


**Cryptography and Quantum Key Distribution**

2022-10-03: IOTALENTUM WORKSHOP TTW2 – CYBERSECURITY AND APPLICATIONS

S. R. Verschoor

Department of Electrical Engineering




1

**Outline**

- Cryptography
  - Basics
  - Post-Quantum Cryptography (PQC)
- Quantum Key Distribution (QKD)
  - QKD Network
  - TU/e testbed

2 Cryptography and Quantum Key Distribution




2

**Cryptography – the basics**

- Alice and Bob want to communicate
  - Mallory is actively interfering with them
    - (in some weaker models Eve is only passively eavesdropping)
- Kerckhoff's principle
  - aka Shannon's Maxim: "the enemy knows the system"
  - but Mallory does not know the keys
- Mallory carries the messages (Dolev-Yao model)
  - she can inspect, change, re-order, replay, drop, inject any message
  - may (sometimes) compromise some participants

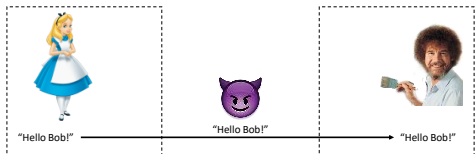
4 Cryptography and Quantum Key Distribution




4

**Confidentiality**

- Alice and Bob want their message to remain secret




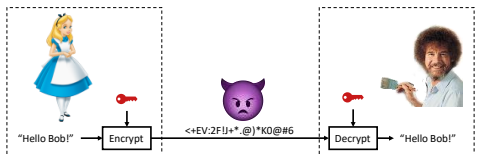
5 Cryptography and Quantum Key Distribution




5

**Encryption**

- Symmetric encryption: Alice and Bob need to share a secret key 
- examples: AES, ChaCha20, one-time pad



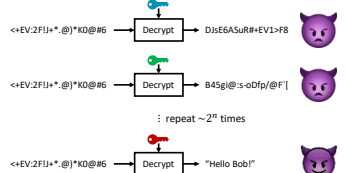
6 Cryptography and Quantum Key Distribution




6

**Confidentiality (computational)**

- The ciphertext "gives no information" about the plaintext
- $n$ -bit security: Mallory expects to try  $2^n$  keys before finding the right one



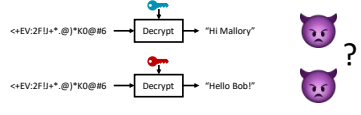
7 Cryptography and Quantum Key Distribution



7

### Confidentiality (information theoretical)

- Perfect security
- Mallory has no way of distinguishing correct decryptions from incorrect ones
- Requires a **one-time pad**
- Key can only be used once



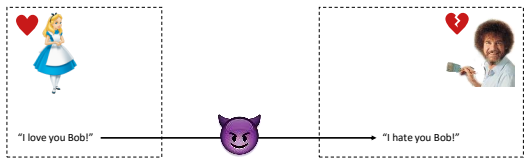
8 Cryptography and Quantum Key Distribution



8

### Integrity and authentication

- Integrity: nobody should be able to change the message
- Authentication: Bob knows the message came from Alice



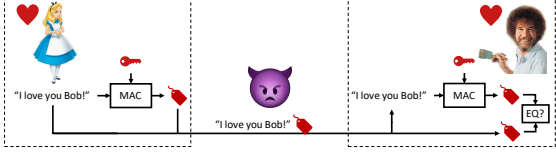
9 Cryptography and Quantum Key Distribution



9

### Message Authentication Code (MAC)

- Symmetric: Alice and Bob need to share a secret key 🔑
- allows Bob to detect any changes
- examples: HMAC, Poly1305



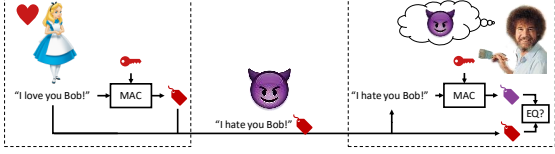
10 Cryptography and Quantum Key Distribution



10

### Message Authentication Code (MAC)

- Symmetric: Alice and Bob need to share a secret key 🔑
- allows Bob to detect any changes
- examples: HMAC, Poly1305



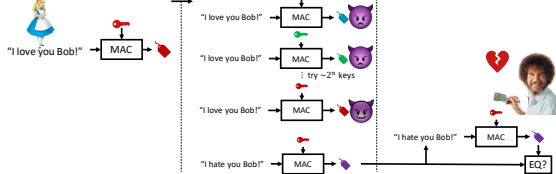
11 Cryptography and Quantum Key Distribution



