Title: Quantum Cryptography: A Tale of Secrets Hidden and Revealed Through the Laws of Physics

Date: Apr 04, 2007 07:00 PM

URL: http://pirsa.org/07040020

Abstract: Sensitive information can be valuable to others - from your personal credit card numbers to state and military secrets. Throughout history, sophisticated codes have been developed in an attempt to keep important data from prying eyes. But now, new technologies are emerging based on the surprising laws of quantum physics that govern the atomic scale. These powerful techniques threaten to crack some secret codes in widespread use today and, at the same time, offer new quantum cryptographic protocols which could one day profoundly alter the way we safeguard critical information. <kw>Quantum cryptography, quantum physics, Daniel Gottesman, cryptography, one-time pad, RSA, encryption, public key, decryption, private key, quantum computer, qubit, shor\'s algorithm, quantum key distribution, QKD </kw>

Quantum Cryptography

A Tale of Secrets Hidden and Revealed Through the Laws of Quantum Physics

Presented by Daniel Gottesman, Perimeter Institute

Ciphertext:

MZFDL FAYRM LEHZI VJQVM QTNDU HZNED

VLGUD MZXPY DMTRT LEABM POHYZ DXMSD

HMEMN DTDPP MTPRN LCUUS AHFUN ZZAHO

PMELB F...

Ciphertext: “Key” is (14, 19, 1)

```
14 19 1 14 19  1 14 19 1 14  19 1 14 19 1  14 19 1 14 19  1 14 19 1 14  19 1 14 19 1
MZFDL          FAYRM         LEHZI         VJQVM          QTNDU         HZNED
14 19 1 14 19  1 14 19 1 14  19 1 14 19 1  14 19 1 14 19  1 14 19 1 14  19 1 14 19 1
VLGUD          MZXPY         DMTRT         LEABM          POHYZ         DXMSD
14 19 1 14 19  1 14 19 1 14  19 1 14 19 1  14 19 1 14 19  1 14 19 1 14  19 1 14 19 1
HMEMN          DTDPP         MTPRN         LCUUS          AHFUN         ZZAHO
14 19 1 14 19  1
PMELB          F...
```

Ciphertext: “Key” is (14, 19, 1)

```
14 19 1 14 19  1 14 19 1 14  19 1 14 19 1  14 19 1 14 19  1 14 19 1 14  19 1 14 19 1
MZFDL          FAYRM         LEHZI         VJQVM          QTNDU         HZNED
14 19 1 14 19  1 14 19 1 14  19 1 14 19 1  14 19 1 14 19  1 14 19 1 14  19 1 14 19 1
VLGUD          MZXPY         DMTRT         LEABM          POHYZ         DXMSD
14 19 1 14 19  1 14 19 1 14  19 1 14 19 1  14 19 1 14 19  1 14 19 1 14  19 1 14 19 1
HMEMN          DTDPP         MTPRN         LCUUS          AHFUN         ZZAHO
14 19 1 14 19  1
PMELB          F...
```

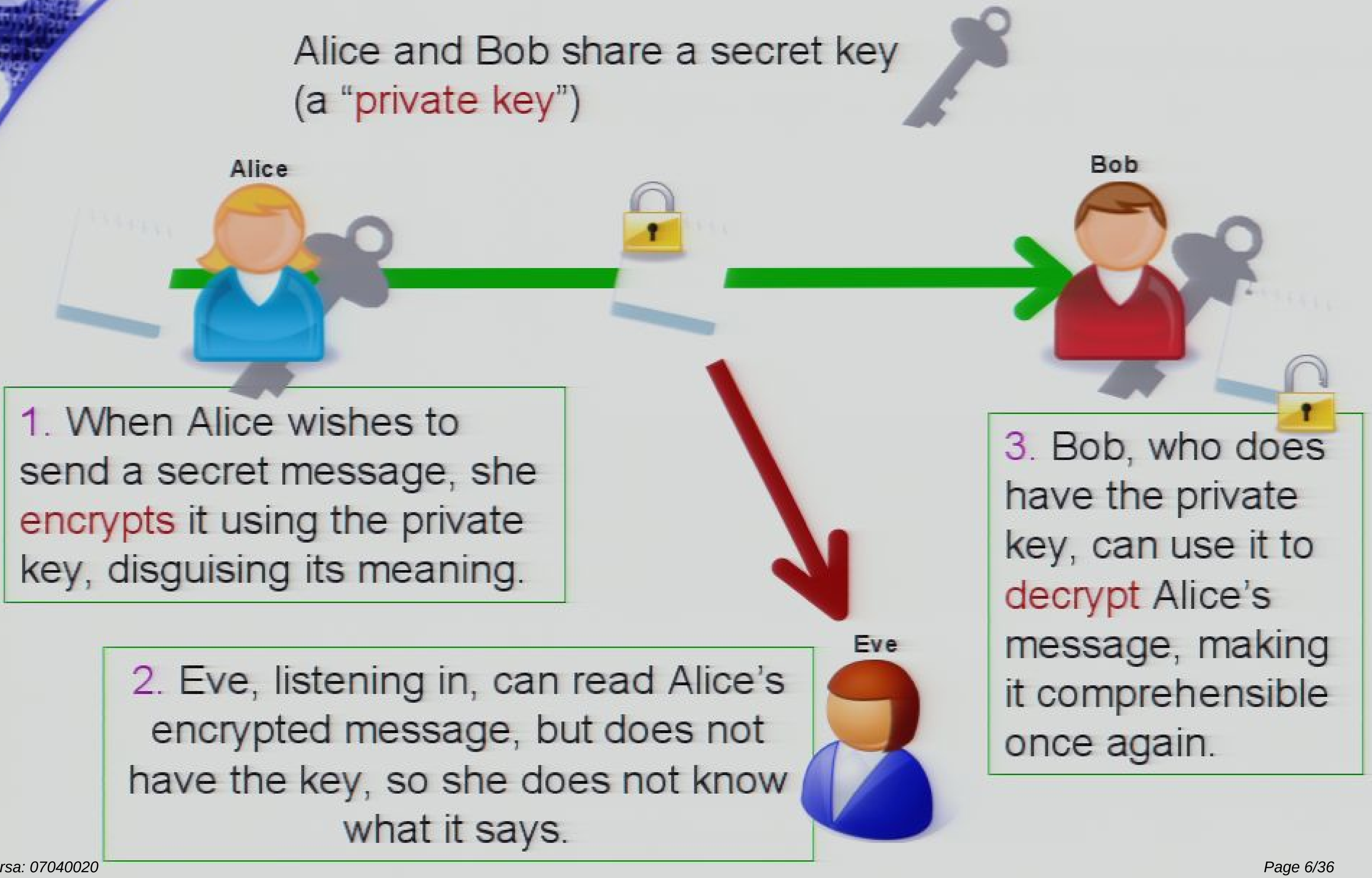
Plaintext:

AS GREGOR SAMSA AWOKE ONE MORNING
FROM UNEASY DREAMS HE FOUND HIMSELF
TRANSFORMED IN HIS BED INTO A
GIGANTIC INSECT ...

Patterns allow decoding

Private Key Cryptography

Alice and Bob share a secret key
(a “private key”)



1. When Alice wishes to send a secret message, she encrypts it using the private key, disguising its meaning.

2. Eve, listening in, can read Alice’s encrypted message, but does not have the key, so she does not know what it says.

3. Bob, who does have the private key, can use it to decrypt Alice’s message, making it comprehensible once again.

One-Time Pad

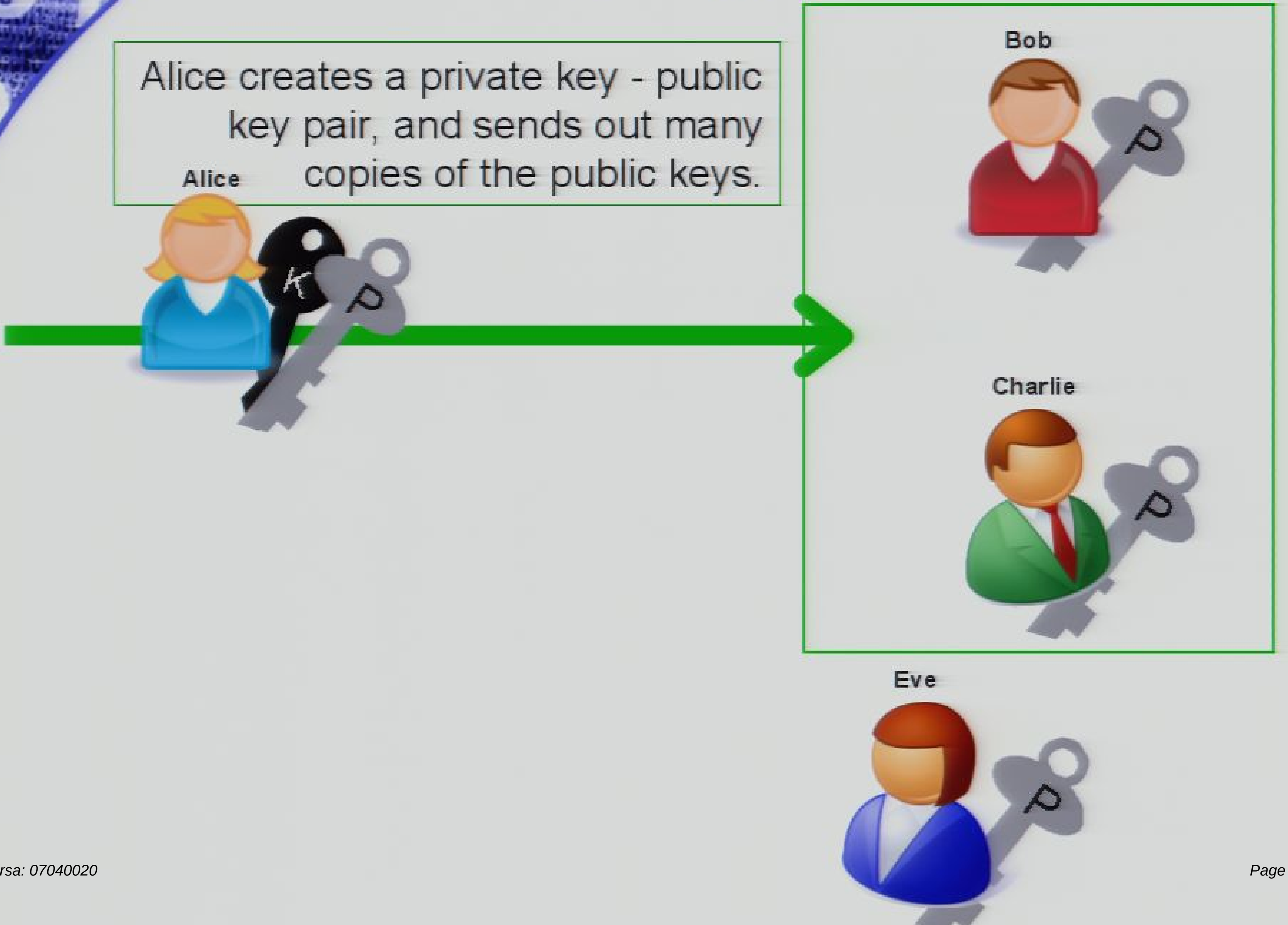
As the key becomes longer, the encryption becomes harder to break. It is unbreakable if the key is as long as the message. (The “one-time pad”)

Message	00001 10011 00111 10010 . . .
Key	00101 11011 10100 10100 . . .
Ciphertext	00100 01000 10011 00110 . . .

