi layered protection is default: for example, detection systems on top of cryptographic protection.

Generic requirements: standards

Standards are needed for interoperability and security

- In crypto, open standards are fundamental of course
- EMV, ISO, ETSI, IETF
- NIST is often used because it is well documented and free, but payments providers are not the target audience (that’s the US government)



Payment protocol standards:

- Nexo (too many options, not a standard)
- C-TAP (underused)

Generic requirements: compliance

Which is not the same as quality nor security

Thanks to astronomical fines put on the payment industry by PCI, we cannot help but being compliant.

Currently, this means we have to:

- Use enormous key sizes
- Use “key blocks” even for session keys (more on that later)

Example of insecure but compliant:

- “Transparent” disk encryption

A page from PCI PIN

This is how PCI understands cryptography

	Algorithm				
Bits of Security	**DEA**	**IFC** (RSA)	**ECC** (ECDSA, ECDH, ECMQV)	**FFC** (DSA, DH, MQV)	**AES**
80	112	1024	160	1024/160	–
112	168	2048	224	2048/224	–
128	–	3072	256	3072/256	128
192	–	7680	384	7680/384	192
256	–	15360	512	15360/512	256

Source: PCI PIN Requirements and testing procedures, version 3.1, March 2021.

Specific requirements for crypto for payments

Online payments
Offline payments
A word on HSMs

Payments over the internet: everybody does it

And still, it is not a priority for me

Security if of the internet is everybody's problem, not only of the payment industry

- What is going to be the PKI infrastructure of the future internet?
- Key exchange seems to be solved (newHope trial), but authentication isn't; stateful signatures are not going to be the solution, I believe

What everybody needs, is "PQC-TLS" that allows a party (read: web site) to authenticate itself towards an unsuspecting other side (read: browser). This is more than just a crypto protocol.

For internet payments, protocols like Ideal are used.

- Post quantum secure in principle, but limited by smart card hardware

Online payments: quantum proof, but we still need help

Not to be confused with internet payments

The “western” countries use a payment model where there is an online connection to the cardholder’s bank (called Issuer in our terminology).
This allows “classical” cryptography to be used: Smart cards use a secret key that is shared with the issuing bank, allowing secure communication for verification and balance verification. No needs for PQC here, although the protocols probably need updating.

For transmission of payment information, PCI has the following requirements that are hard to implement with current standards:

- Use session keys
- Exchange session keys either with key blocks, or with (D)UKPT.

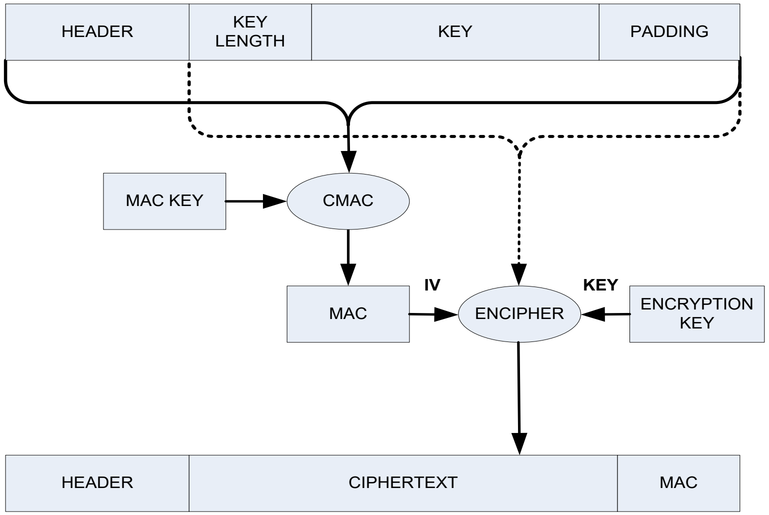
Online payments 2: key blocks

Now required by PCI for all “external” keys

If used well, you can get extra functionality for your HSM.
In the header, properties of the key can be defined.