11

### Unforgeability (computational)

- Mallory cannot forge tags for any (other) message
- $n$ -bit security: Mallory can locally try to find the correct key among  $2^n$  keys



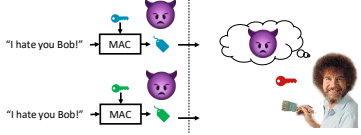
12 Cryptography and Quantum Key Distribution



12

### Authentication (information theoretical)

- Mallory cannot verify forgeries locally
- $n$ -bit security: each forgery succeeds with probability  $2^{-n}$
- statistical security
- Requires discarding the authentication key (or at least some part of it)
- "encrypt" the tag with a one-time pad



13 Cryptography and Quantum Key Distribution



13

### Authenticated Encryption

- Combined encryption and authentication
- required for confidentiality against active attackers!

14 Cryptography and Quantum Key Distribution

14

TU/e

### Cryptographic hashing

Given a long message  $M$ , a hash function computes a *small* message digest

The digest is also called the fingerprint, or simply "the hash of  $M$ ". Note there is no key involved.

Hash should behave as a random function:

- given  $M$ , it should be hard to compute  $M$
- it is hard to find any  $M_0, M_1$  such that  $\text{Hash}(M_0) = \text{Hash}(M_1)$

Hash functions are used everywhere in cryptography. Examples: MD5 (broken), SHA2, SHA3

15 Cryptography and Quantum Key Distribution

15

TU/e

### Public key cryptography

- Parties generate a keypair: (🔑, 🗝️)
- give the public key (🗝️) to everybody, so anybody can use it
- keep the private key (🔑) secret
- Also called asymmetric cryptography
- Example usage:
  - key exchange
  - digital signatures
  - public key encryption
- Example systems, used on the internet today:
  - RSA
  - Elliptic curve cryptography (ECC)
  - Diffie-Hellman key exchange (DH)

16 Cryptography and Quantum Key Distribution

16

TU/e

### Public key encryption

- Bob generates a keypair: (🔑, 🗝️), gives 🗝️ to Alice
- Provides confidentiality, but no authenticity (because everybody can encrypt)

17 Cryptography and Quantum Key Distribution

17

TU/e

### Digital signatures

- Alice generates a keypair: (🔑, 🗝️), gives 🗝️ to Bob
- Alice can put a signature (📄) on any message, using her private key (🔑)
- Provides:
  - message integrity (nobody can change the message)
  - message authentication (Bob knows message came from Alice)
  - non-repudiation (Alice can't deny signing message)

18 Cryptography and Quantum Key Distribution

18

TU/e

### Example: RSA

- Rivest-Shamir-Adleman (RSA)
- Private key (🔑): two random large primes ( $p, q$ )
- Public key (🗝️):  $N = p \cdot q$
- System parameter:  $e$  (usually 65537)
- Security based on the hardness of factoring
  - given  $N$ , it should be hard to find  $p, q$

19 Cryptography and Quantum Key Distribution

19

TU/e

### Example: RSA-KEM

- Key encapsulation mechanism (KEM)
  - generate a random symmetric key  $k$
  - authenticate+encrypt the message using  $k$
  - encapsulate  $k$  to the recipient's public key  $(N)$
- Alice knows Bob's public key  $N$ :
  - she generates a random  $k$
  - she encapsulates  $k$ :  $c = k^e \text{ mod } N$
- Bob, given  $c$  and using his private key  $(p, q)$ :
  - he computes:  $d = e^{-1} \text{ mod } (p-1)(q-1)$
  - he decapsulates:  $k = c^d \text{ mod } N$

20 Cryptography and Quantum Key Distribution TU/e

20

### Example: RSA signature

- Alice wants to sign message  $M$  using her private key  $(p, q)$ 
  - she hashes the message  $\sigma = H = \text{Hash}(M)$
  - she computes the signature  $\sigma = H^d \text{ mod } N$
- Bob verifies  $(M, \sigma)$  using Alice's public key  $(N)$ 
  - he computes  $H' = \sigma^e \text{ mod } N$
  - he hashes the message  $H = \text{Hash}(M)$
  - he checks if  $H = H'$

21 Cryptography and Quantum Key Distribution TU/e

21

### Key authentication

- Public keys are usually provided at the start of a protocol
- How do you know the key actually belongs to the claimed owner?
  - you need key authentication, otherwise you are vulnerable to a Mallory-in-the-Middle attack

22 Cryptography and Quantum Key Distribution TU/e

22

### Certificates

- Requires a trusted third party
- Alice must have a key, for example pre-installed on her computer

23 Cryptography and Quantum Key Distribution TU/e

23

### Quantum Computers

Two algorithms threaten existing cryptography

- Shor's algorithm for period finding can efficiently ...
  - ... factor  $N \Rightarrow$  breaks RSA
  - ... find discrete logarithms  $\Rightarrow$  breaks ECC, breaks DH
- Grover's search algorithm can ...
  - ... try  $2^n$  keys with only  $2^{n/2}$  quantum queries  $\Rightarrow$  double key-length suffices for symmetric cryptography

\*) More algorithms exist, but their impact on widely deployed cryptography is roughly the same as Grover's algorithm.