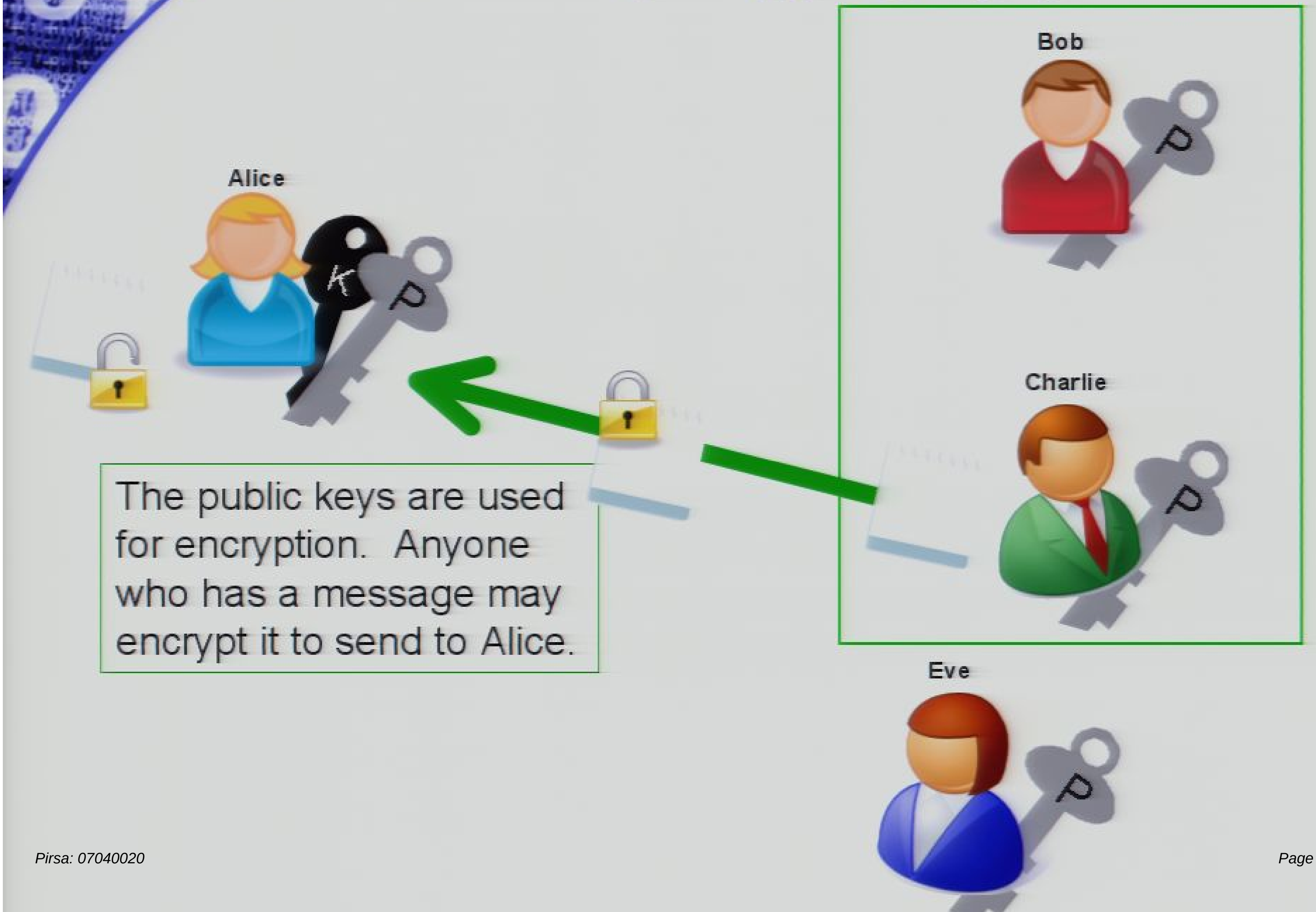
- The key in a one-time pad can only be used once.
- The key must be protected.

Public Key Cryptography

Alice creates a private key - public key pair, and sends out many copies of the public keys.

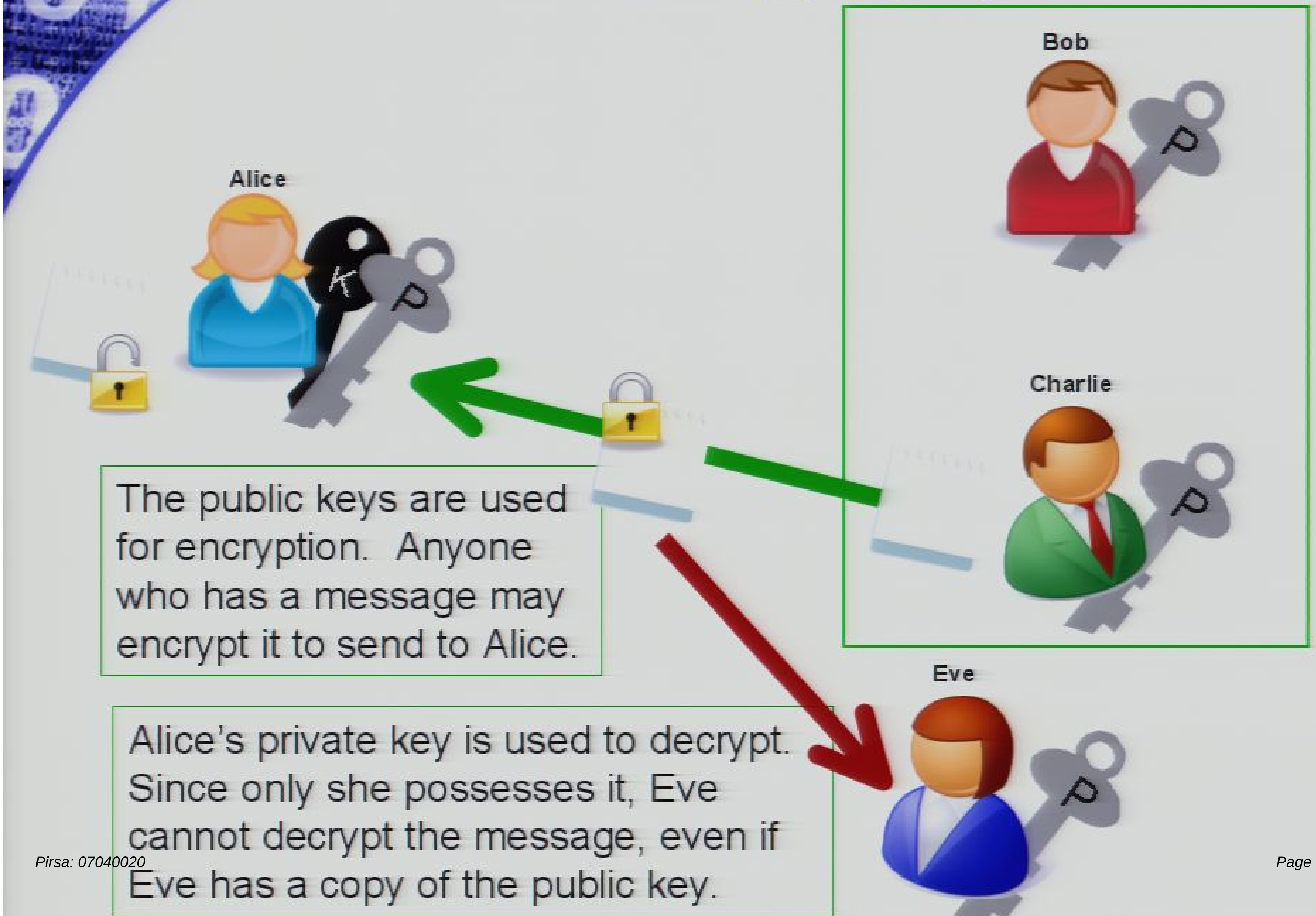


Public Key Cryptography



The public keys are used for encryption. Anyone who has a message may encrypt it to send to Alice.

Public Key Cryptography



The public keys are used for encryption. Anyone who has a message may encrypt it to send to Alice.

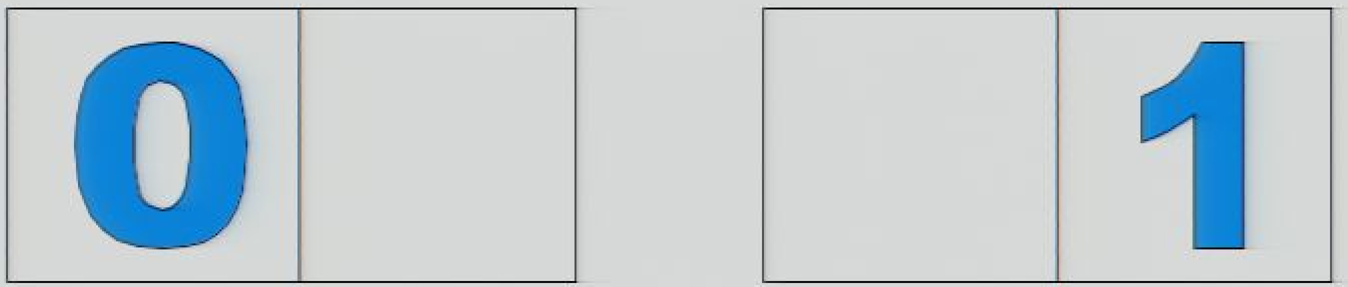
Alice's private key is used to decrypt. Since only she possesses it, Eve cannot decrypt the message, even if Eve has a copy of the public key.

