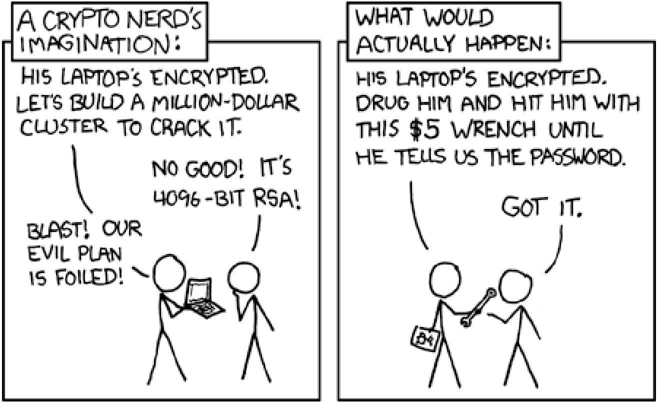

A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Quantum Key Distribution

Foundational Aspects of Quantum Mechanics

Simon Hirscher & Max Snijders

01-02-2017

Contents

The setting

Alice and Bob

$$\mathrm{ENC} : \underbrace{\{\text{plaintexts}\}}_{\cong\, \mathbb{N}} \xrightarrow{\text{bijective}} \underbrace{\{\text{ciphertexts}\}}_{\cong\, \mathbb{N}}$$

$$\mathrm{ENC} : \underset{\cong \mathbb{N}}{\{\text{plaintexts}\}} \xrightarrow{\text{bijective}} \underset{\cong \mathbb{N}}{\{\text{ciphertexts}\}}$$

- Encryption function hard to reverse for a 3rd party.

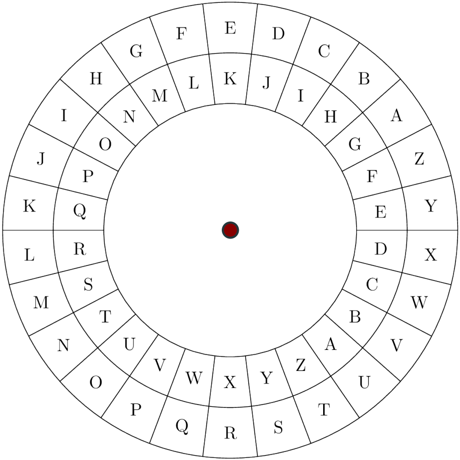
$$\mathrm{ENC} : \underbrace{\{\text{plaintexts}\}}_{\cong \mathbb{N}} \xrightarrow{\text{bijective}} \underbrace{\{\text{ciphertexts}\}}_{\cong \mathbb{N}}$$

- Encryption function hard to reverse for a 3rd party.
- Symmetric (*shared secret*)

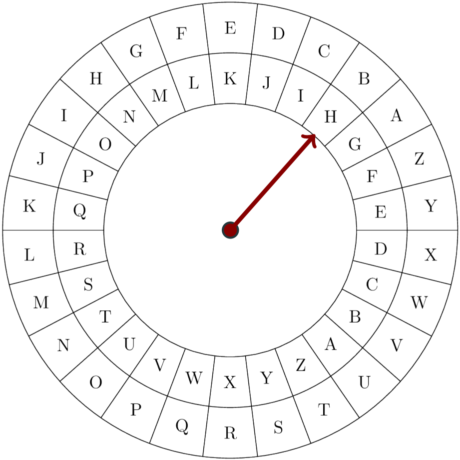
$$\mathrm{ENC} : \underbrace{\{\text{plaintexts}\}}_{\cong \mathbb{N}} \xrightarrow{\text{bijective}} \underbrace{\{\text{ciphertexts}\}}_{\cong \mathbb{N}}$$

- Encryption function hard to reverse for a 3rd party.
- Symmetric (*shared secret*)
- Asymmetric (*public/private key*)