25 Cryptography and Quantum Key Distribution TU/e

25

### Harvest now, decrypt later

26 Cryptography and Quantum Key Distribution TU/e

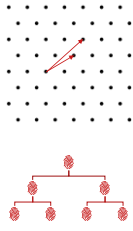
26

### Post-Quantum Cryptography (PQC)

Alice & Bob have classical computer  
Mallory has a quantum computer

Replace factoring (or discrete log) with other problems:

- Lattice-based cryptography
  - both KEMs and signatures
- Hash-based cryptography
  - signatures
- Error correcting codes
  - KEMs
- Multivariate cryptography
  - (mainly) signatures
- Isogeny-based cryptography (maybe broken?)

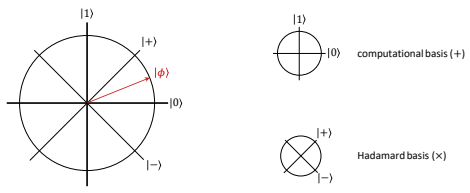


27 Cryptography and Quantum Key Distribution TU/e

27

### Quantum information (the bare minimum for QKD)

A qubit is a vector



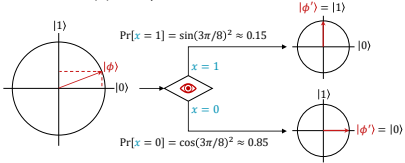
29 Cryptography and Quantum Key Distribution TU/e

29

### Measurement

If we measure (⊗) a qubit

- it collapses onto the measurement basis
- with probability defined by the in-product of qubit and basis vector
- we get a classical bit (x) as output



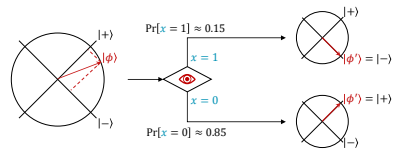
30 Cryptography and Quantum Key Distribution TU/e

30

### Measurement

If we measure (⊗) a qubit

- it collapses onto the measurement basis
- with probability defined by the in-product of qubit and basis vector
- we get a classical bit (x) as output



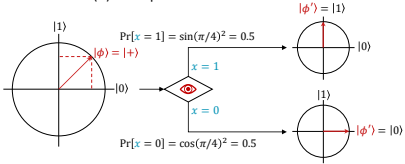
31 Cryptography and Quantum Key Distribution TU/e

31

### Measurement

If we measure (⊗) a qubit

- it collapses onto the measurement basis
- with probability defined by the in-product of qubit and basis vector
- we get a classical bit (x) as output



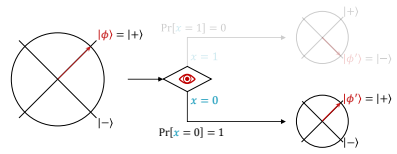
32 Cryptography and Quantum Key Distribution TU/e

32

### Measurement

If we measure (⊗) a qubit

- it collapses onto the measurement basis
- with probability defined by the in-product of qubit and basis vector
- we get a classical bit (x) as output



33 Cryptography and Quantum Key Distribution TU/e

33

### Bennett-Brassard (BB84)

1. |+⟩ |1⟩ |−⟩ |0⟩ |1⟩ |1⟩ |0⟩ |0⟩ |−⟩ |+⟩ |−⟩ |+⟩ |+⟩ |1⟩

2. + × + × × + + × × × +

3. ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

4. 1 0

5. ✓

6. 1 0 1 1

- Alice sends **random qubits** (some may not arrive)
- Bob measures in **random bases**, reveals them to Alice **after** the measurement
- Alice confirms when sending/measurement basis were the same
- Bob reveals each measurement outcome bit with probability 1/2
- Alice confirms the bits are correct (and aborts if any bit is incorrect)
- Both use the remaining bits as shared key: 1011

All classical messages are authenticated, as indicated by the tags (🔒).