One example: “MAC verification keys”

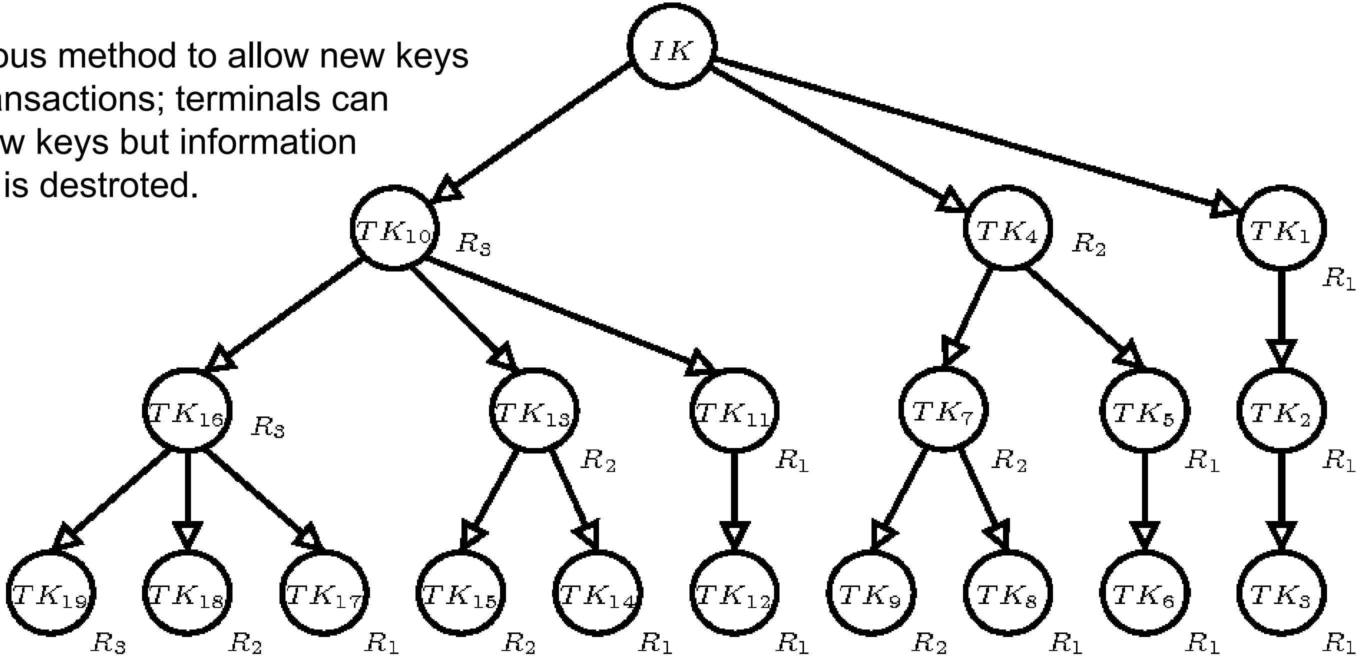
Admittedly, not much advanced cryptography here, but this is currently something that keeps everyone busy!



Online payments 3: DUKPT

A clever way to make terminals forget their old keys

Very ingenious method to allow new keys for every transactions; terminals can compute new keys but information on old keys is destroted.



Online payments 4: Jurjen’s naïve design for UKPT

I made this up while cycling through the dunes of Heemskerk

R1 →

← R2

SK = HMAC(TK, R1 || R2)

SK: session key

TK: transport key (exchanged separately beforehand)

R1: random number sent by Alice

R2: random number sent by Bob

Offline payments: something NIST doesn't address

Smart card payments without connection

Although not used very frequently in Europe, there is a need for a protocol for offline payments between cards and terminals.
The Current EMV implementation for cards uses authentication based on certificates: (like internet PKI).

For the future, we need a protocol that fulfills these needs, and is PQC, and with limited computation and not much bandwidth.

WORLDLINE

Offline payments, 2

Smart card payments without connection

In order to prevent PIN theft using a “shim device”,
EMV uses public key encryption of PIN by the terminal.

In the current implementation, the terminal uses RSA public key encryption using the cards’ public key to encrypt the PIN before transmission to the card.

Here, a replacement protocol is needed that allows to verify the validity of the smard card, and encrypt something with the smart card’s public key.

Advantages of HSMs

Where it is useful to spend money on hardware

Using HSMs for secure protocols

HSMs can do great things for security and cryptography, but they don’t

HSMs are used everywhere in payments

- Strangely enough, not always in a secure fashion:
  - “Encrypt” command is most often used
  - Signing of messages provided to HSM as a hash
  - HSM in the cloud is more popular that it should be

The future of HSMs is in understanding what they do, e.g.:

- They are a key management tool rather than an encryption tool
- API designed so that the protocol is secure
- How to make stateful signatures with HSMs

Needs for crypto research from payment industry

Secure payment protocols

- Allowing multi-issuer offline card authentication
- Efficient key exchange for high speed processing
- Lightweight protocol for public encryption for PINs

Help

If HSMs are involved

- design new HSM API that allows PQC and is secure from the ground up

HAPKIDO and PractiQal may be the solution!

For more information

Jurjen N.E. Bos

Jurjen.Bos@Worldline.com

Thanks for your attention