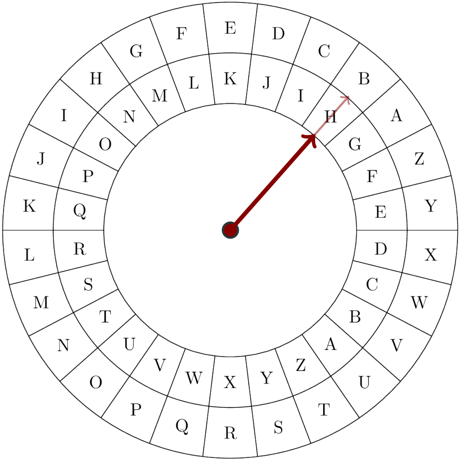
Shifting Caesar Cipher

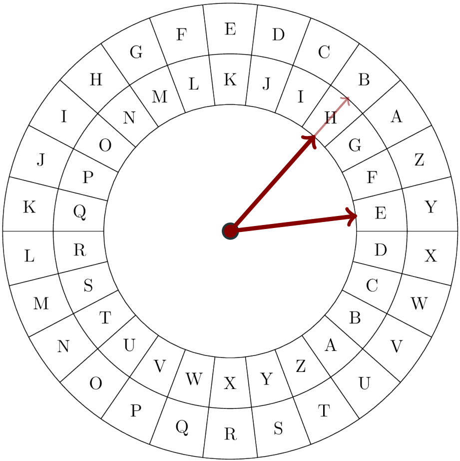


Shifting Caesar Cipher

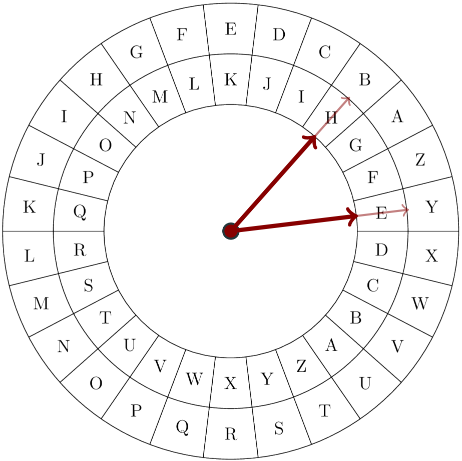


Shifting Caesar Cipher

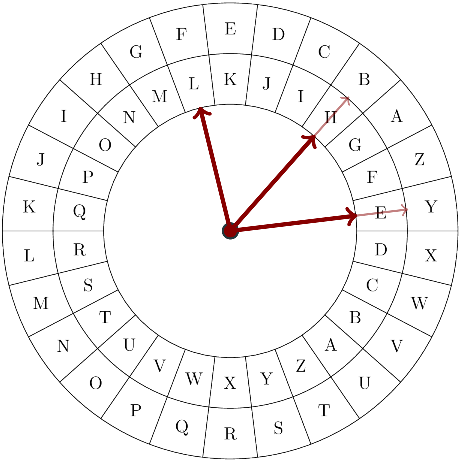



E D C B A Z Y X W V U T S R Q P O N M L K J I H G F
K J I H G F E D C B A Z Y X W V U T S R Q P O N M L

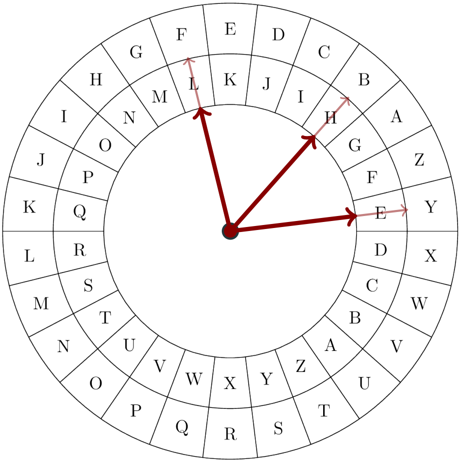
Shifting Caesar Cipher



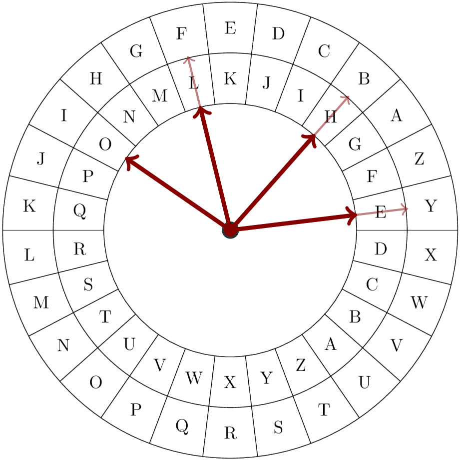
Shifting Caesar Cipher



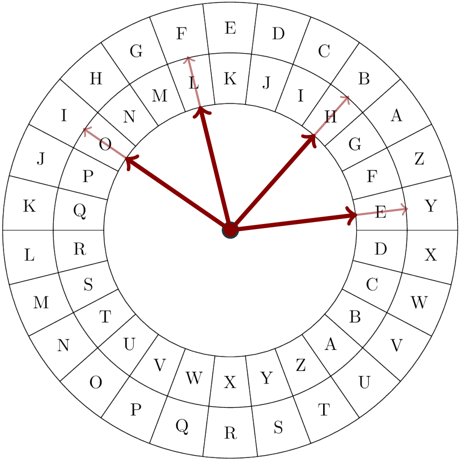
Shifting Caesar Cipher



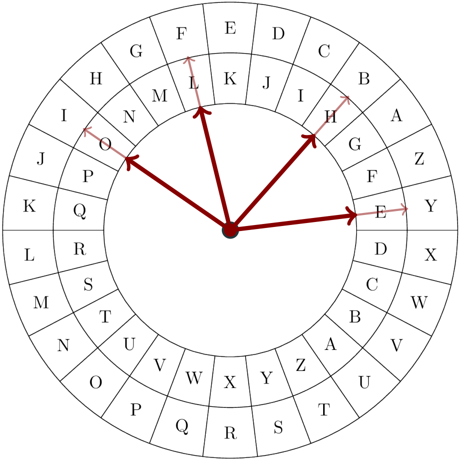
Shifting Caesar Cipher



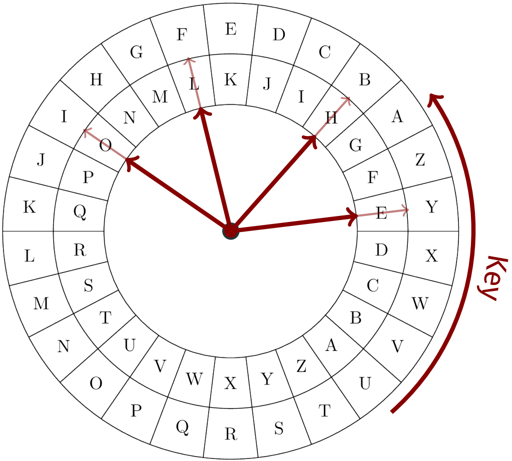
Shifting Caesar Cipher

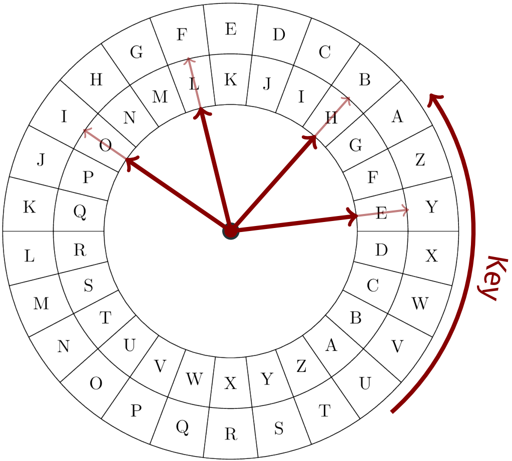


Shifting Caesar Cipher

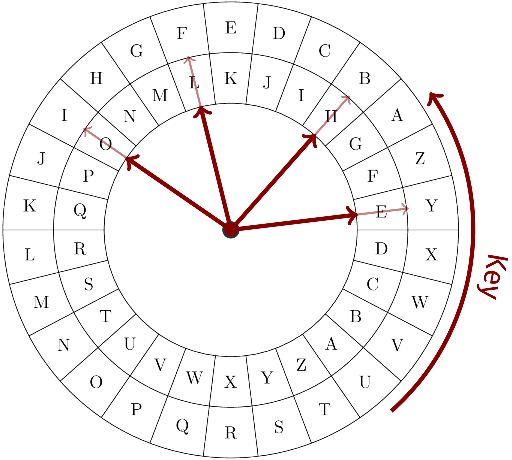


Shifting Caesar Cipher

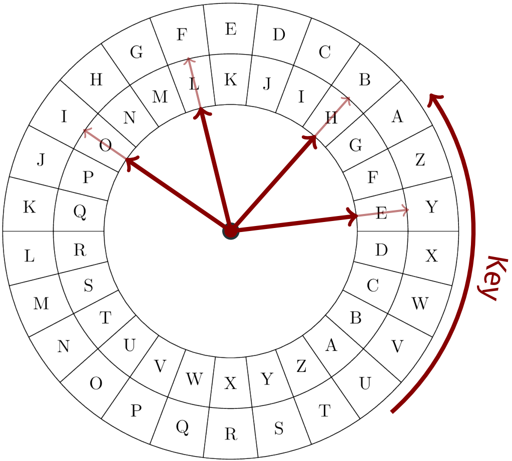




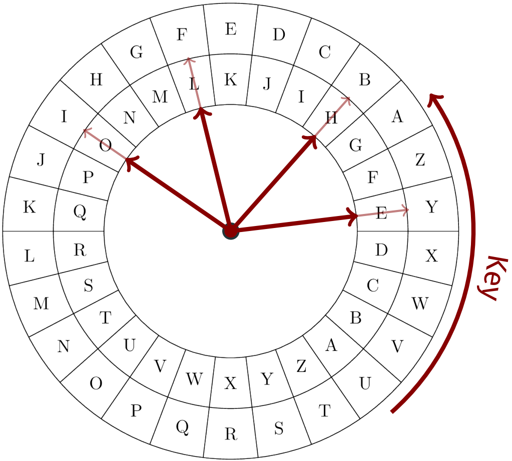
- "HELLO" $\rightarrow$ "BYFFI"



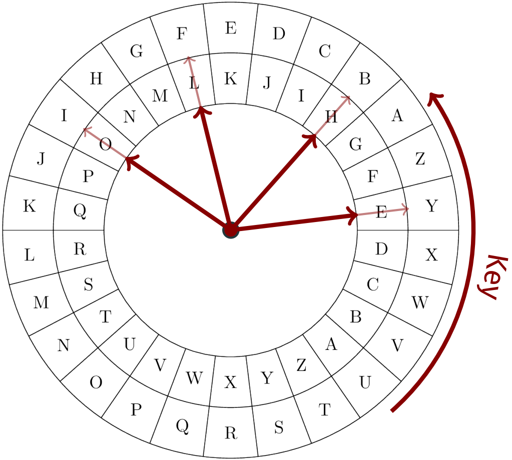
- “HELLO” → “BYFFI”
- 26 options



- “HELLO” → “BYFFI”
- 26 options
- Vulnerabilities

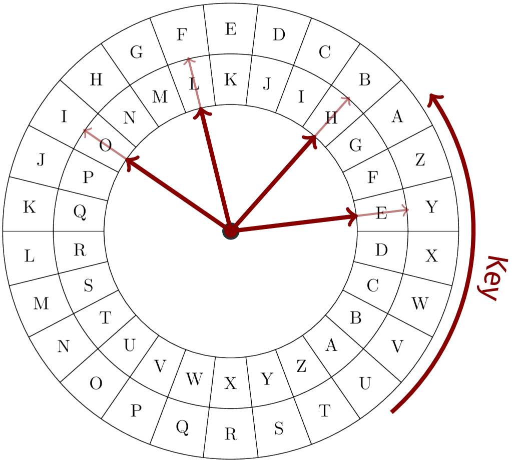


- “HELLO” → “BYFFI”
- 26 options