34 Cryptography and Quantum Key Distribution **TU/e**

34

### BB84, improvements

- Information reconciliation
  - error *correction* instead of error detection
- Privacy amplification
  - Mallory may have some information about the secret bits
  - "distill" these bits a shorter key so Mallory has only negligible information
- Require fewer check bits

35 Cryptography and Quantum Key Distribution **TU/e**

35

### Security of QKD

- Key is statistically independent from Mallory's observations
  - cannot be broken by trying more keys or future cryptanalysis
  - can be broken by exploiting discrepancies between hardware and model
- Use key as one-time pad + statistical MAC:
  - security independent of any computational assumptions
- Use key in computational (symmetric) cryptography
  - breaks only if the computational cryptography breaks
  - (this is often done because of the low key-rate of QKD)

36 Cryptography and Quantum Key Distribution **TU/e**

36

### QKD authentication

- Authentication typically done with statistically secure MACs
  - but then we assume shared keys
    - so it's not key *distribution*, so much as it is key *expansion*
  - and we have to discard some key material
    - consumed keys can be replaced with fresh QKD output
    - requires some care to prevent key exhaustion (by Mallory)
- However, we can authenticate with computational MACs or signatures
  - if authentication isn't broken now, **then** the output key will never be broken later

37 Cryptography and Quantum Key Distribution **TU/e**

37

### QKD limitation

- QKD is a point-to-point protocol
- Single photon travel distance in fiber/free-space is limited
  - up to hundreds of kilometer (<100 km in practice)
  - but key-rate drops with larger distance
- No repeaters allowed
  - You cannot measure and resend the qubits (for the same reason Mallory can't)
  - quantum repeaters theoretically exist, but require stable quantum memory

40 Cryptography and Quantum Key Distribution **TU/e**

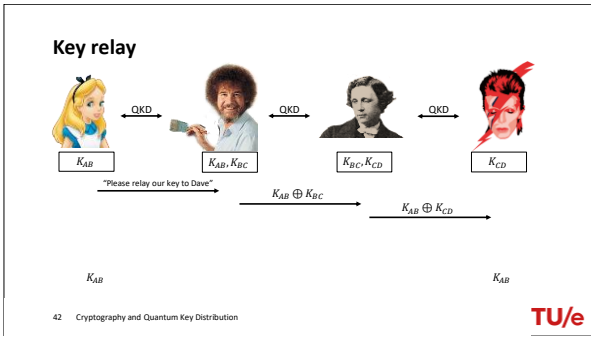
40

### Trusted repeater network

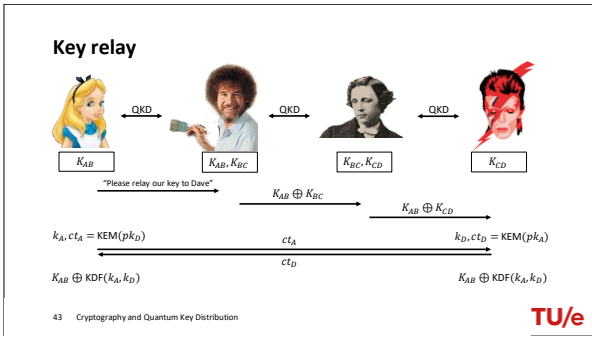
- Meet Alice, Bob, Carroll, and David
- Each neighbouring pair is linked via QKD
- They trust each other, which means ...
  - ... they follow the protocol specification
  - ... *throw away keys* after they have been used
  - ... take care of their devices and keep out hackers/three letter agencies

