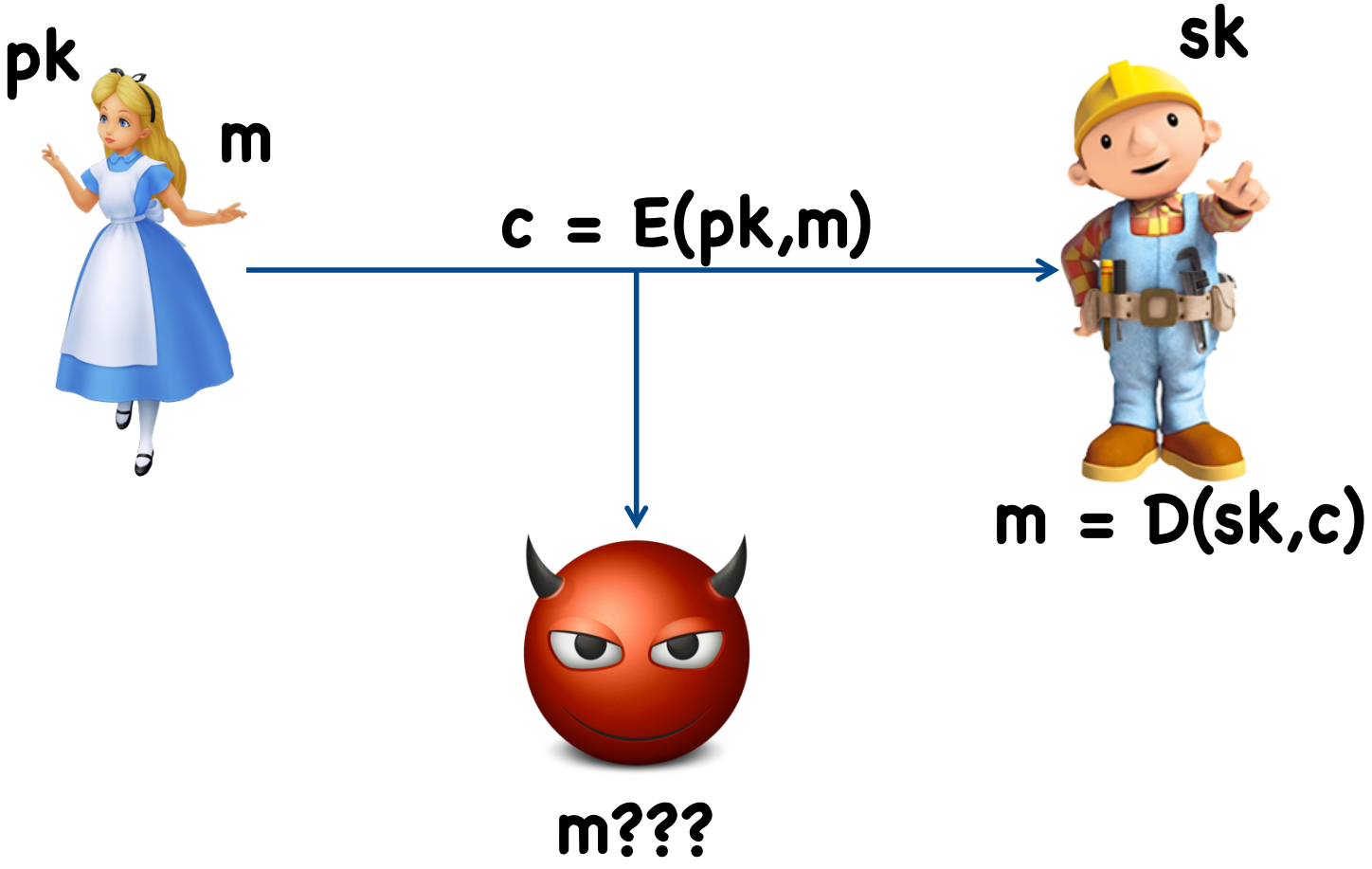
QUANTUM COMPUTERS AND CRYPTOGRAPHY

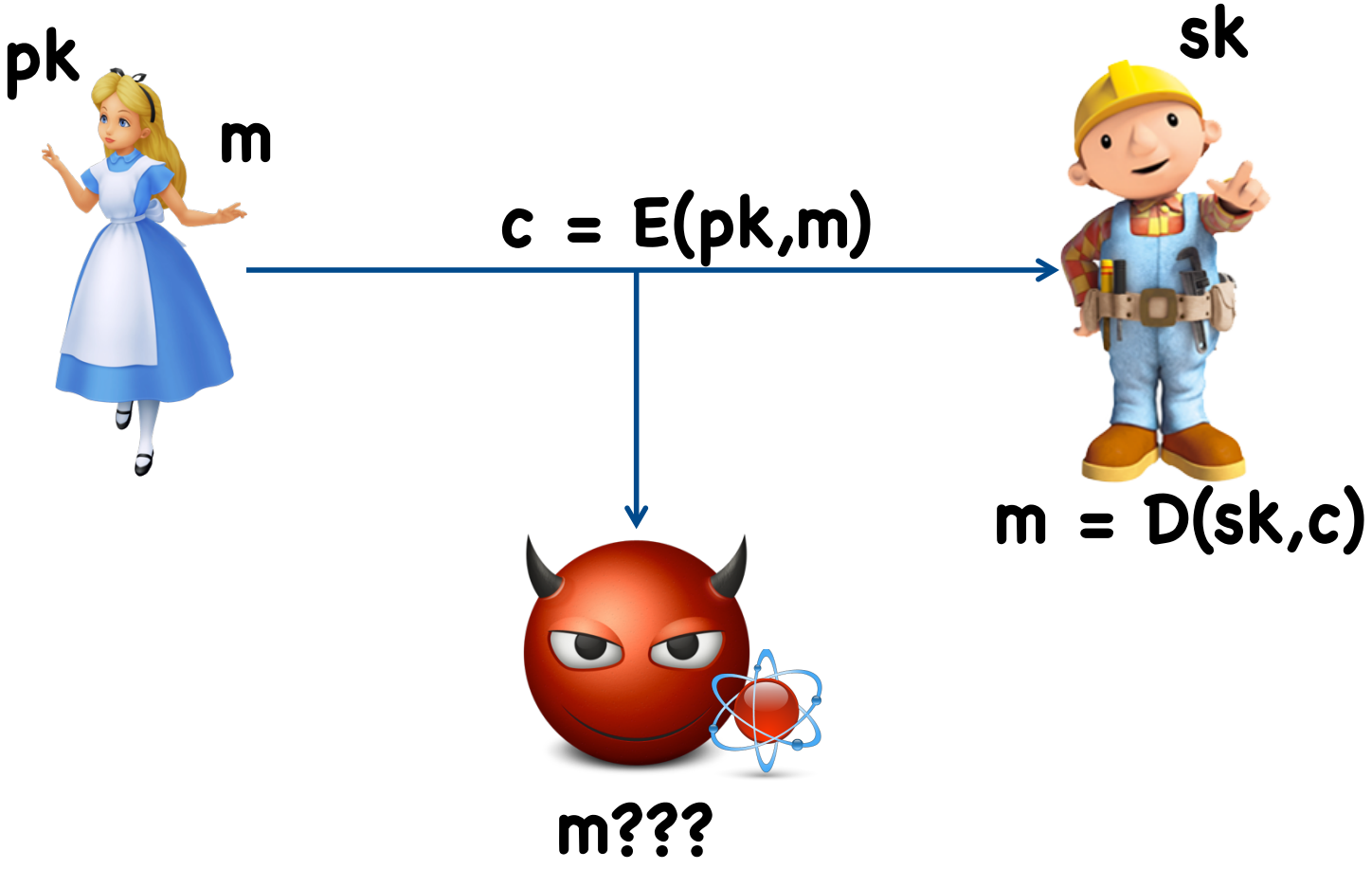
Mark Zhandry – Stanford University

Classical Encryption



Quantum Computing Attack

aka Post-quantum Crypto



Adversary uses quantum computer to speed up attack

This Talk

Shor's algorithm for factoring integers

Other quantum attacks on classical crypto

Quantum Crypto

Number Theory

x mod N : remainder of **x** when divided by **N**

x = y (mod N) : **(x-y)** is divisible by **N**

Suppose **p**,**q** prime, **N=pq**. Let **$\phi(N)=(p-1)(q-1)$**

Fermat's Little Theorem:

For all **x** relatively prime to **N**, **$x^{\phi(N)} = 1 \pmod N$**

Efficient (classical) modular operations:

- **x+y mod N**
- **x × y mod N**
- **x^{-1} mod N** (as long as **x** and **N** are relatively prime)
- **x^a mod N**

RSA Cryptosystem

KeyGen:

- Pick large (>500 bits) random primes **p**, **q**
- Let **N = pq**
- Let **e** be relatively prime to **ϕ(N)** (ex: **e=3**)

} public key

- Let $d = e^{-1} \bmod \phi(N)$

} private key

Encrypt **m < N**:

- Output $c = m^e \bmod N$

Decrypt **c**:

- Output $m = c^d \bmod N$

$$c^d = m^{ed} = m^{1+k\phi(N)} = m \pmod N$$

RSA Cryptosystem

If adversary can factor **N**, can decrypt

- Factoring also best known attack

Typical RSA key sizes:

Size of **N** (in bits)	Classical Security Level	Quantum Security Level
768	2^{70} (broken!)	2^{29}
1024	2^{80}	2^{30}
2048	2^{112}	2^{33}
3072	2^{128}	2^{35}
15360	2^{256}	2^{42}

Exponential growth

Polynomial growth

Shor’s Algorithm

Factoring on a Quantum Computer

Goal: Given **N=pq**, find **p**

Alternative Goal: Given **N**, find non-trivial root of unity **x**

> **x < N** is a *non-trivial root of unity* if:
> - $x^2 = 1 \pmod N$
> - $x \neq \pm 1 \pmod N$

Observation: $(x-1)(x+1) = 0 \pmod N$

- Therefore, **p** and **q** are factors of $(x-1)(x+1)$
- Both cannot be factors of $x-1$ (same for $x+1$)
- $GCD(N,x+1)$ gives **p** or **q**

Factoring on a Quantum Computer

Goal: Given **N**, find non-trivial root of unity

Fix $a < N$, a relatively prime to **N**. Let $f_a(z) = a^z \bmod N$

The *period* of a function **f** is an integer **r** where:

- $f(z+r) = f(z)$ for all integers **z**
- There is no $r' < r$ with $f(z+r')=f(z)$ for all **x**

Alternate Goal: Find period of f_a for random **a**

Factoring on a Quantum Computer

$f_a(z) = a^z \bmod N$

$f_a(z + \phi(N)) = a^{z+\phi(N)} = a^z \, a^{\phi(N)} = a^z \pmod N = f_a(z)$

Therefore, the period of f_a divides $\phi(N)$

Let r be the period of f_a. Suppose r is even.

Let $x = f_a(r/2) = a^{r/2} \bmod N$

Then $x^2 = 1 \pmod N$ and $x \neq 1 \pmod N$

Fact: with probability at least **3/8** over choice of a, r is even and $x \neq -1 \pmod N$

Factoring Strategy

- Pick a random integer $a < N$
- Check that a is relatively prime to N
  - If not, $GCD(a,N)$ is a factor of N
- Find period r of $f_a(z) = a^z \bmod N$
- If r is odd, abort
- Compute $x = f_a(r/2) = a^{r/2} \bmod N$
- If $x = -1 \bmod N$, abort
- Otherwise, x is a non-trivial root of 1

  → $GCD(x-1,N)$ is a factor of N

Only remaining step: period finding

Quantum Fourier Transform

Given modulus **n**, the *Quantum Fourier Transform* (QFT) maps the state

$$|x\rangle$$ (**log n** qubits corresponding to binary expansion of **x**)

(**0 ≤ x < n**) to the state

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{n}} \sum_{y=0}^{n-1} e^{2\pi i x y/n} |y\rangle$$

Quantum Fourier Transform

n=2:

$$\mathrm{QFT}|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

$$\mathrm{QFT}|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

→ Exactly the Hadamard **H** gate

$n=2^k$: Decompose into **k** QFTs for **$n'=2$**

- Basically implementing FFT as a quantum circuit

Similar idea for **smooth** numbers (all factors small)

Period Finding on a Quantum Computer

Let **f** be a periodic function with (unknown) period **r**

Suppose we know multiple **R** of **r**

Assume **f(0) ≠ f(1) ≠ f(2) … ≠ f(r-1)**

Step 0: Initialize two registers to the state:

$$|\psi_0\rangle = \frac{1}{\sqrt{R}} \sum_{z=0}^{R-1} |z, 0\rangle$$

Period Finding on a Quantum Computer

Step 1: Apply **f** to the registers:

$$|\psi_1\rangle = \frac{1}{\sqrt{R}} \sum_{z=0}^{R-1} |z, f(z)\rangle$$

Since **f(z+r) = f(z)**:

$$|\psi_1\rangle = \frac{1}{\sqrt{R}} \sum_{z=0}^{r-1} \sum_{w=0}^{(R/r)-1} |z + rw, f(z)\rangle$$