- Vulnerabilities:
  - Brute-force attacks



- "HELLO" $\rightarrow$ "BYFFI"
- 26 options
- Vulnerabilities:
  - Brute-force attacks
  - Frequency analysis

Shifting Caesar Cipher



- “HELLO” → “BYFFI”
- 26 options
- Vulnerabilities:
    - Brute-force attacks
    - Frequency analysis
    - Known-plaintext attacks

Every character gets mapped to a unique character

Plaintext	Ciphertext

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J
⋮	⋮

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J
⋮	⋮

- "ABBACD" $\rightarrow$ "GXXGCJ"

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J
⋮	⋮

- "ABBACD" $\rightarrow$ "GXXGCJ"
- $26 \cdot 25 \cdot 24 \cdot \ldots \cdot 1 = 26! \approx 10^{26}$ options

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J
$\vdots$	$\vdots$

- "ABBACD" $\rightarrow$ "GXXGCJ"
- $26 \cdot 25 \cdot 24 \cdot \ldots \cdot 1 = 26! \approx 10^{26}$ options
- Vulnerabilities

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J
$\vdots$	$\vdots$

- "ABBACD" $\rightarrow$ "GXXGCJ"
- $26 \cdot 25 \cdot 24 \cdot \ldots \cdot 1 = 26! \approx 10^{26}$ options
- Vulnerabilities:
  - frequency analysis.