RSA Public Key Encryption

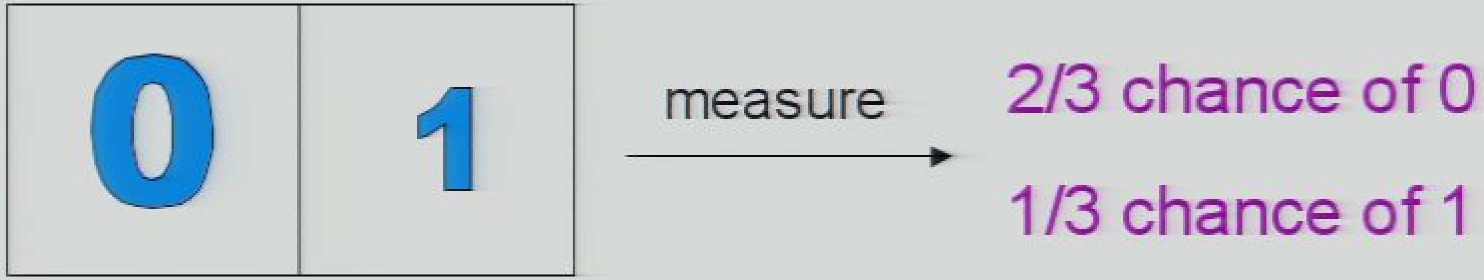
- Public encryption key is a pair of large numbers (N, e) (e.g., 300 digits long)
- To encrypt a message, Bob:
  - Converts the message to numbers less than N.
  - Raises the message to the power e.
  - Divides by N and takes the remainder.
    (Modular arithmetic)
- Private decryption key d
  Derived from prime factors of N
- To decrypt y, Alice:
  - Raises the encrypted message to the power d
  - Divides by N and takes the remainder.
- Breaking RSA believed as hard as factoring N

Quantum Bits

Regular classical bits have two possible values: 0 and 1.



Quantum bits (or “qubits”) can be both at once, a “superposition,” but when you measure a qubit, it collapses to be either a 0 or a 1 with some probability.

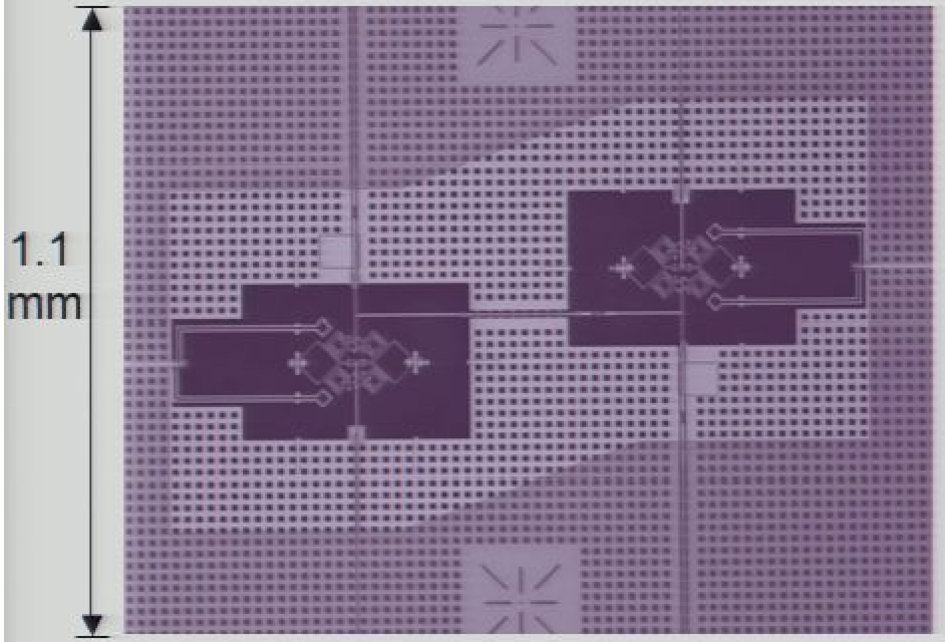


2/3 chance of 0

1/3 chance of 1

Quantum Computers

A “quantum computer” is built out of microscopic components, so small that the laws of quantum mechanics are important. A quantum computer thus uses qubits in place of the bits of a regular computer.

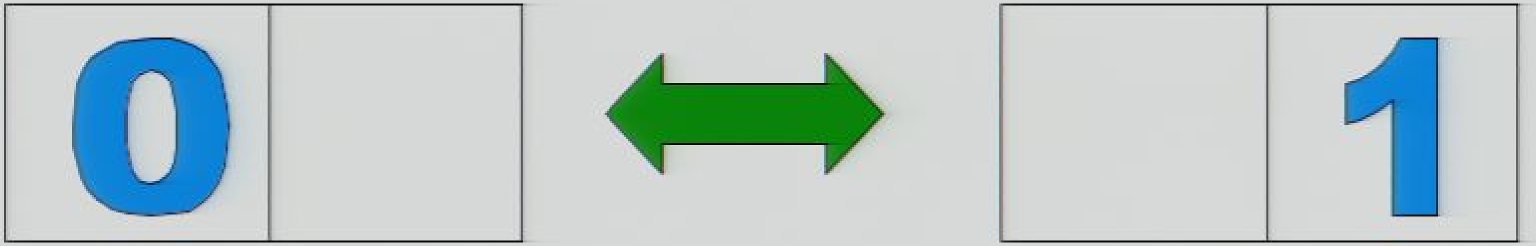


By using superpositions, quantum computers can in some sense do many computations at once, but the randomness of measurement sets severe limits.

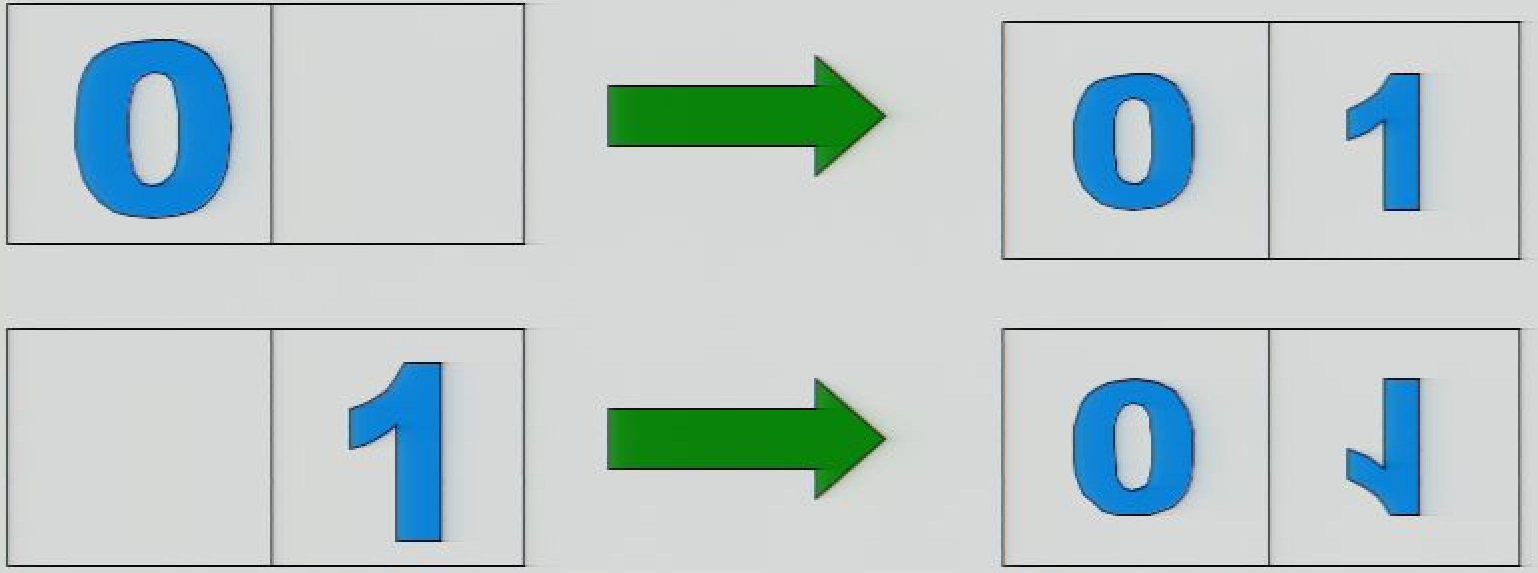
For some problems, quantum computers are vastly faster than a regular “classical” computer, but for other problems, a quantum computer offers no advantages.

Computing With Qubits

We can manipulate qubits in various ways. E.g., bit flip:



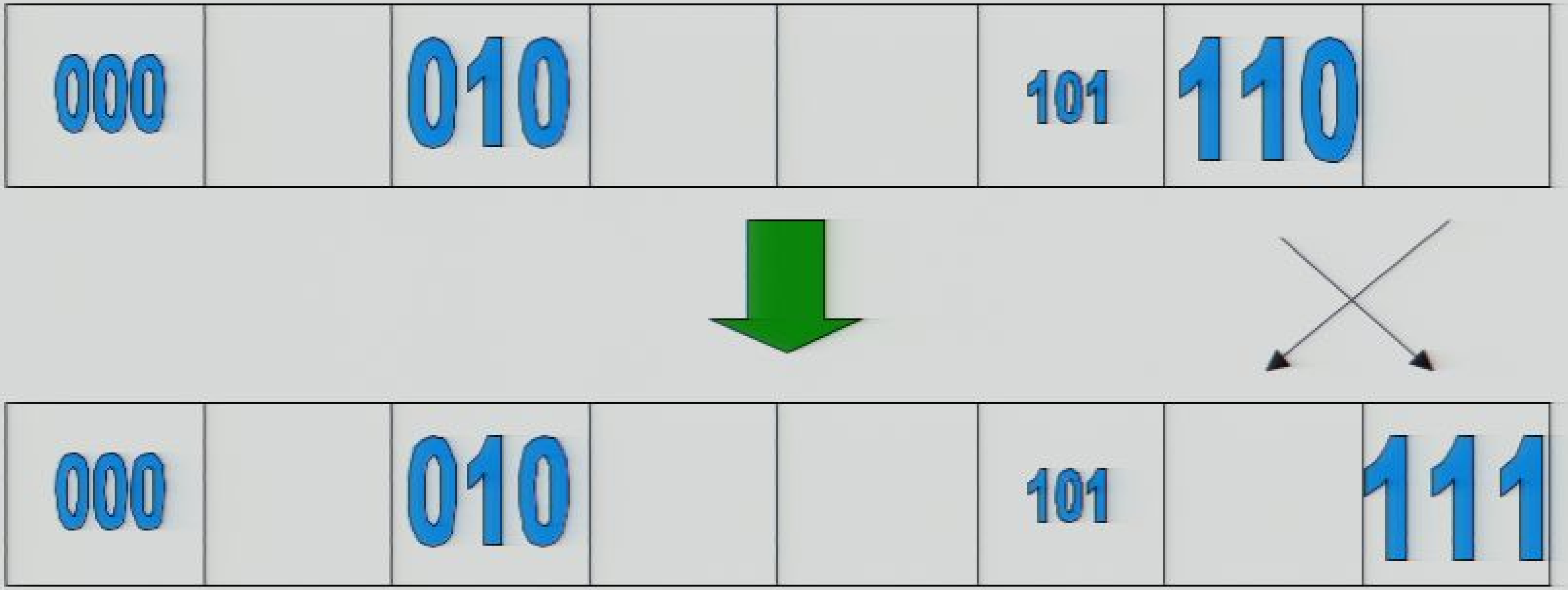
We can also create superpositions out of 0 and 1:



0 and 1 are different, so there must still be a difference bewteen them after we have altered the qubit.