Period Finding on a Quantum Computer

Step 2: Measure second register

- Output of measurement is **f(z)** for a random **z**
- State collapses to be consistent with measurement

$$|\psi_2^z\rangle = \frac{1}{\sqrt{R/r}} \sum_{w=0}^{(R/r)-1} |z + rw, f(z)\rangle$$

Ignore second register:

$$|\psi_2^z\rangle = \frac{1}{\sqrt{R/r}} \sum_{w=0}^{(R/r)-1} |z + rw\rangle$$

Period Finding on a Quantum Computer

Step 3: Apply QFT with modulus **R** (ignore efficiency for now)

$$|\psi_2^z\rangle = \frac{1}{\sqrt{R/r}} \sum_{w=0}^{(R/r)-1} |z + rw\rangle$$

$$|\psi_3^z\rangle = \frac{\sqrt{r}}{R} \sum_{w=0}^{(R/r)-1} \sum_{y=0}^{R-1} e^{2\pi i y(z+rw)/R} |y\rangle$$

Period Finding on a Quantum Computer

$$|\psi_3^z\rangle = \frac{\sqrt{r}}{R} \sum_{w=0}^{(R/r)-1} \sum_{y=0}^{R-1} e^{2\pi i y(z+rw)/R} |y\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{y=0}^{R-1} e^{2\pi i z y/R} \left(\frac{r}{R} \sum_{w=0}^{(R/r)-1} e^{2\pi i r y w/R} \right) |y\rangle$$

$$\frac{r}{R} \sum_{w=0}^{(R/r)-1} \left(e^{2\pi i r y/R} \right)^w = \begin{cases} 1 & \text{if } y \text{ is a multiple of } R/r \\ 0 & \text{otherwise} \end{cases}$$

Period Finding on a Quantum Computer

Putting it together:

$$|\psi_3^z\rangle = \frac{1}{\sqrt{r}} \sum_{y=0}^{r-1} e^{2\pi i z y/r} |(R/r)y\rangle$$

Measure final register

→ Obtain **(R/r)y** for a uniform **y=0,…,r-1**

Repeat multiple times, take **GCD** of all results to obtain **R/r**

→ Obtain **r**

Caveats for Integer Factorization

We don't know any **R**

- **ϕ(N)** is a multiple of the period
- But knowing **ϕ(N)** is equivalent to knowing factors of **N**

QFT only efficient for "smooth" moduli, **ϕ(N)** is probably not smooth

Careful analysis shows:

If **R>>r**, can round **R** to a smooth number

(ex: powers of 2) and everything works out

Impact Of Quantum Computers

Much public-key crypto is broken:

- RSA-based
- Discrete Log-based
  - Diffie-Hellman key exchange
  - Elliptic Curve crypto
  - Shor's algorithm can be adapted for Dlog

Must use other crypto:

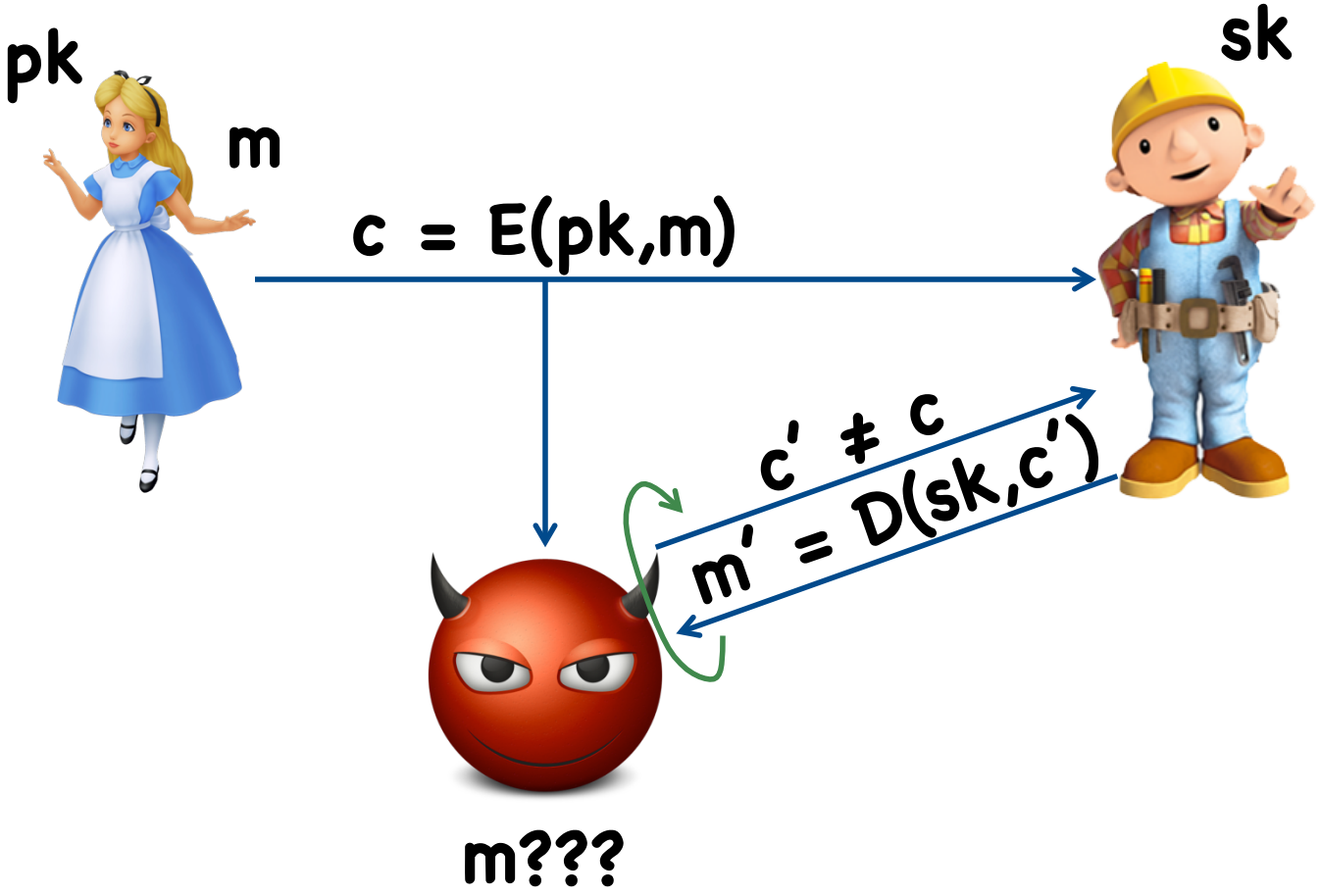
- Ex: Lattice crypto

Private-key crypto seems relatively safe

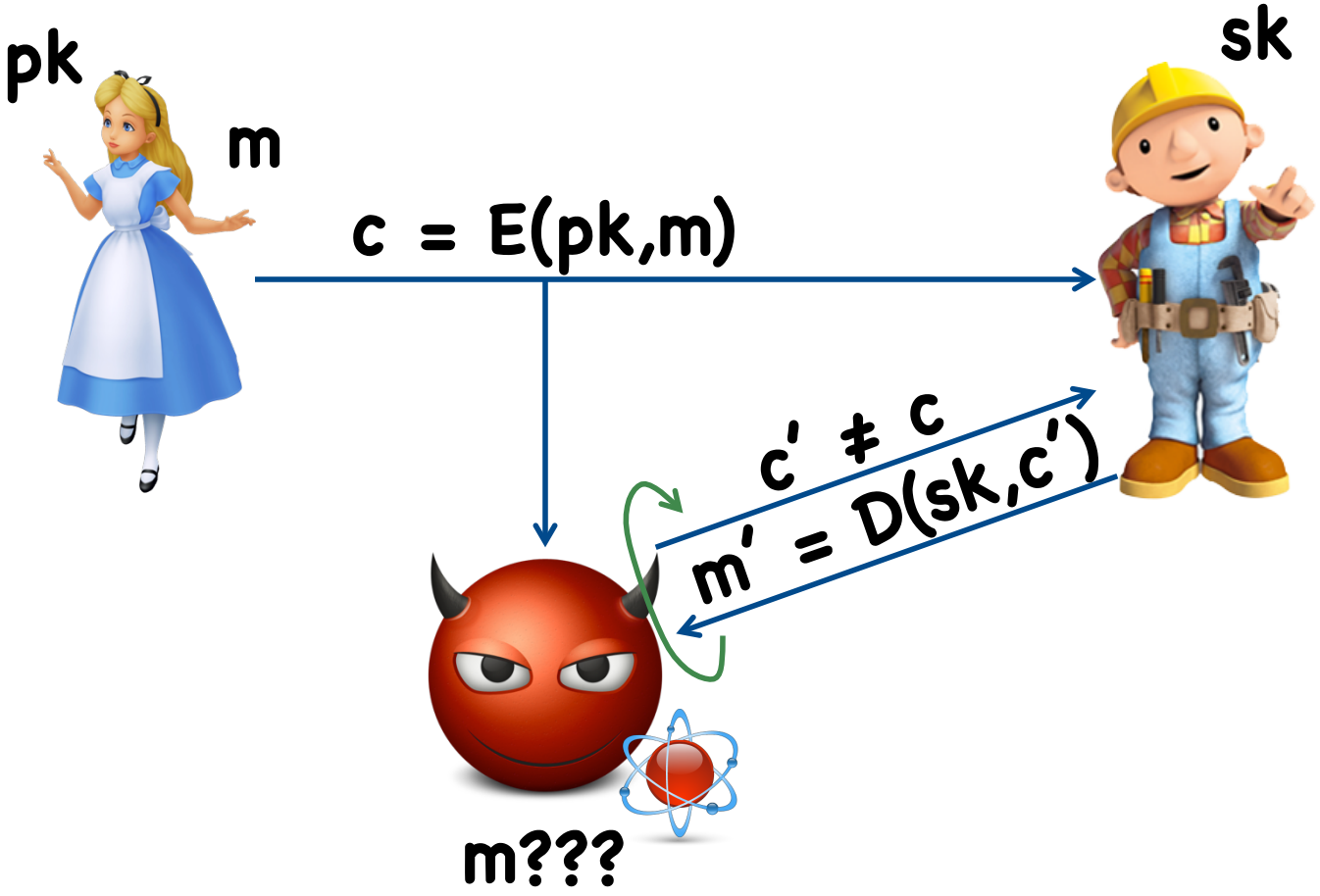
- Must increase key sizes to cope with Grover's algorithm

Quantum Channel Attacks

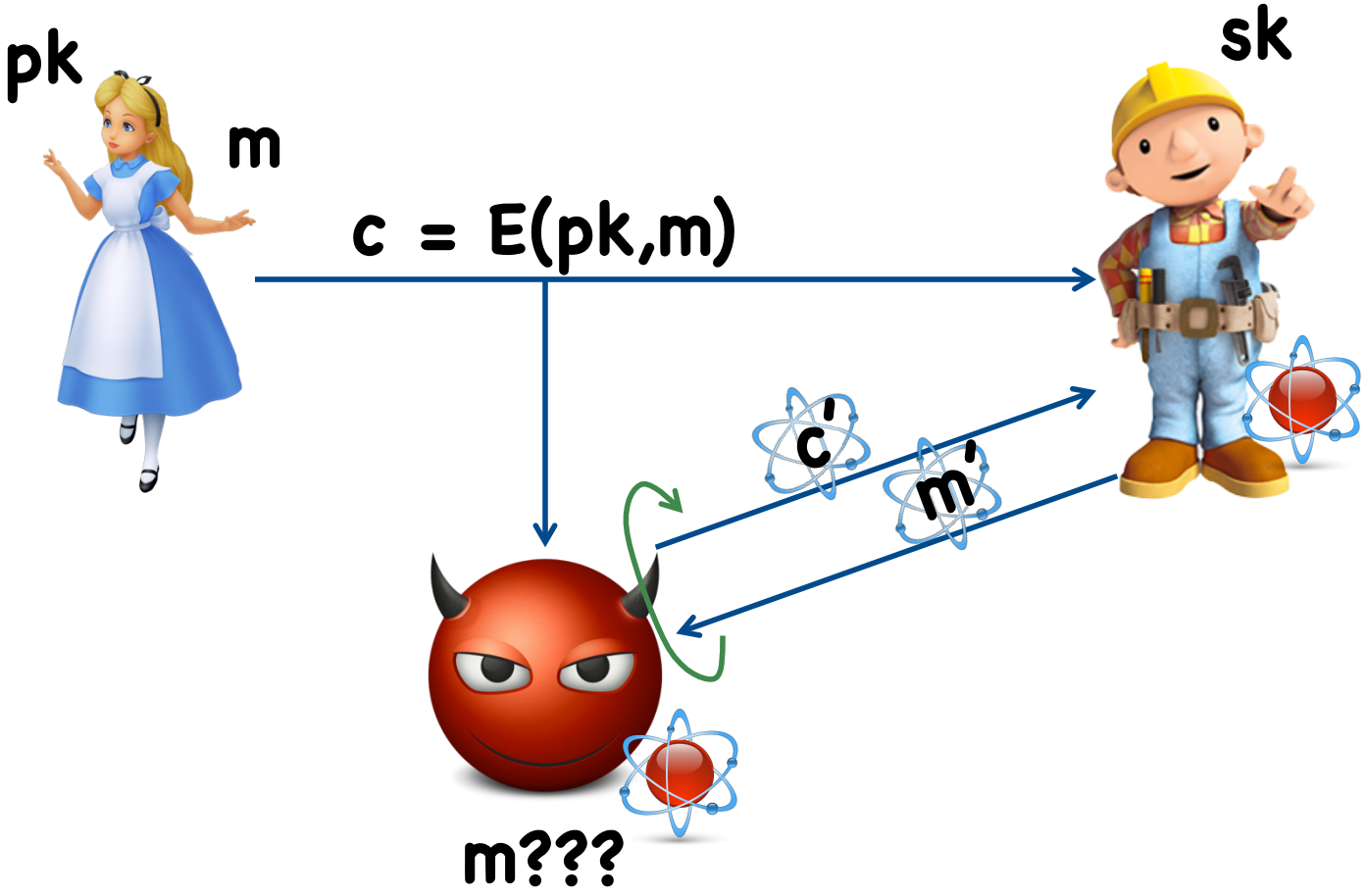
Classical Chosen Ciphertext Attack (CCA)