Permutation Cipher

Every character gets mapped to a unique character

Plaintext	Ciphertext
A	G
B	X
C	C
D	J
$\vdots$	$\vdots$

- "ABBACD" $\rightarrow$ "GXXGCJ"
- $26 \cdot 25 \cdot 24 \cdot \ldots \cdot 1 = 26! \approx 10^{26}$ options
- Vulnerabilities:
  - frequency analysis.
  - known-plaintext attacks.

XOR

- Key K of n bits
- Successively apply K to blocks A of n bits of plaintext by xor'ing data and key bits: $\text{ENC}_K(A) := A \oplus K$

Bit #	1	2	3	n=4	
Plaintext	1	0	1	1	
Key	1	1	0	1	
Ciphertext	0	1	1	0	

- Decryption: $\text{DEC}_K(\text{ENC}_K(A)) := \text{ENC}_K(A) \oplus K = A \oplus K \oplus K = A$

XOR

- Key K of n bits
- Successively apply K to blocks A of n bits of plaintext by xor'ing data and key bits: $\mathrm{ENC}_K(A) := A \oplus K$

Bit #	1	2	3	n=4	5	6	7	8
Plaintext	1	0	1	1	0	0	0	1
Key	1	1	0	1	1	1	0	1
Ciphertext	0	1	1	0	1	1	0	0

- Decryption: $\mathrm{DEC}_K(\mathrm{ENC}_K(A)) := \mathrm{ENC}_K(A) \oplus K = A \oplus K \oplus K = A$

XOR

- Key K of n bits
- Successively apply K to blocks A of n bits of plaintext by xor'ing data and key bits: $\mathrm{ENC}_K(A) := A \oplus K$

Bit #	1	2	3	n=4	5	6	7	8	...
Plaintext	1	0	1	1	0	0	0	1	...
Key	1	1	0	1	1	1	0	1	...
Ciphertext	0	1	1	0	1	1	0	0	...

- Decryption: $\mathrm{DEC}_K(\mathrm{ENC}_K(A)) := \mathrm{ENC}_K(A) \oplus K = A \oplus K \oplus K = A$

XOR

- Key K of n bits
- Successively apply K to blocks A of n bits of plaintext by xor'ing data and key bits: $\text{ENC}_K(A) := A \oplus K$

Bit #	1	2	3	n=4	5	6	7	8	...
Plaintext	1	0	1	1	0	0	0	1	...
Key	1	1	0	1	1	1	0	1	...
Ciphertext	0	1	1	0	1	1	0	0	...

- Decryption: $\text{DEC}_K(\text{ENC}_K(A)) := \text{ENC}_K(A) \oplus K = A \oplus K \oplus K = A$
- Susceptible to **frequency analysis**, **known-plaintext attacks** and **brute force**.

XOR

- Key K of n bits
- Successively apply K to blocks A of n bits of plaintext by xor'ing data and key bits: $\text{ENC}_K(A) := A \oplus K$

Bit #	1	2	3	n=4	5	6	7	8	...
Plaintext	1	0	1	1	0	0	0	1	...
Key	1	1	0	1	1	1	0	1	...
Ciphertext	0	1	1	0	1	1	0	0	...

- Decryption: $\text{DEC}_K(\text{ENC}_K(A)) := \text{ENC}_K(A) \oplus K = A \oplus K \oplus K = A$
- Susceptible to **frequency analysis**, **known-plaintext attacks** and **brute force**.
- Moreover: $\text{ENC}_K(A) \oplus \text{ENC}_K(B) = A \oplus K \oplus B \oplus K = A \oplus B$

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4
Plaintext	1	0	1	1
Key	1	1	0	1
Ciphertext	0	1	1	0

Unbreakable

One-Time Pad

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5
Plaintext	1	0	1	1	1
Key	1	1	0	1	0
Ciphertext	0	1	1	0	1

Unbreakable

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5	6
Plaintext	1	0	1	1	1	0
Key	1	1	0	1	0	1
Ciphertext	0	1	1	0	1	1

Unbreakable

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5	6	...
Plaintext	1	0	1	1	1	0	...
Key	1	1	0	1	0	1	...
Ciphertext	0	1	1	0	1	1	...

Unbreakable

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5	6	...	n
Plaintext	1	0	1	1	1	0	...	1
Key	1	1	0	1	0	1	...	1
Ciphertext	0	1	1	0	1	1	...	0

Unbreakable

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5	6	...	n
Plaintext	1	0	1	1	1	0	...	1
Key	1	1	0	1	0	1	...	1
Ciphertext	0	1	1	0	1	1	...	0

Unbreakable since:

- No correlation

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5	6	...	n
Plaintext	1	0	1	1	1	0	...	1
Key	1	1	0	1	0	1	...	1
Ciphertext	0	1	1	0	1	1	...	0

Unbreakable since:

- No correlation
- Any plaintext $\iff$ any ciphertext

One-time pad = random key that **is as long as the message**, only used **once**.

Bit #	1	2	3	4	5	6	...	n
Plaintext	1	0	1	1	1	0	...	1
Key	1	1	0	1	0	1	...	1
Ciphertext	0	1	1	0	1	1	...	0

Unbreakable

The problem with exchanging the key

How do we agree on the key in the first place?

How can we do that securely?

How do we agree on the key in the first place?
How can we do that securely?

Two ways:

1. Meet in person every time

How do we agree on the key in the first place?
How can we do that securely?