Multiple Qubits

If we have many qubits, we could have a superposition of all possible values of the bits, and perform complicated computations on them:



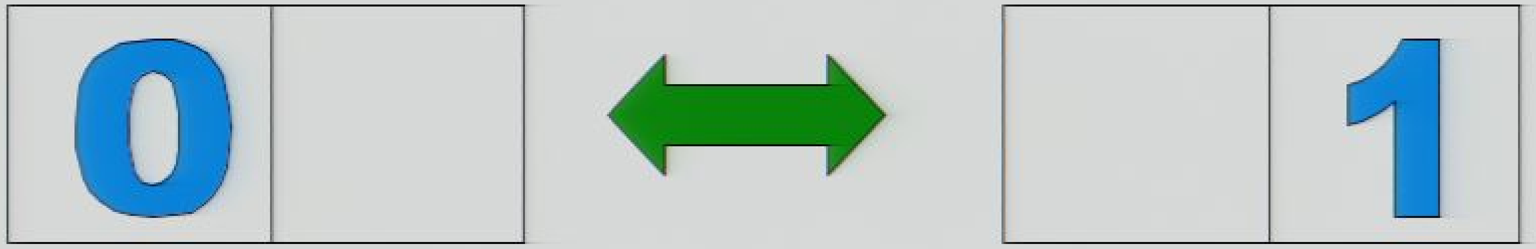
000		010			101	110	

000		010			101		111

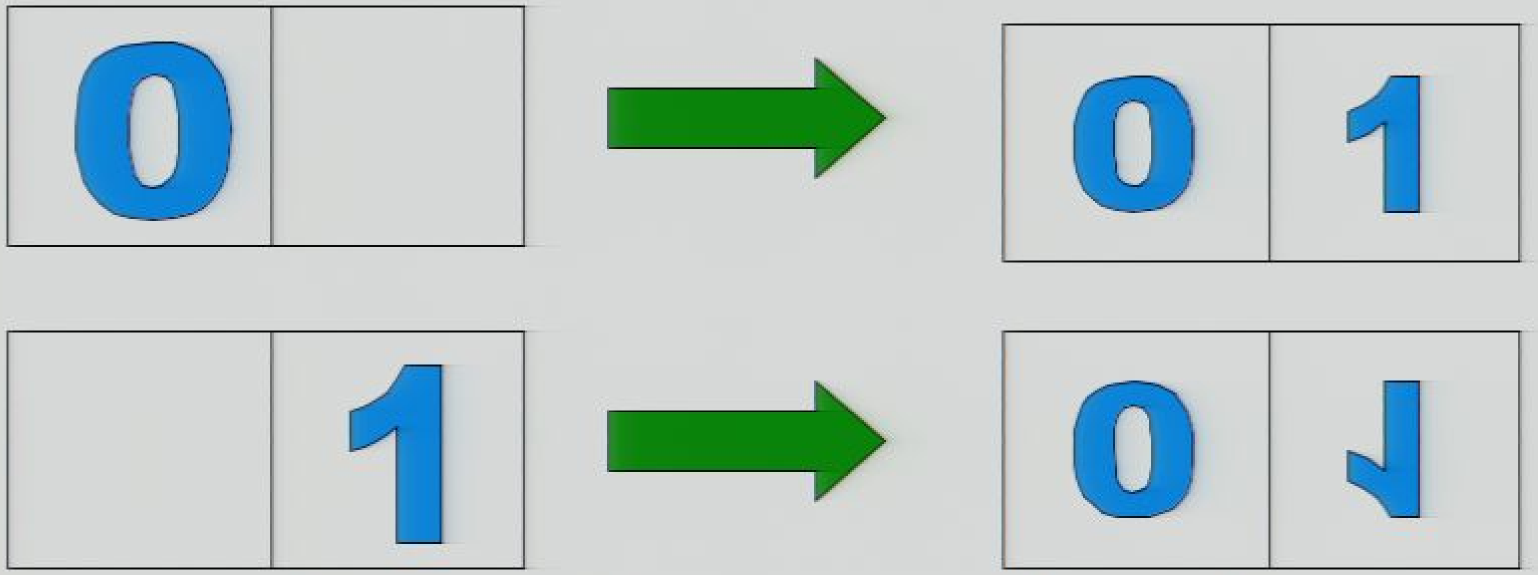
(Here we flip the third qubit only if both of the first two qubits are 1.)

Computing With Qubits

We can manipulate qubits in various ways. E.g., bit flip:



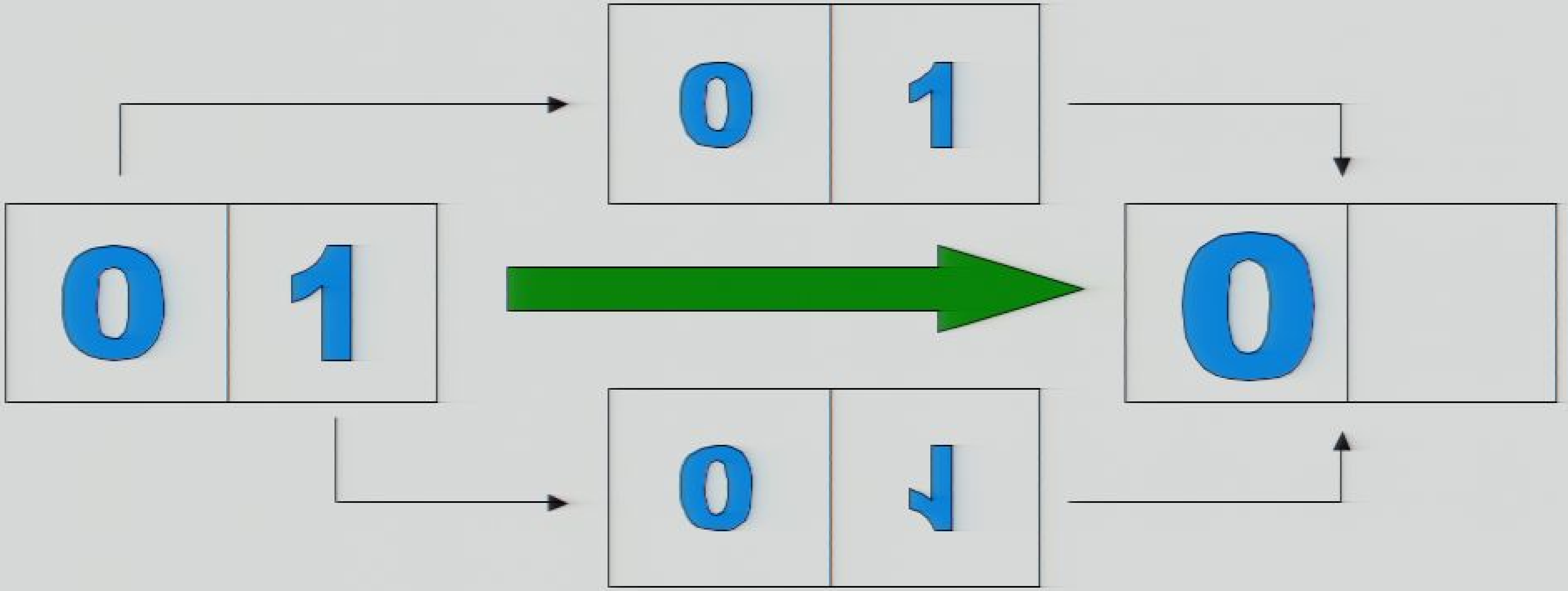
We can also create superpositions out of 0 and 1:



0 and 1 are different, so there must still be a difference bewteen them after we have altered the qubit.

Interference

What if we attempt to create a superposition, but we already had one?



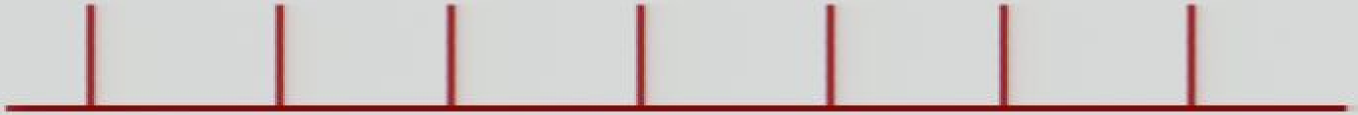
Right-side up 0s add together ("constructive interference")

Opposite 1s cancel ("destructive interference")

Shor's Algorithm

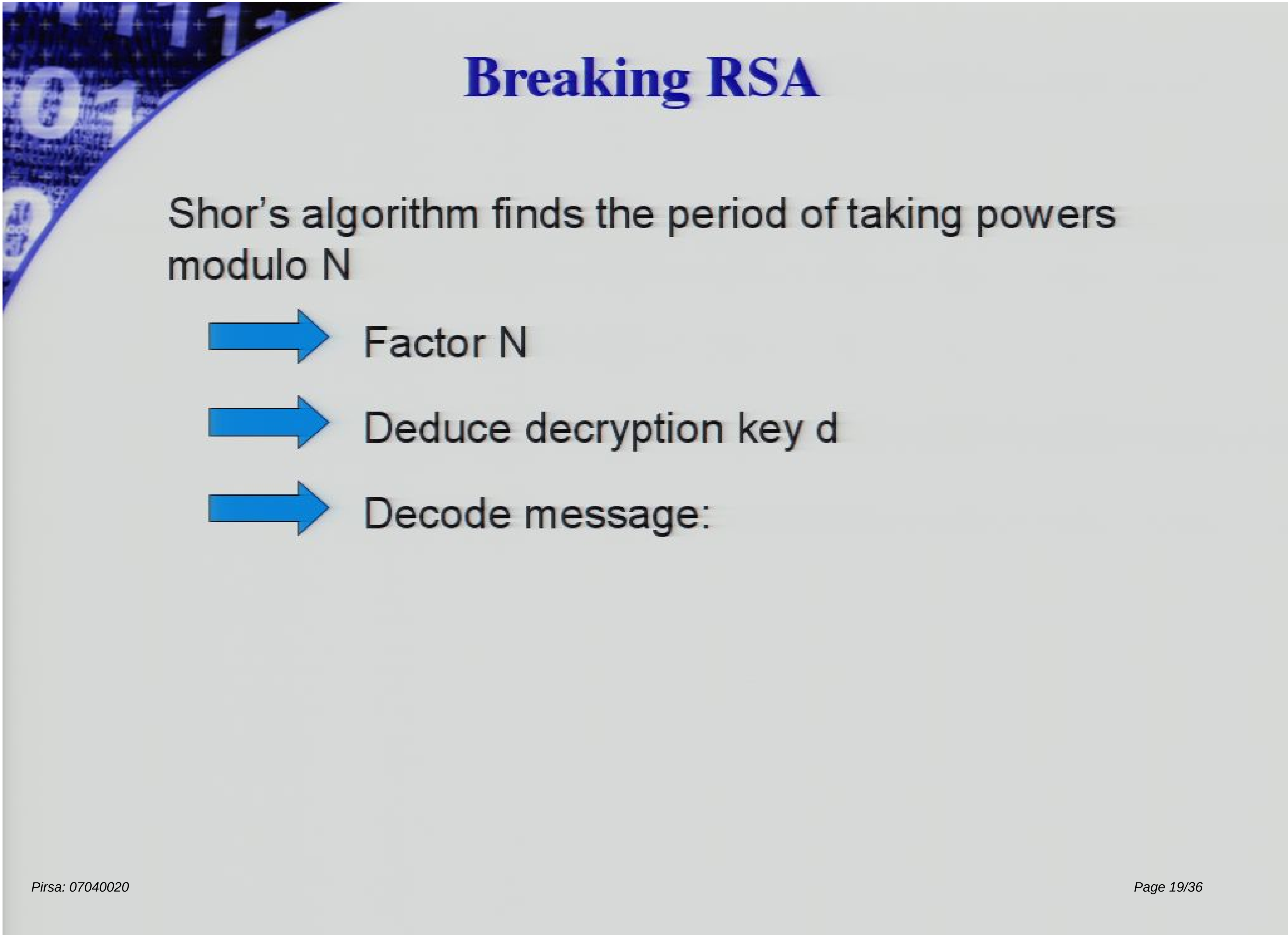
Working with modular arithmetic in RSA ensures that raising to a power periodically returns to the starting point. Shor's algorithm uses this fact to break RSA (decrypt without the private key).