Post-Quantum Chosen Ciphertext Attack



Quantum Chosen Ciphertext Attack (qCCA)



Our Results on Quantum CCA Security

No known attacks on post-quantum schemes

Contrived post-quantum scheme susceptible to quantum CCA attack

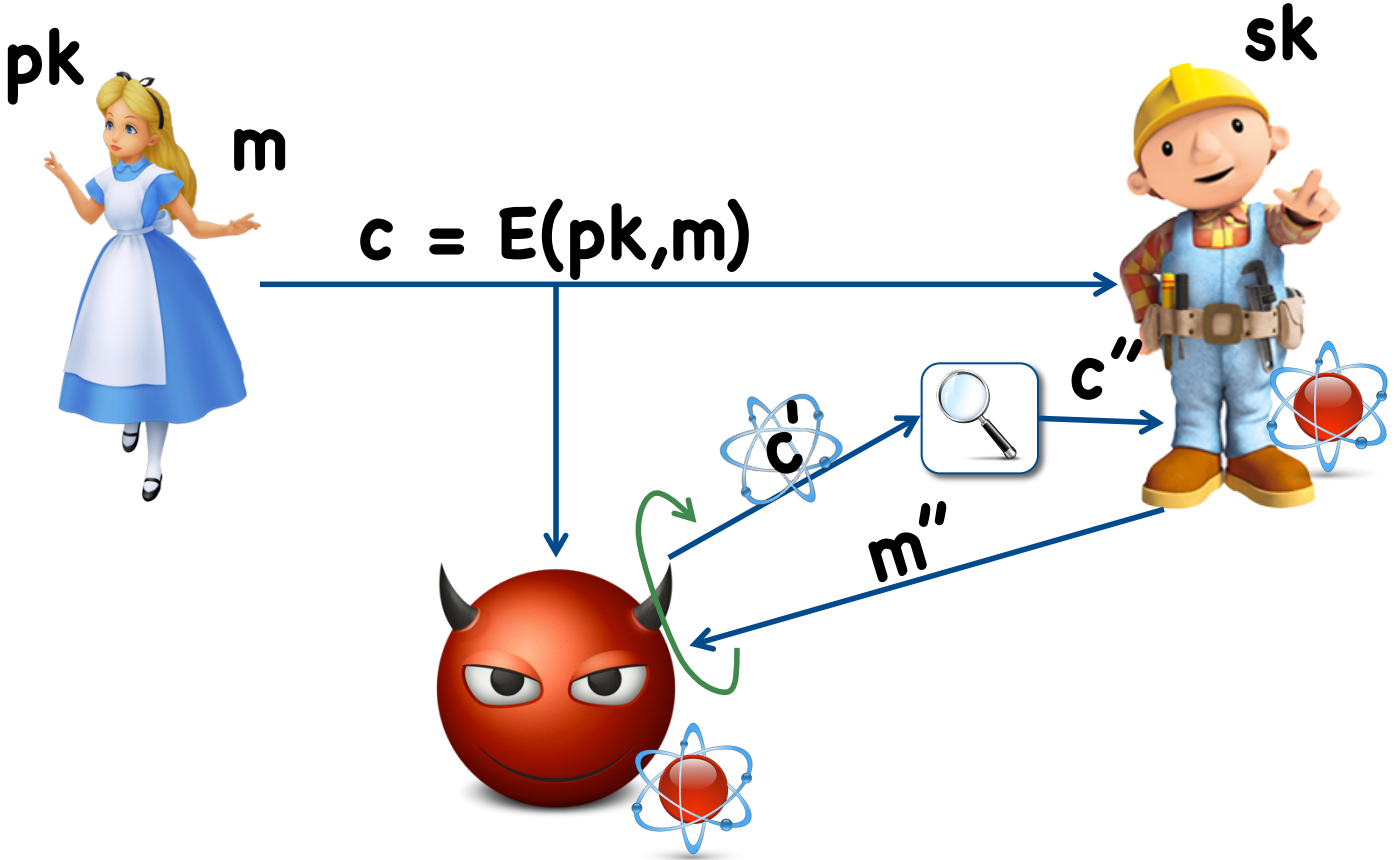
Example of scheme resistant to quantum CCA attack

- Basically, use scheme that is secure, even if adversary learns decryptions on **all** ciphertexts other than **c**

Discussion

Are these quantum channel attacks reasonable?