Two ways:

1. Meet in person every time
2. Meet in person once

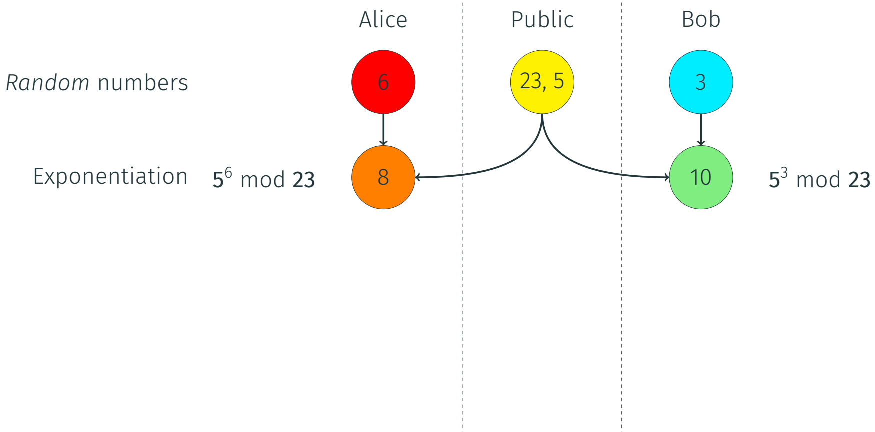
Diffie-Hellman Key Exchange (DHE)

	Alice	Public	Bob
Random numbers		23, 5	

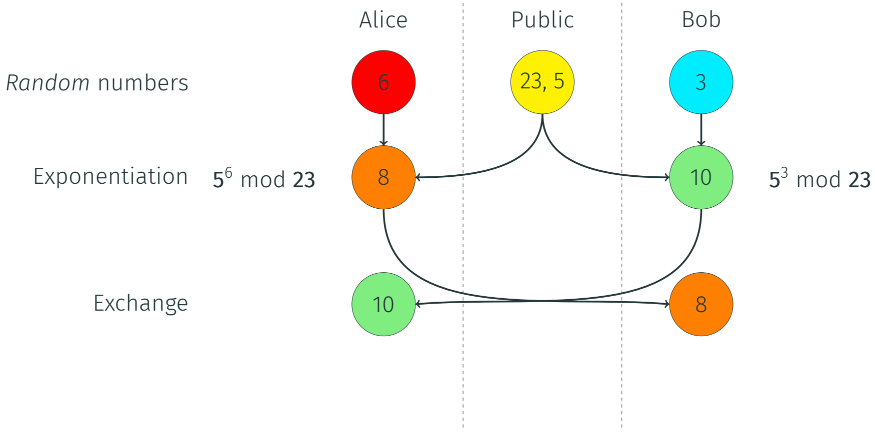
Diffie-Hellman Key Exchange (DHE)

	Alice	Public	Bob
Random numbers	6	23, 5	3

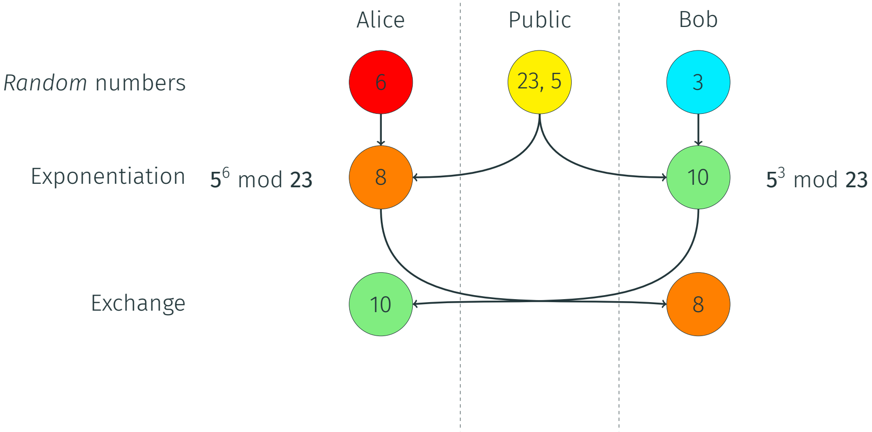
DIFFIE-HELLMAN KEY EXCHANGE (DHE)



Diffie-Hellman Key Exchange (DHE)



Diffie-Hellman Key Exchange (DHE)



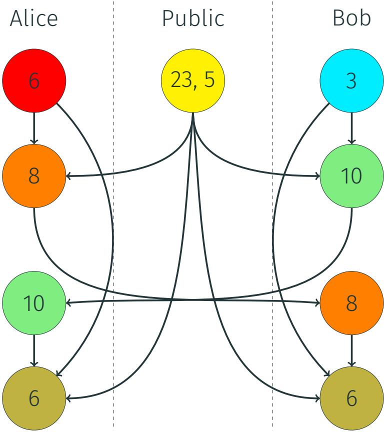
Diffie-Hellman Key Exchange (DHE)

Random numbers

Exponentiation $5^6 \bmod 23$ — $5^3 \bmod 23$

Exchange

Exponentiation $10^6 \bmod 23$ — $8^3 \bmod 23$



- Exponentiation is hard to invert classically ("discrete logarithm problem")

- Exponentiation is hard to invert classically ("discrete logarithm problem")
- Impractical for one-time pad use

- Exponentiation is hard to invert classically ("discrete logarithm problem")
- Impractical for one-time pad use
- Insecure in light of quantum algorithms:
  discrete logarithm $\sim$ integer factorization ($\rightarrow$ last week)

In QKD, quantum states carry the key information.

In QKD, quantum states carry the key information.

QKD makes use of fundamental principles of quantum mechanics:

1. Measurement changes system (unless in eigenstate of observable)

In QKD, quantum states carry the key information.

QKD makes use of fundamental principles of quantum mechanics:

1. Measurement changes system (unless in eigenstate of observable)
   - will prevent Eve from measuring quantum states without being detected

In QKD, quantum states carry the key information.