Create superposition: (over each time we cycle around)



However, we don't know the starting point.

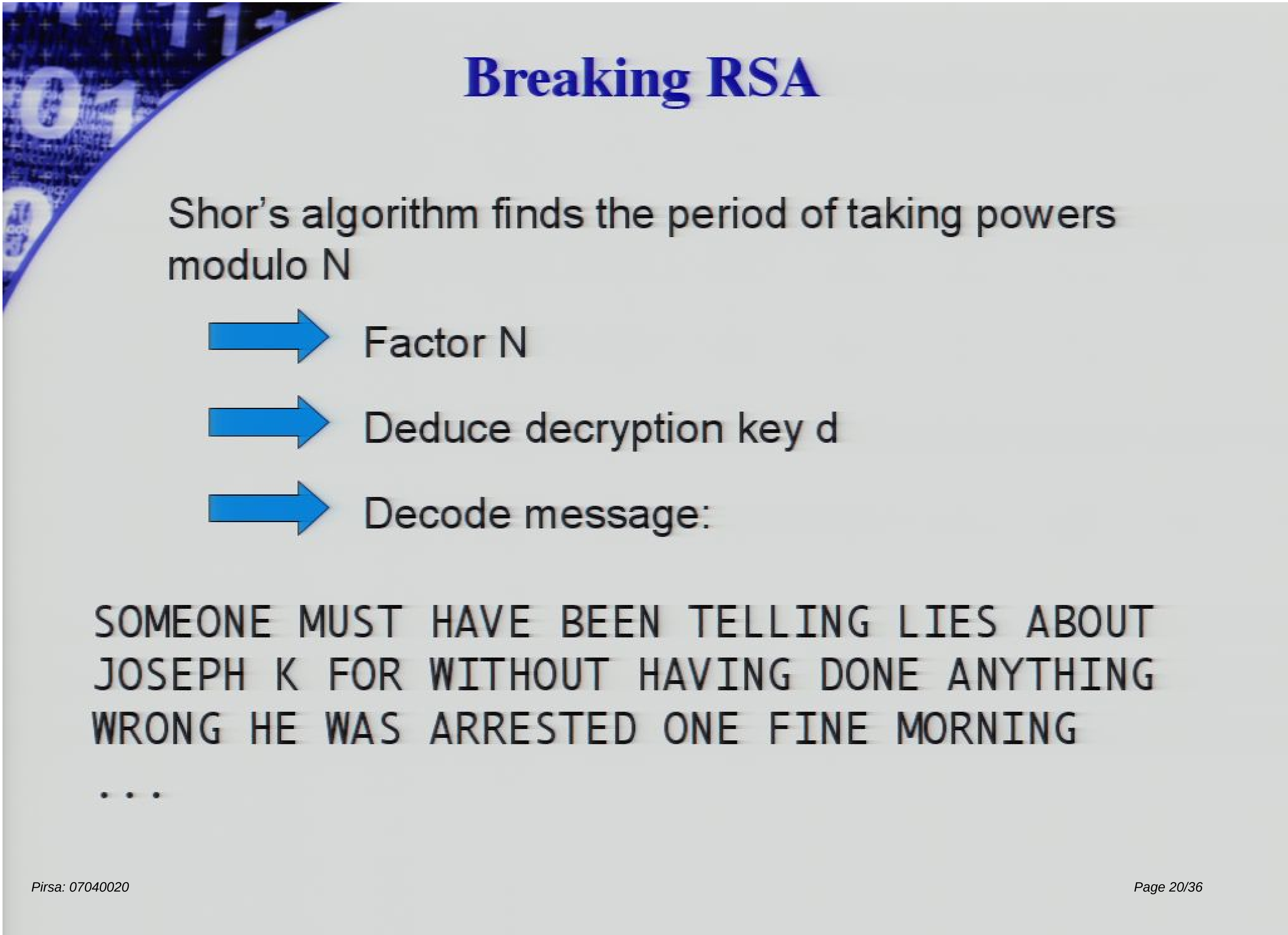
The Fourier Transform determines the period of a repeating function, so Shor's algorithm applies the Fourier transform to the superposition. Once we know the period of RSA, that tells us the value of the private key d.



Breaking RSA

Shor’s algorithm finds the period of taking powers modulo N

- Factor N
- Deduce decryption key d
- Decode message:



Breaking RSA

Shor’s algorithm finds the period of taking powers modulo N

➡ Factor N

➡ Deduce decryption key d

➡ Decode message:

```
SOMEONE MUST HAVE BEEN TELLING LIES ABOUT
JOSEPH K FOR WITHOUT HAVING DONE ANYTHING
WRONG HE WAS ARRESTED ONE FINE MORNING
...
```

Quantum Key Distribution: Alice

Alice can send quantum systems to Bob in order to share a private key with him, keeping it secret from Eve.

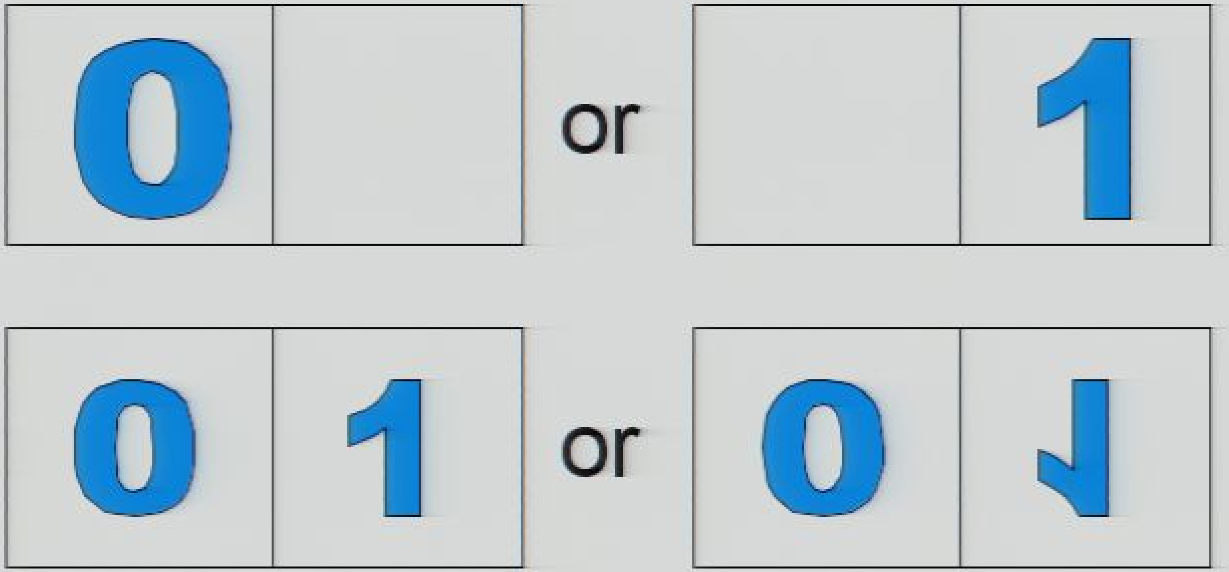
(“Quantum key distribution” = QKD)

Alice sends single particles of light (“photons”), in one of four possible quantum states:

key bit 0: key bit 1:

“Z” states:

“X” states:



Quantum Key Distribution: Bob

When Bob receives a photon, he can either measure it right away (a “Z” measurement): is it 0 or 1?

Or he can shift to a superposition (the “X” measurement) to try to distinguish the two X states.

But if he guesses wrong, his measurement result is random:

50% chance of 0

50% chance of 1

To avoid this problem, once Bob has measured, Alice and Bob compare their choices, and only keep the bit if Bob guessed the right measurement to make.

Quantum Key Distribution: Alice

Alice can send quantum systems to Bob in order to share a private key with him, keeping it secret from Eve.

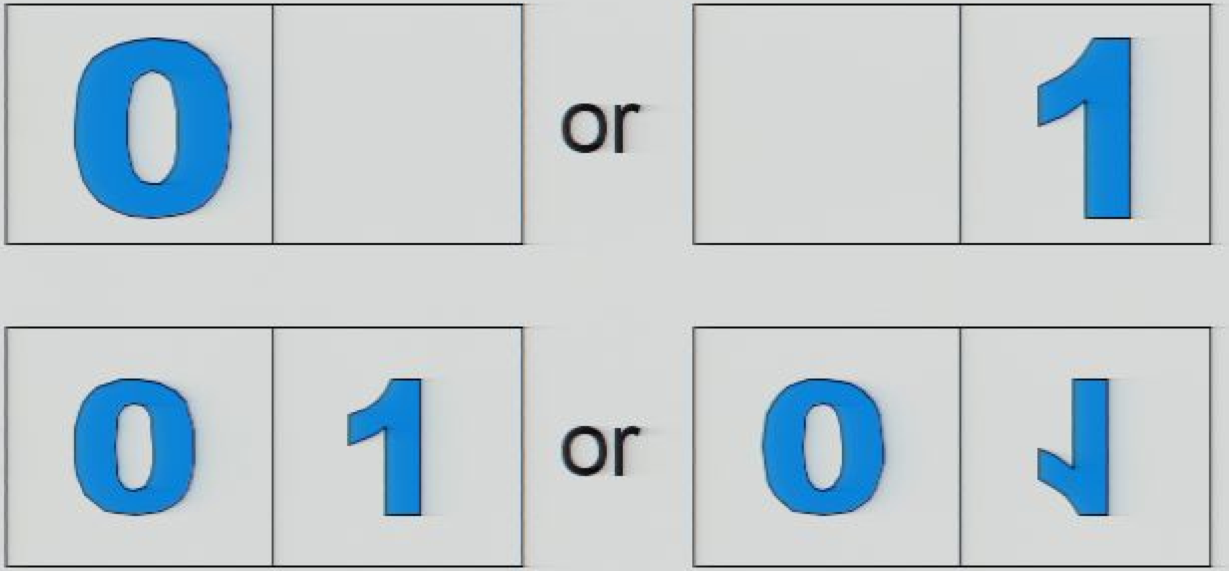
("Quantum key distribution" = QKD)