41 Cryptography and Quantum Key Distribution **TU/e**

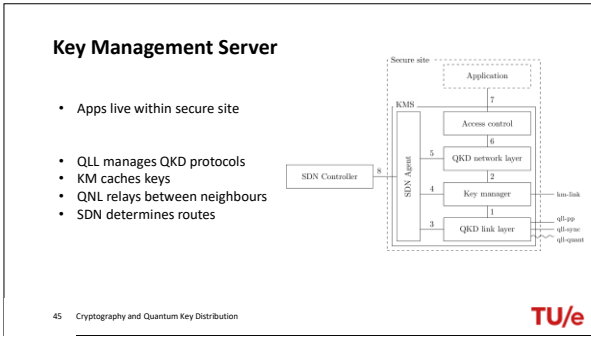
41



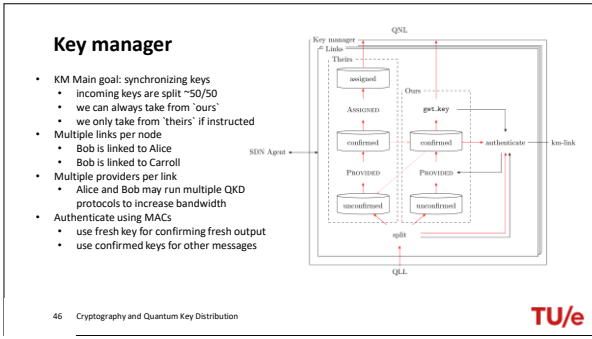
42



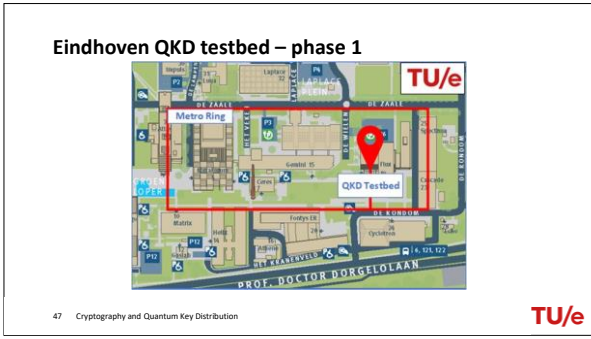
43



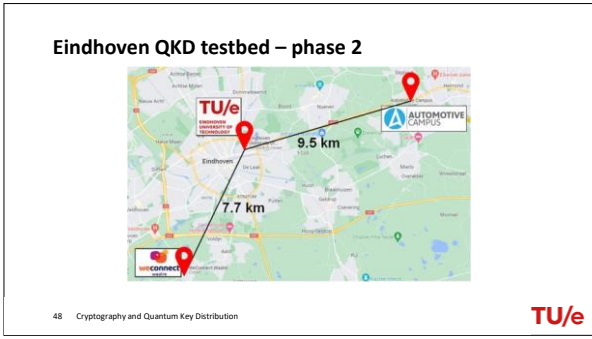
45



46

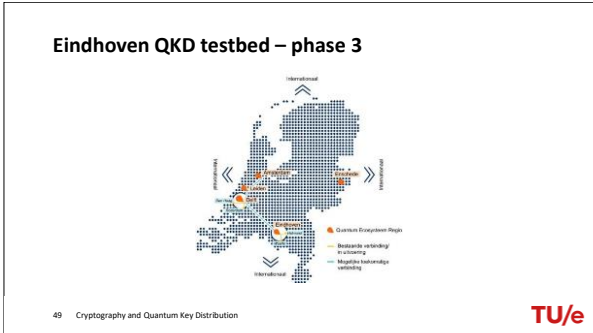


47



48





49



50

### Thank you

Slides are available online:  
<https://zeroknowledge.me/talks/#iotalentum22>

s.r.verschoor@tue.nl

51 Cryptography and Quantum Key Distribution

51

### Quantum information (slightly beyond the bare minimum)

A qubit is a binary state of a quantum system

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Generally  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , with  $|\alpha|^2 + |\beta|^2 = 1$

52 Cryptography and Quantum Key Distribution

52

### Quantum information (slightly beyond the bare minimum)

The dual vector of  $|\psi\rangle$  is  $\langle\psi| = (\alpha^*, \beta^*)$  (the conjugate transpose)  
 Then  $\langle\phi|\psi\rangle = \langle\phi| \cdot |\psi\rangle$  is an inner product.

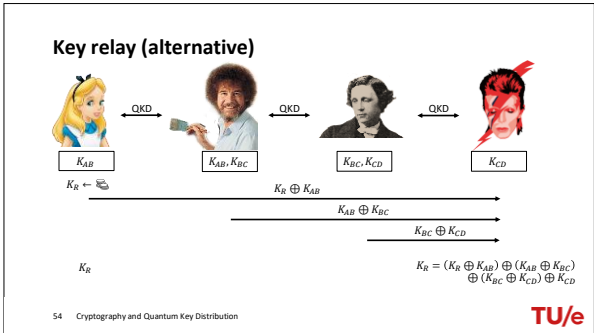
If we measure  $|\psi\rangle$  in computational basis  $\{|0\rangle, |1\rangle\}$ , then  $|\psi\rangle$  is destroyed and we get an output label  $x$ :  
 $\Pr[x = 0] = |\langle 0|\psi\rangle|^2$   
 $\Pr[x = 1] = |\langle 1|\psi\rangle|^2$

Similarly if we measure in Hadamard basis  $\{|+\rangle, |-\rangle\}$ :  
 $\Pr[x = 0] = |\langle +|\psi\rangle|^2$  and  $\Pr[x = 1] = |\langle -|\psi\rangle|^2$

Example: if we measure  $|+\rangle$  (see picture) in either basis, we get output label 0 with probability  $\frac{2+\sqrt{2}}{4} \approx 0.85$

53 Cryptography and Quantum Key Distribution

53



54