QKD makes use of fundamental principles of quantum mechanics:

1. Measurement changes system (unless in eigenstate of observable)
   - will prevent Eve from measuring quantum states without being detected
2. No-cloning theorem

In QKD, quantum states carry the key information.

QKD makes use of fundamental principles of quantum mechanics:

1. Measurement changes system (unless in eigenstate of observable)
   - will prevent Eve from measuring quantum states without being detected
2. No-cloning theorem
   - will prevent Eve from copying the quantum states for later measurement

- Consider two Hilbert spaces $H_A \cong H_B$, $\dim H_A \geq 2$.

No-cloning theorem

- Consider two Hilbert spaces $H_A \cong H_B$, $\dim H_A \geq 2$.
- Want to find unitary operator (time evolution) U such that $\forall\, |\psi\rangle \in H_A, |b\rangle \in H_B : U(|\psi\rangle \otimes |b\rangle) \stackrel{!}{=} |\psi\rangle \otimes |\psi\rangle$ (up to a phase)

No-cloning theorem

- Consider two Hilbert spaces $H_A \cong H_B$, $\dim H_A \geq 2$.
- Want to find unitary operator (time evolution) U such that $\forall \left|\psi\right\rangle \in H_A, \left|b\right\rangle \in H_B : U(\left|\psi\right\rangle \otimes \left|b\right\rangle) \overset{!}{=} \left|\psi\right\rangle \otimes \left|\psi\right\rangle$ (up to a phase)
- But then take another $\left|\phi\right\rangle \in H_A$:

$$
\begin{aligned}
\langle\psi|\phi\rangle = \langle\psi|\phi\rangle \langle b|b\rangle &= \big(\langle\psi| \otimes \langle b|\big)\big(\left|\phi\right\rangle \otimes \left|b\right\rangle\big) \\
&= \big(\langle\psi| \otimes \langle b|\big)U^\dagger U\big(\left|\phi\right\rangle \otimes \left|b\right\rangle\big) \\
&= \big(\langle\psi| \otimes \langle\psi|\big)\big(\left|\phi\right\rangle \otimes \left|\phi\right\rangle\big) = \langle\psi|\phi\rangle^2
\end{aligned}
$$

No-cloning theorem

- Consider two Hilbert spaces $H_A \cong H_B$, $\dim H_A \geq 2$.
- Want to find unitary operator (time evolution) U such that $\forall \left|\psi\right\rangle \in H_A, \left|b\right\rangle \in H_B : U(\left|\psi\right\rangle \otimes \left|b\right\rangle) \stackrel{!}{=} \left|\psi\right\rangle \otimes \left|\psi\right\rangle$ (up to a phase)
- But then take another $\left|\phi\right\rangle \in H_A$:

$$\begin{aligned}\langle\psi|\phi\rangle = \langle\psi|\phi\rangle \langle b|b\rangle &= \big(\left\langle\psi\right| \otimes \left\langle b\right|\big)\big(\left|\phi\right\rangle \otimes \left|b\right\rangle\big)\\ &= \big(\left\langle\psi\right| \otimes \left\langle b\right|\big)U^\dagger U\big(\left|\phi\right\rangle \otimes \left|b\right\rangle\big)\\ &= \big(\left\langle\psi\right| \otimes \left\langle\psi\right|\big)\big(\left|\phi\right\rangle \otimes \left|\phi\right\rangle\big) = \langle\psi|\phi\rangle^2\end{aligned}$$

$\implies$ $\langle\psi|\phi\rangle = 1$, i.e. identical, or $\langle\psi|\phi\rangle = 0$
$\implies$ Can never work with different, non-orthogonal states

The BB-84 Protocol

Charles Bennet & Gilles Brassard, 1984

Core concept: measurements affect states

$\rightarrow$ Blackoard

1. Alice chooses bit value $b_i \in \{0, 1\}$

Bit # i	1	2	3	4
Bit b_i	0	1	1	0

1. Alice chooses bit value $b_i \in \{0, 1\}$
2. Alice picks basis $A_i \in \{+, \times\}$

Bit # i	1	2	3	4
Bit b_i	0	1	1	0
Basis A_i	$\times$	$\times$	$\times$	$+$

1. Alice chooses bit value $b_i \in \{0, 1\}$
2. Alice picks basis $A_i \in \{+, \times\}$
3. Alice encodes b_i as polarized photon using basis

Bit # i	1	2	3	4
Bit b_i	0	1	1	0
Basis A_i	$\times$	$\times$	$\times$	$+$
Alice sends	$\nearrow\!\!\!\!\swarrow$	$\nwarrow\!\!\!\!\searrow$	$\nwarrow\!\!\!\!\searrow$	$\leftrightarrow$

Polarization	$\nearrow\!\!\!\!\swarrow$	$\nwarrow\!\!\!\!\searrow$	$\leftrightarrow$	$\updownarrow$
Bit value	0	1	0	1

1. Alice chooses bit value $b_i \in \{0, 1\}$
2. Alice picks basis $A_i \in \{+, \times\}$
3. Alice encodes b_i as polarized photon using basis
4. Bob picks basis $B_i \in \{+, \times\}$

Bit # i	1	2	3	4
Bit b_i	0	1	1	0
Basis A_i	$\times$	$\times$	$\times$	$+$
Alice sends	$\swarrow\!\!\!\nearrow$	$\nwarrow\!\!\!\searrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$
Basis B_i	$+$	$\times$	$+$	$+$

Polarization	$\swarrow\!\!\!\nearrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$	$\updownarrow$
Bit value	0	1	0	1