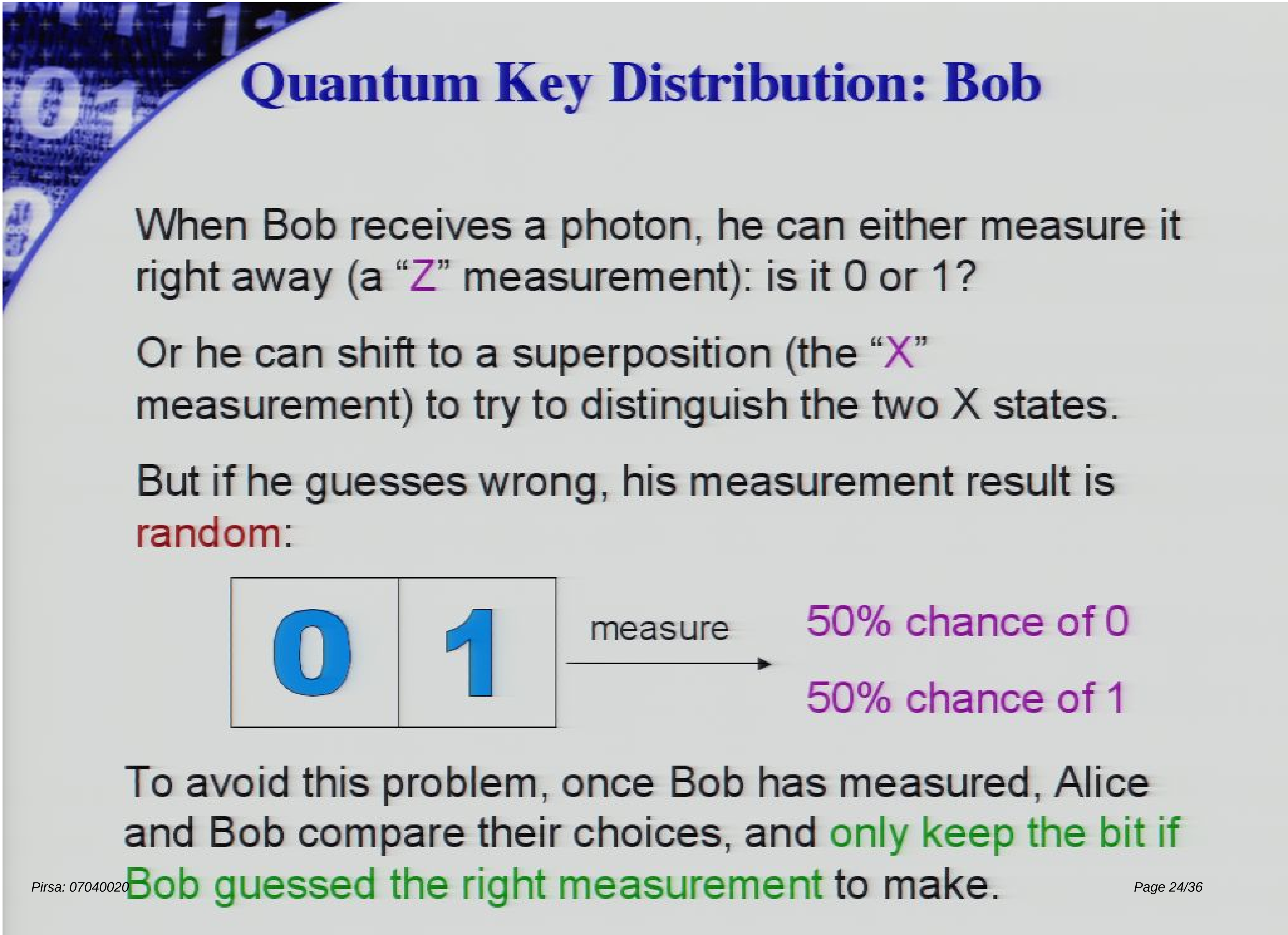
Alice sends single particles of light ("photons"), in one of four possible quantum states:

key bit 0: key bit 1:

"Z" states: or

"X" states: or





Quantum Key Distribution: Bob

When Bob receives a photon, he can either measure it right away (a "Z" measurement): is it 0 or 1?

Or he can shift to a superposition (the "X" measurement) to try to distinguish the two X states.

But if he guesses wrong, his measurement result is random:

0 | 1 $\xrightarrow{\text{measure}}$ 50% chance of 0

50% chance of 1

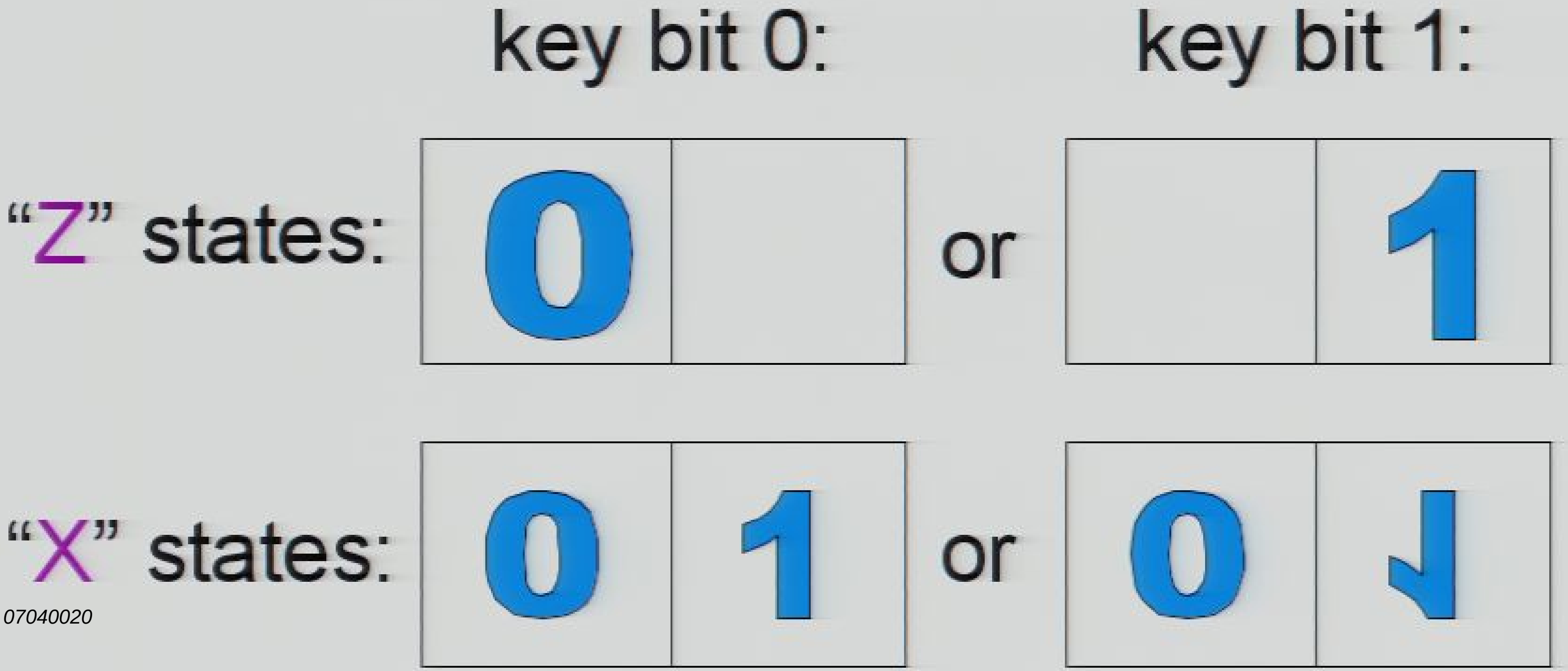
To avoid this problem, once Bob has measured, Alice and Bob compare their choices, and only keep the bit if Bob guessed the right measurement to make.

Quantum Key Distribution: Alice

Alice can send quantum systems to Bob in order to share a private key with him, keeping it secret from Eve.

("Quantum key distribution" = QKD)

Alice sends single particles of light ("photons"), in one of four possible quantum states:



key bit 0:

key bit 1:

"Z" states:

or

"X" states:

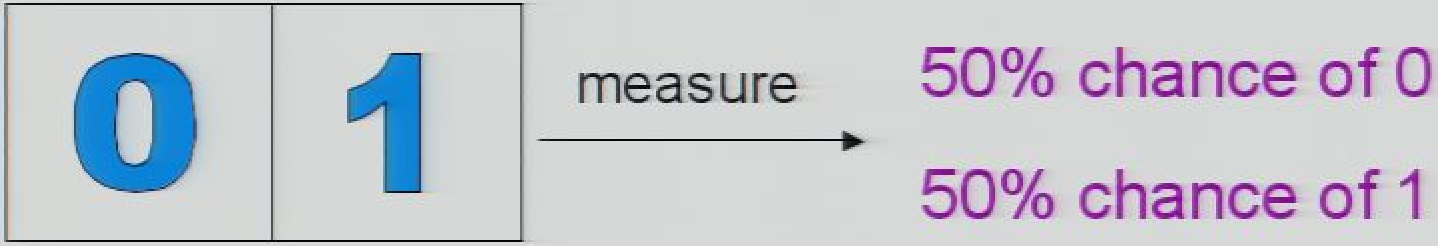
or

Quantum Key Distribution: Bob

When Bob receives a photon, he can either measure it right away (a “Z” measurement): is it 0 or 1?

Or he can shift to a superposition (the “X” measurement) to try to distinguish the two X states.

But if he guesses wrong, his measurement result is random:



50% chance of 0

50% chance of 1

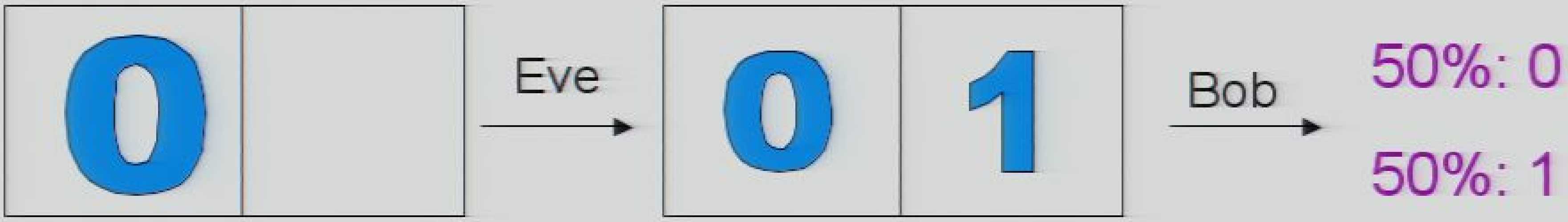
To avoid this problem, once Bob has measured, Alice and Bob compare their choices, and only keep the bit if Bob guessed the right measurement to make.