Objection: Bob can always measure before decrypting

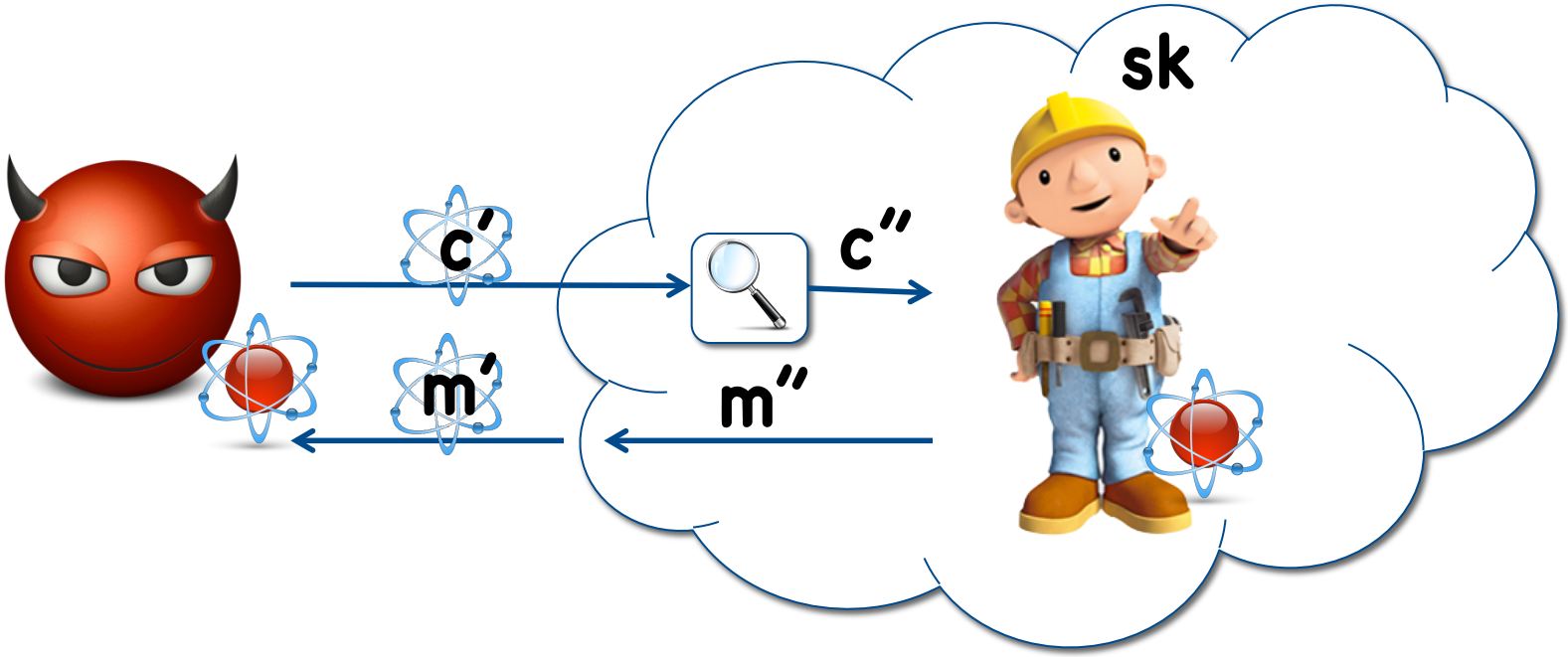


Discussion

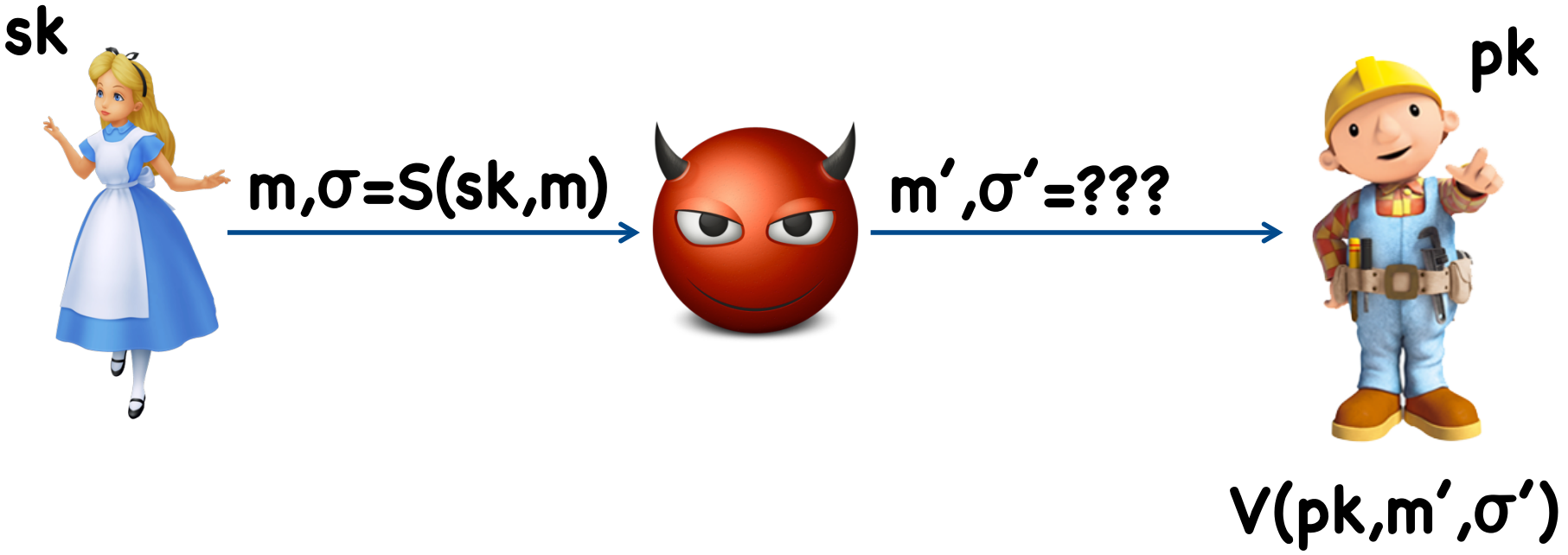
Are these quantum channel attacks reasonable?

Answer: measuring entangles with environment

- Adversary may control environment – attack restored!