1. Alice chooses bit value $b_i \in \{0, 1\}$
2. Alice picks basis $A_i \in \{+, \times\}$
3. Alice encodes b_i as polarized photon using basis
4. Bob picks basis $B_i \in \{+, \times\}$
5. Bob measures photon polarization using the basis

Bit # i	1	2	3	4
Bit b_i	0	1	1	0
Basis A_i	$\times$	$\times$	$\times$	$+$
Alice sends	$\swarrow\!\!\!\nearrow$	$\nwarrow\!\!\!\searrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$
Basis B_i	$+$	$\times$	$+$	$+$
Bob sees	?	1	?	0

Polarization	$\swarrow\!\!\!\nearrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$	$\updownarrow$
Bit value	0	1	0	1

1. Alice chooses bit value $b_i \in \{0,1\}$
2. Alice picks basis $A_i \in \{+,\times\}$
3. Alice encodes b_i as polarized photon using basis
4. Bob picks basis $B_i \in \{+,\times\}$
5. Bob measures photon polarization using the basis
6. Alice & Bob exchange list of bases over classical channel.

Bit # i	1	2	3	4
Bit b_i	0	1	1	0
Basis A_i	$\times$	$\times$	$\times$	$+$
Alice sends	$\swarrow\!\!\!\nearrow$	$\nwarrow\!\!\!\searrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$
Basis B_i	$+$	$\times$	$+$	$+$
Bob sees	?	1	?	0

Polarization	$\swarrow\!\!\!\nearrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$	$\updownarrow$
Bit value	0	1	0	1

1. Alice chooses bit value $b_i \in \{0, 1\}$
2. Alice picks basis $A_i \in \{+, \times\}$
3. Alice encodes b_i as polarized photon using basis
4. Bob picks basis $B_i \in \{+, \times\}$
5. Bob measures photon polarization using the basis
6. Alice & Bob exchange list of bases over classical channel. If bases A_i & B_i match $\implies$ append bit b_i to shared key

Bit # i	1	2	3	4
Bit b_i	0	1	1	0
Basis A_i	$\times$	$\times$	$\times$	$+$
Alice sends	$\nearrow\!\!\!\swarrow$	$\nwarrow\!\!\!\searrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$
Basis B_i	$+$	$\times$	$+$	$+$
Bob sees	?	1	?	0
Shared key	-	1	-	0

Polarization	$\nearrow\!\!\!\swarrow$	$\nwarrow\!\!\!\searrow$	$\leftrightarrow$	$\updownarrow$
Bit value	0	1	0	1

Eve's detectability:

Eve's detectability:

- If all bases match Eve is not detectable

Eve's detectability:

- If all bases match Eve is not detectable
- If Alice and Bob's bases don't match Eve is not detectable

Eve's detectability:

- If all bases match Eve is not detectable
- If Alice and Bob's bases don't match Eve is not detectable
- If Alice and Bob's bases do match but Eve's is different then Bob will measure Alice's value 50% of the time.

Eve's detectability:

- If all bases match Eve is not detectable
- If Alice and Bob's bases don't match Eve is not detectable
- If Alice and Bob's bases do match but Eve's is different then Bob will measure Alice's value 50% of the time.

$\Longrightarrow$ Alice and Bob will match values when their bases match **75%** of the time.

Eve's detectability:

- If all bases match Eve is not detectable
- If Alice and Bob's bases don't match Eve is not detectable
- If Alice and Bob's bases do match but Eve's is different then Bob will measure Alice's value 50% of the time.

$\implies$ Alice and Bob will match values when their bases match **75%** of the time.

$\implies$ Alice and Bob will match values when their bases don't match **50%** of the time.

- In practice: Transmissions erroneous

- In practice: Transmissions erroneous
- Eavesdropping: 25% error rate

- In practice: Transmissions erroneous
- Eavesdropping: 25% error rate

$\Longrightarrow$ To detect Eve:

- Keep systematic error rate (**noise level** N) far below 25%
  - Eve will eavesdrop on every nth bit if error rate is $\frac{25\%}{n}$

- In practice: Transmissions erroneous
- Eavesdropping: 25% error rate

$\implies$ To detect Eve:

- Keep systematic error rate (**noise level** N) far below 25%
  - Eve will eavesdrop on every n$^{\text{th}}$ bit if error rate is $\frac{25\%}{n}$
- Compute **quantum bit error rate** E
  - $E > N \implies$ discard key
  - $E \sim N \implies$ do error correction and proceed

BB-84 – Privacy amplification

Situation:

- Alice and Bob now share a key
- Eve might still have partial knowledge of the key

Situation:

- Alice and Bob now share a key
- Eve might still have partial knowledge of the key

Reduce her knowledge through **privacy amplification**:

Situation:

- Alice and Bob now share a key
- Eve might still have partial knowledge of the key

Reduce her knowledge through **privacy amplification**:

- Publicly announce positions i, j of two bits b_i, b_j

Situation:

- Alice and Bob now share a key
- Eve might still have partial knowledge of the key

Reduce her knowledge through **privacy amplification**:

- Publicly announce positions i, j of two bits b_i, b_j
- Replace bit i with XOR(b_i, b_j), discard bit j.

Situation:

- Alice and Bob now share a key
- Eve might still have partial knowledge of the key

Reduce her knowledge through **privacy amplification**:

- Publicly announce positions i, j of two bits b_i, b_j
- Replace bit i with XOR(b_i, b_j), discard bit j.

$\implies$ Eve's average knowledge of the key **decreases** at the expense of decreasing the key length.

Situation:

- Alice and Bob now share a key
- Eve might still have partial knowledge of the key

Reduce her knowledge through **privacy amplification**:

- Publicly announce positions i, j of two bits b_i, b_j
- Replace bit i with XOR(b_i, b_j), discard bit j.