Quantum Key Distribution: Eve

Eve wants to learn the key bits too, but she faces the same problem as Bob: she does not know which measurement to make.

If Bob makes the correct measurement, his result is supposed to agree with the bit Alice sent.

If Eve guesses wrong, Eve's measurement might introduce an error into Bob's result:



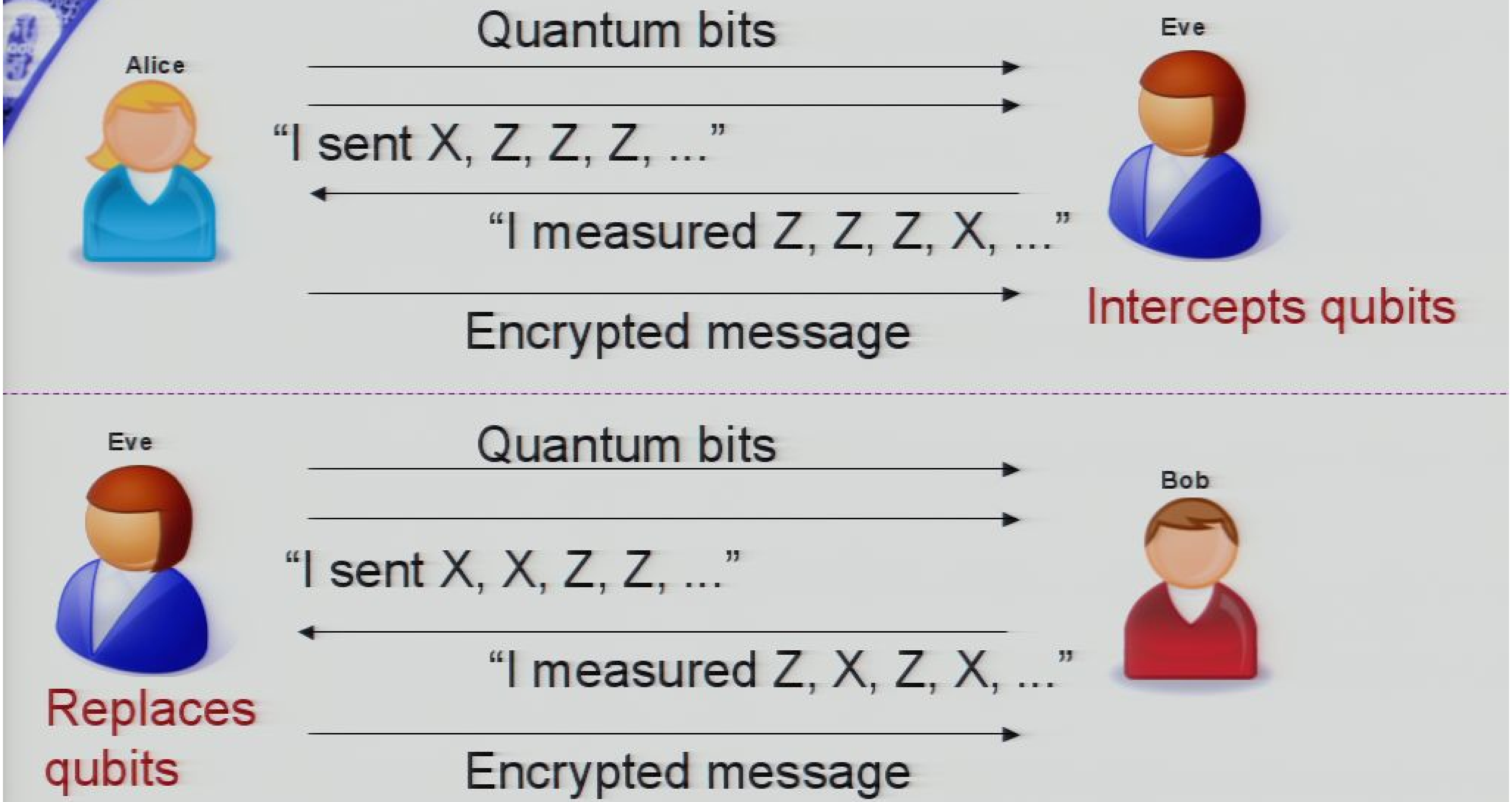
Alice and Bob can compare a few of their bits to look for errors and detect Eve!

QKD: Full Protocol

- Alice chooses random sequence of bits and X/Z
- Alice sends corresponding qubits to Bob
- Alice and Bob:
  - Compare X/Z values.
  - Discard bits where Bob chose wrong.
  - Compare bit values on a test subset.
  - Abort if error rate is too high.
  - Use an error-correcting code to fix remaining bits (there will always be some errors, even without Eve).
  - Privacy amplification: Mix up remaining bits to eliminate any little bits of information Eve might have.

Alice and Bob either end up with a secret shared key, or detect Eve's attempt at eavesdropping.

Man-in-the-Middle Attack



Quantum bits

"I sent X, Z, Z, Z, ..."

"I measured Z, Z, Z, X, ..."

Encrypted message

Intercepts qubits

Quantum bits

"I sent X, X, Z, Z, ..."

"I measured Z, X, Z, X, ..."

Encrypted message

Replaces qubits

Bob receives the message, but only after Eve has read it.

QKD and Authentication

. . .

Bob: C15

Alice: MISS. D5.

Bob: HIT. YOU SUNK MY BATTLESHIP!

. . .

Alice and Bob must "authenticate" their classical transmissions, so Eve cannot masquerade as one of them.

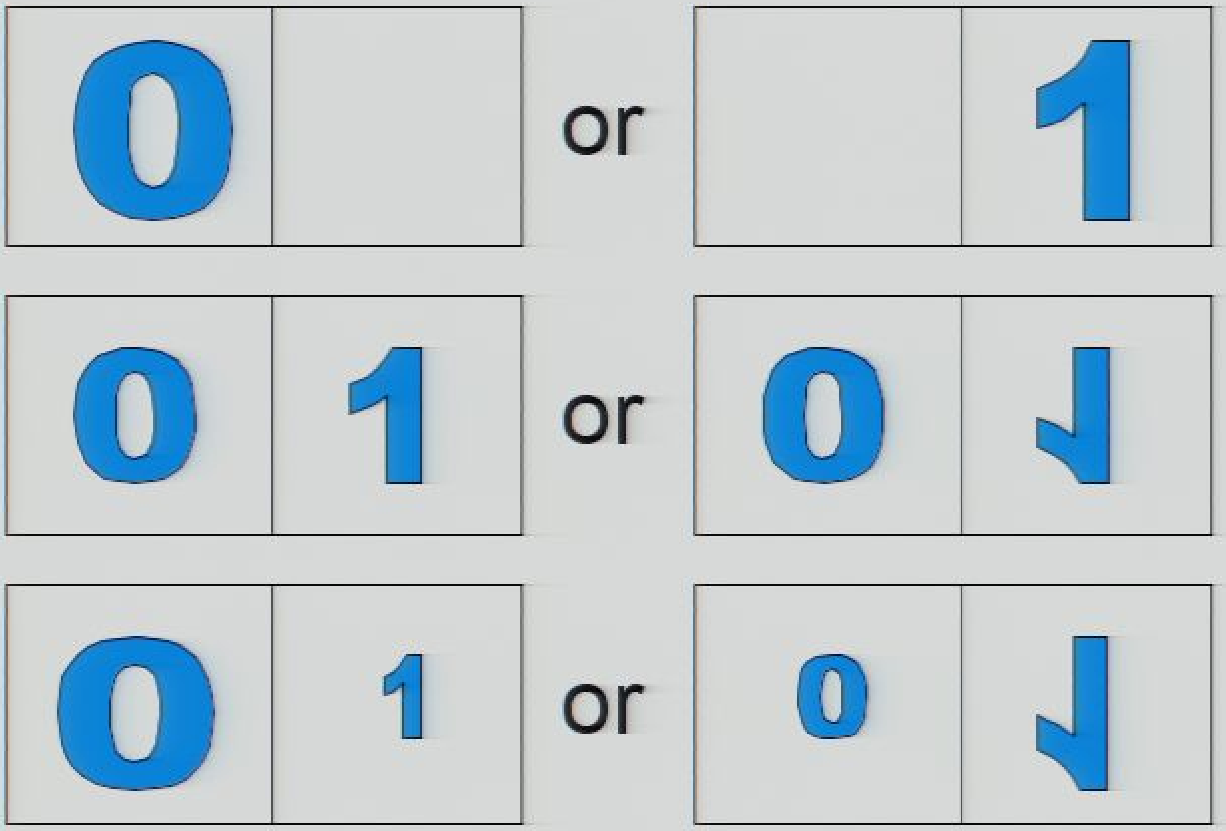
(This can be done with a small amount of shared private key.)

QKD increases the amount of private key Alice and Bob share.

(Commercial QKD products have recently become available.)