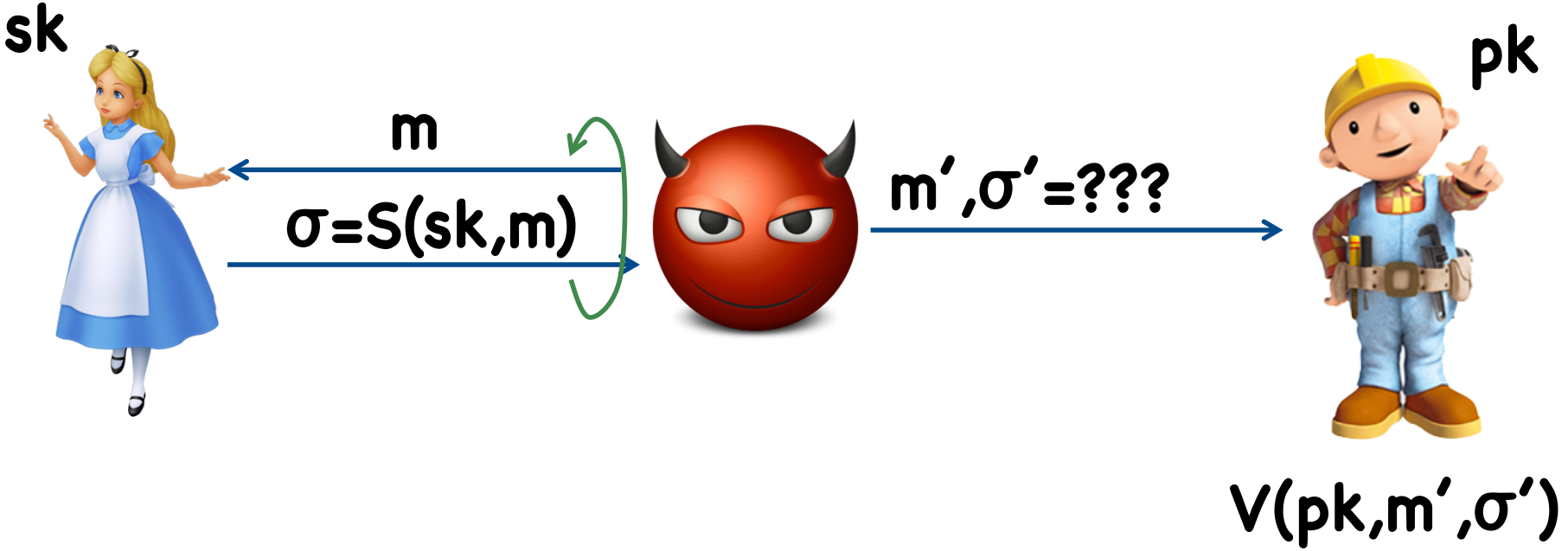


Digital Signatures



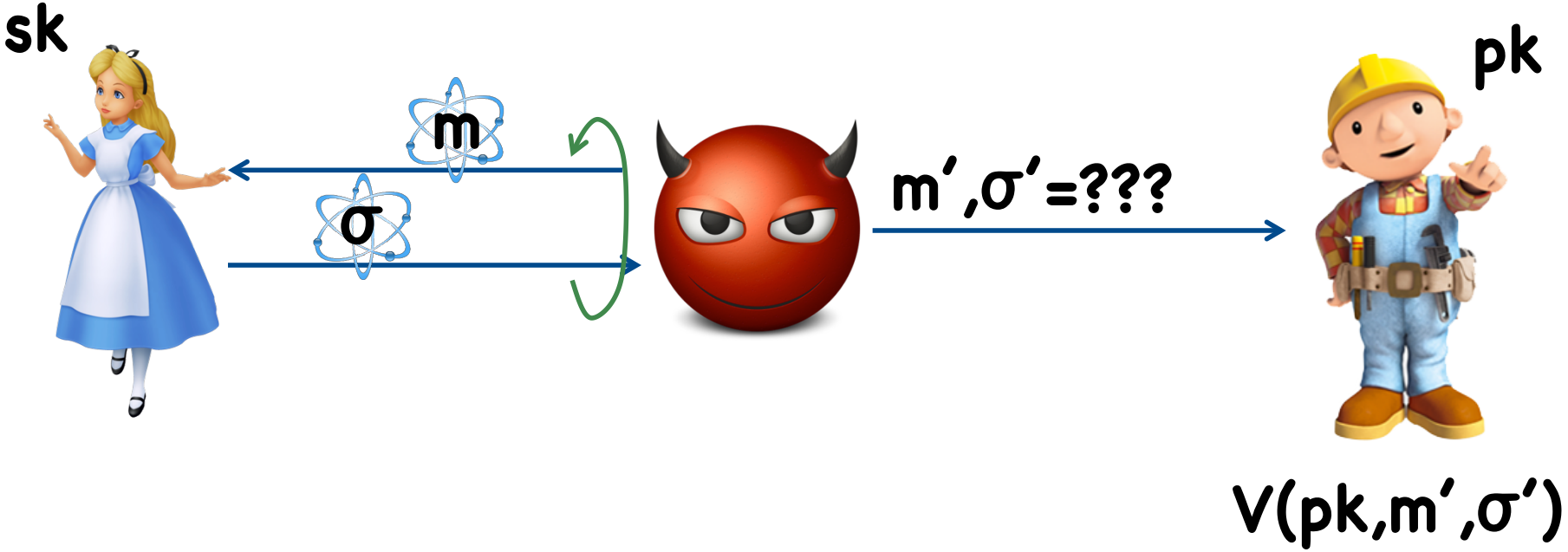
Adversary should not be able to sign **m′**

Classical Chosen Message Attack (CMA)



Adversary should not be able to sign any **m′** other than the **m**s he queried

Quantum Chosen Message Attack



Our Results on Quantum CMA Security

Again, no attacks on existing post-quantum schemes

Contrived post-quantum scheme susceptible to quantum CMA attack

Two simple conversions from post-quantum signatures

- Hash before signing
  - Likely to be used in practice anyway for efficiency
  - Heuristic security guarantees against qCMA attacks
- Use special hash function
  - Less efficient
  - Provable security guarantees

Other Results

Pseudorandom Functions (funcs that look random):

- Simple insecure example
- Common post-quantum constructions are secure

Message Authentication Codes (sigs with private verification):

- One existing post-quantum construction is insecure
- Other common constructions secure

Quantum Key Distribution (QKD)

Quantum Key Distribution

In classical crypto, key exchange requires computational assumptions

- At a minimum, P≠NP

With quantum mechanics, no longer true…

Quantum Key Distrubution

Two possible ways to encode bits in photon polarization:

- Rectilinear ($+$): $0 \longrightarrow |\uparrow\rangle$

  $1 \longrightarrow |\rightarrow\rangle$

- Diagonal ($\times$): $0 \longrightarrow |\nearrow\rangle$

  $1 \longrightarrow |\searrow\rangle$

Quantum Key Distribution

Measurement:

- Measure in $+$ basis:

$$|\uparrow\rangle \longrightarrow 0$$
$$|\rightarrow\rangle \longrightarrow 1$$
$$|\nearrow\rangle \longrightarrow 0 \text{ or } 1 \text{ with probability } 1/2$$
$$|\searrow\rangle \longrightarrow 0 \text{ or } 1 \text{ with probability } 1/2$$