$\Longrightarrow$ Eve's average knowledge of the key **decreases** at the expense of decreasing the key length. $\rightarrow$ blackboard

- Conceived of by Artur Ekert (Oxford) in 1991
- Difference to BB-84: Source inbetween Alice and Bob produces pairs of entangled photons in state

$$|\psi\rangle := \frac{1}{\sqrt{2}}(\uparrow\uparrow + \rightarrow\rightarrow) = \frac{1}{\sqrt{2}}(\nearrow\nearrow + \nwarrow\nwarrow)$$

- If Alice and Bob choose the same basis ($+$ or $\times$) $\implies$ measurements agree

E-91 – Security: How to rule out that Eve is listening?

- Make sure Bell's inequality is violated when their bases don't agree.

E-91 – Security: How to rule out that Eve is listening?

- Make sure Bell's inequality is violated when their bases don't agree.
- To this end: Introduce another basis, e.g.
  - Alice's bases: $a_1 = +, a_2 = \times$
  - Bob's bases: $b_1 = +, b_2 = (+ \text{ rotated by } \frac{\pi}{8})$
  - Alice and Bob match 25% of the time
  - Eve will match w/ Bob 50% of the time

E-91 – Security: How to rule out that Eve is listening?

- Make sure Bell's inequality is violated when their bases don't agree.
- To this end: Introduce another basis, e.g.
  - Alice's bases: $a_1 = +, a_2 = \times$
  - Bob's bases: $b_1 = +, b_2 = (+ \text{ rotated by } \frac{\pi}{8})$
  - Alice and Bob match 25% of the time
  - Eve will match w/ Bob 50% of the time
- Bell's inequality ($a_i, b_i \in \{\pm 1\}$):

$$
\begin{aligned}
1 \overset{\text{classically}}{\geq} \; & \mathbb{E}(a_1 b_2) + \mathbb{E}(a_2 b_2) - \mathbb{E}(a_2 b_1) \\
&= \cos(2\theta_{a_1 b_2}) + \cos(2\theta_{a_2 b_2}) - \cos(2\theta_{a_2 b_1}) \\
&= \cos(\frac{\pi}{4}) + \cos(\frac{\pi}{4}) - \cos(\frac{\pi}{2}) = \frac{2}{\sqrt{2}} = \sqrt{2} > 1
\end{aligned}
$$

- In practice: Use 4 different bases Z_θ ($\theta =$ rotation w.r.t. $+$):
  - Alice's bases: $a_1 := Z_0 = +, a_2 := Z_{\frac{\pi}{8}}, a_3 := Z_{\frac{\pi}{4}} = \times$
  - Bob's bases: $b_1 := Z_0 = +, b_2 := Z_{\frac{\pi}{8}}, b_3 := Z_{-\frac{\pi}{8}}$
  - Alice and Bob match in 2 out of 9 cases
  - Eve will match w/ Bob only 33% of the time.

- In practice: Use 4 different bases Z_θ ($\theta =$ rotation w.r.t. $+$):
  - Alice's bases: $a_1 := Z_0 = +, a_2 := Z_{\frac{\pi}{8}}, a_3 := Z_{\frac{\pi}{4}} = \times$
  - Bob's bases: $b_1 := Z_0 = +, b_2 := Z_{\frac{\pi}{8}}, b_3 := Z_{-\frac{\pi}{8}}$
  - Alice and Bob match in 2 out of 9 cases
  - Eve will match w/ Bob only 33% of the time.
- CHSH inequality w/ $a := Z_0, a' := Z_{\frac{\pi}{4}}, b := Z_{\frac{\pi}{8}}, b' := Z_{-\frac{\pi}{8}}$

$$
\begin{aligned}
2 \overset{\text{classically}}{\geq} \; & \mathbb{E}(ab) + \mathbb{E}(ab') + \mathbb{E}(a'b) - \mathbb{E}(a'b') \\
&= \cos(2\frac{\pi}{8}) + \cos(2\frac{\pi}{8}) + \cos(2\frac{\pi}{8}) \underbrace{- \cos(2\frac{3\pi}{8})}_{\cos(2\frac{\pi}{8})} \\
&= \frac{4}{\sqrt{2}} = 2\sqrt{2} > 2
\end{aligned}
$$

More vulnerabilities in practice

- Basis choice leak
- Authentication issues
- Pseudo-randomness of basis choice

- One-time pads allow perfect encryption

Summary

- One-time pads allow perfect encryption
- Quantum key distribution allows to generate one-time pads on the fly

Summary

- One-time pads allow perfect encryption
- Quantum key distribution allows to generate one-time pads on the fly
- In theory: Principles of quantum mechanics protect Alice and Bob from Eve

- One-time pads allow perfect encryption
- Quantum key distribution allows to generate one-time pads on the fly
- In theory: Principles of quantum mechanics protect Alice and Bob from Eve
- In practice: Multitude of attack vectors outside of realm of quantum mechanics

Summary

- One-time pads allow perfect encryption
- Quantum key distribution allows to generate one-time pads on the fly
- In theory: Principles of quantum mechanics protect Alice and Bob from Eve
- In practice: Multitude of attack vectors outside of realm of quantum mechanics
- Channel authentication is yet another issue

Summary

- One-time pads allow perfect encryption
- Quantum key distribution allows to generate one-time pads on the fly
- In theory: Principles of quantum mechanics protect Alice and Bob from Eve
- In practice: Multitude of attack vectors outside of realm of quantum mechanics
- Channel authentication is yet another issue
- Bandwidth in practice:
  - 1 Mbit/s through 20km of optical fiber (Cambridge, 2008)
  - 10 kbit/s through 100km of optical fiber (Cambridge, 2008)
  - 12.7 kbits/s through 300km of optical fiber (Geneva, 2015)

The end

> "Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.
> - *Andrew S. Tanenbaum*"

Public/Private Key Authentication