Hiding Information in a Qubit

A bit can only have two possible values, 0 and 1, but a qubit can have many possible values:

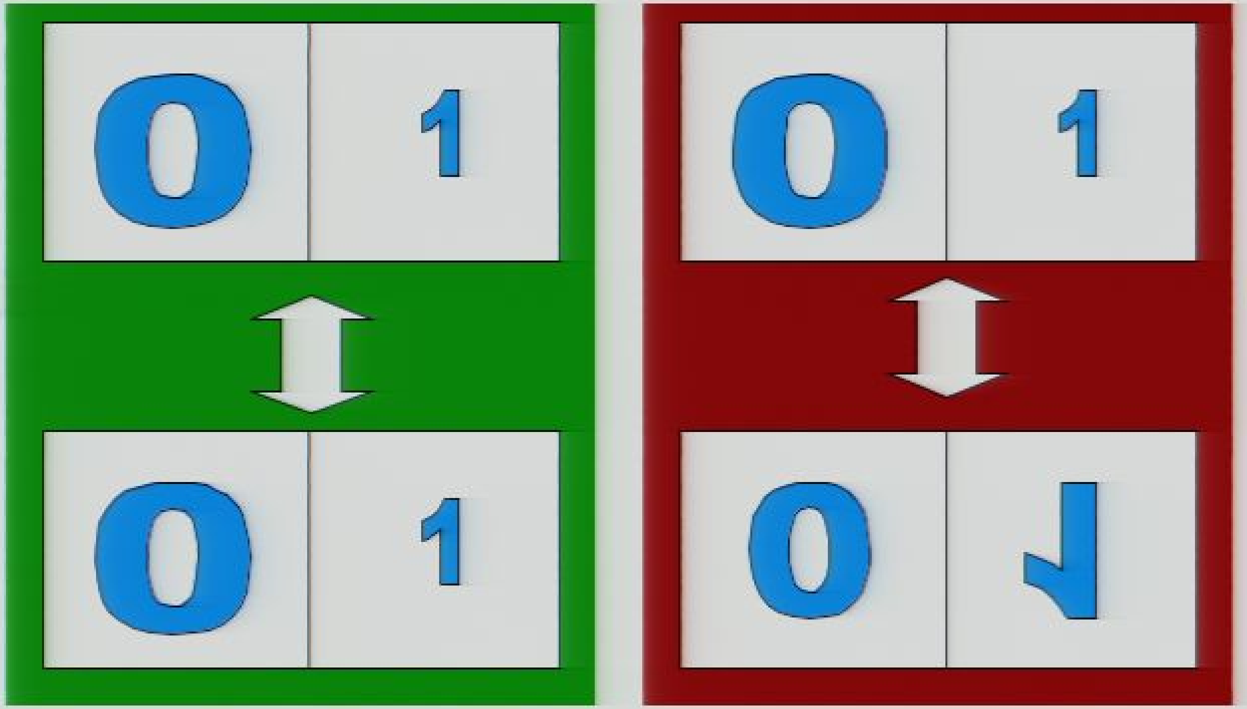


Only the person who created it can precisely identify it, but if told which one it is, anyone can check that.

The SWAP test

Despite not being able to precisely identify the state of a qubit, we can tell if two qubits are the same:

Switch them, and see if anything changes.



unchanged by SWAP

changed by SWAP

There is some randomness here - the example on the right is only partially changed.

Quantum Signature

Once Alice has given out qubits (her "public keys") that only she knows, she can use them to sign messages:

- Alice divides the qubits in half.
- To sign "0": reveal state of half of the qubits.
- To sign "1", reveal state of other half.
- For longer messages, she uses new sets of qubits.

Only Alice knows how to do this, so this proves the message came from her.

Quantum Signature

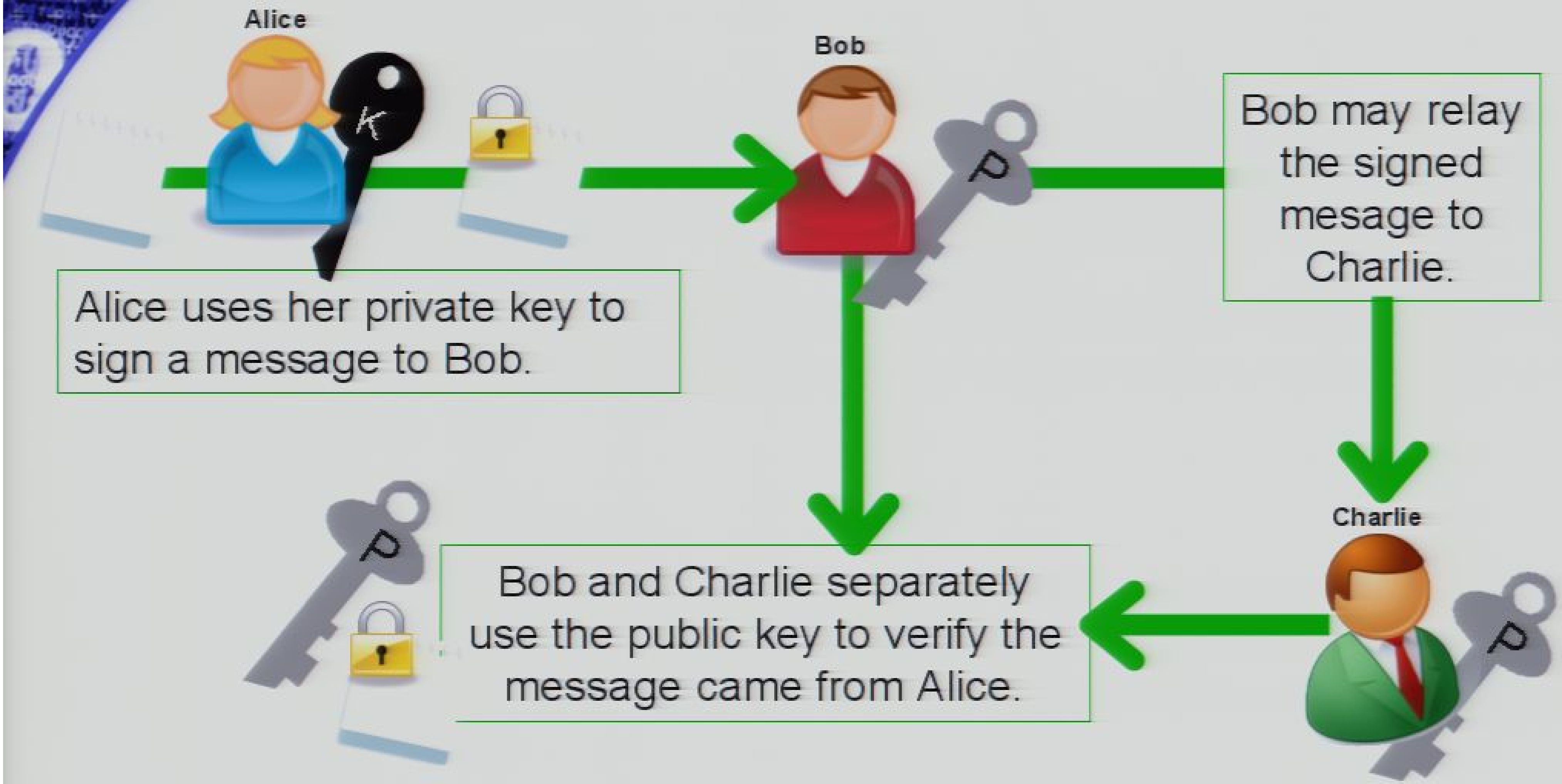
Once Alice has given out qubits (her “public keys”) that only she knows, she can use them to sign messages:

- Alice divides the qubits in half.
- To sign “0”: reveal state of half of the qubits.
- To sign “1”, reveal state of other half.
- For longer messages, she uses new sets of qubits.

Only Alice knows how to do this, so this proves the message came from her.

Alice: “Your mission, should you choose to accept it, is to [...]. If you are killed or captured, the agency will of course disavow all knowledge of you.”

Digital Signature

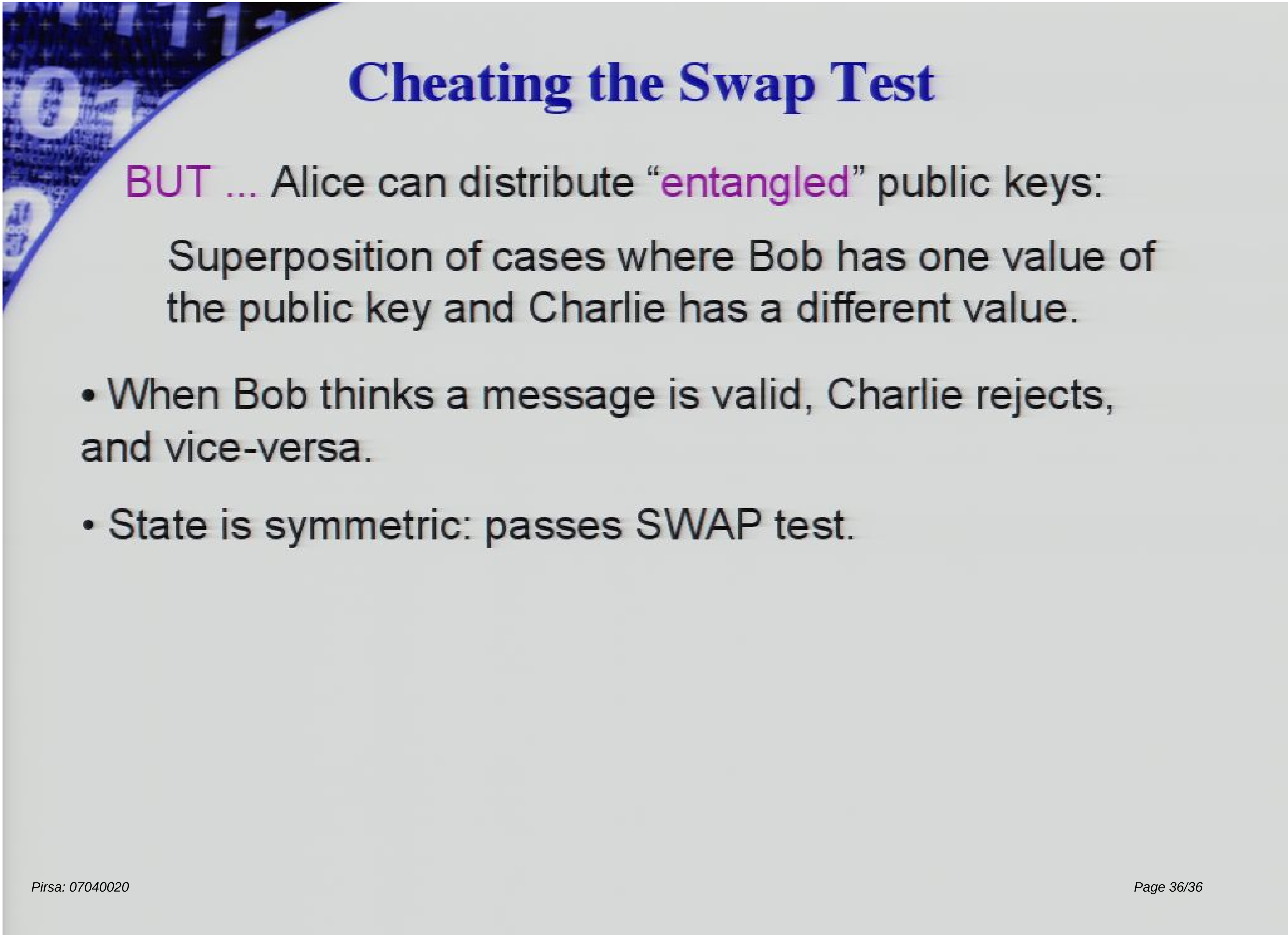


Alice uses her private key to sign a message to Bob.

Bob may relay the signed mesage to Charlie.

Bob and Charlie separately use the public key to verify the message came from Alice.

- Bob cannot alter the message he relays to Charlie.
- Bob knows that Charlie will agree Alice sent the message.



Cheating the Swap Test

BUT ... Alice can distribute “entangled” public keys:

Superposition of cases where Bob has one value of the public key and Charlie has a different value.

- When Bob thinks a message is valid, Charlie rejects, and vice-versa.
- State is symmetric: passes SWAP test.