- Measure in $\times$ basis:

$$|\uparrow\rangle \longrightarrow 0 \text{ or } 1 \text{ with probability } 1/2$$
$$|\rightarrow\rangle \longrightarrow 0 \text{ or } 1 \text{ with probability } 1/2$$
$$|\nearrow\rangle \longrightarrow 0$$
$$|\searrow\rangle \longrightarrow 1$$

Quantum Key Distribution

Needed properties:

- If measure in wrong basis, random output
- Cannot tell which basis used to build state
- Once measured, original state destroyed

Quantum Key Distribution

Phase 1:

Random bit: b
Random basis: c
Encode bit in basis: $|\psi\rangle$

$|\psi\rangle$

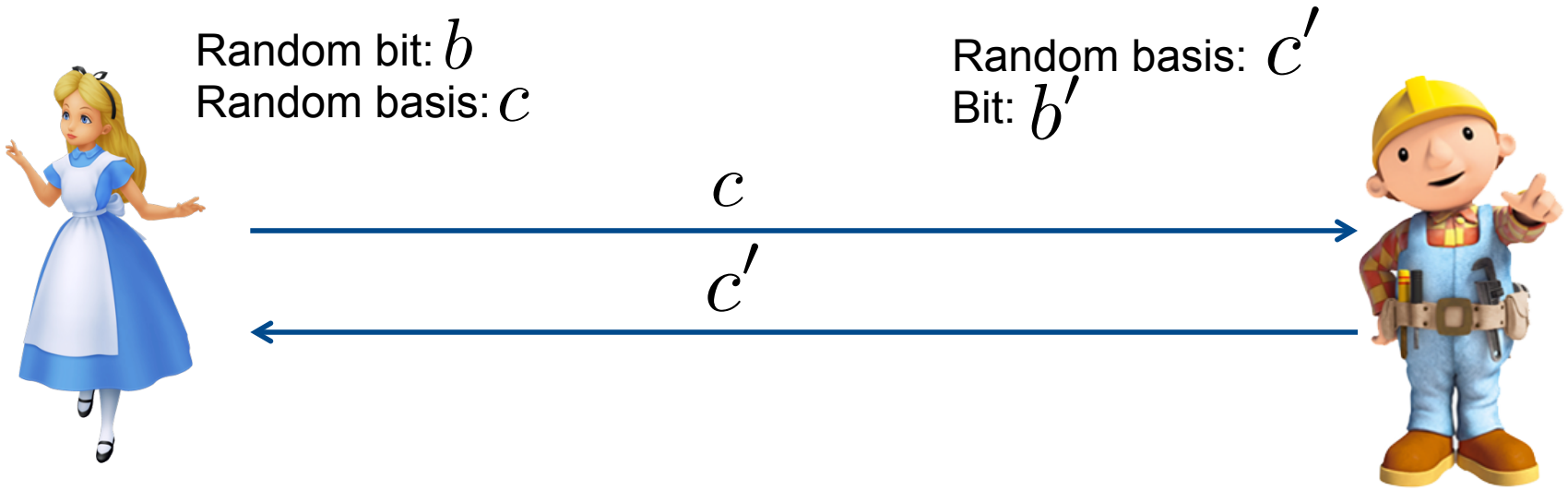
Random basis: c'
Measure in basis: b'

If $c = c'$, then $b = b'$
Otherwise, b, b' uncorrelated

Repeat many times

Quantum Key Distribution

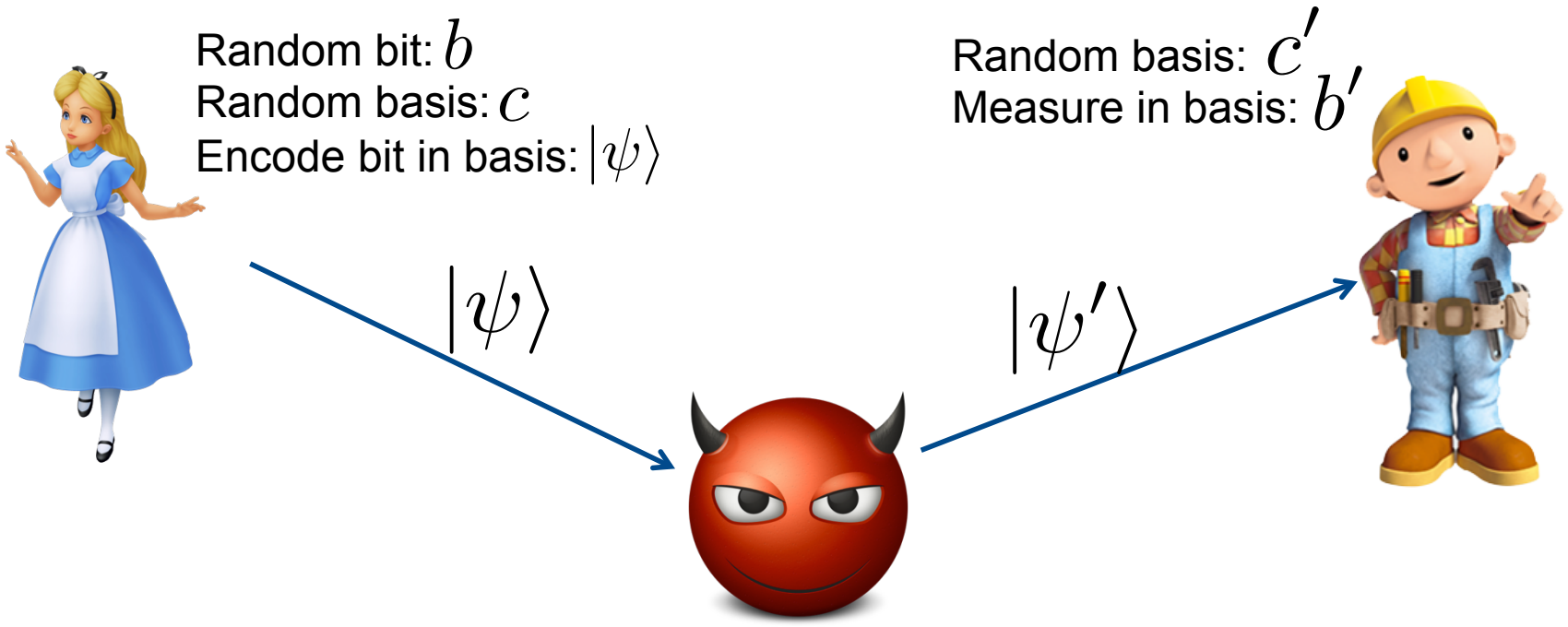
Phase 2:



If $c \neq c'$, throw away

Quantum Key Distribution

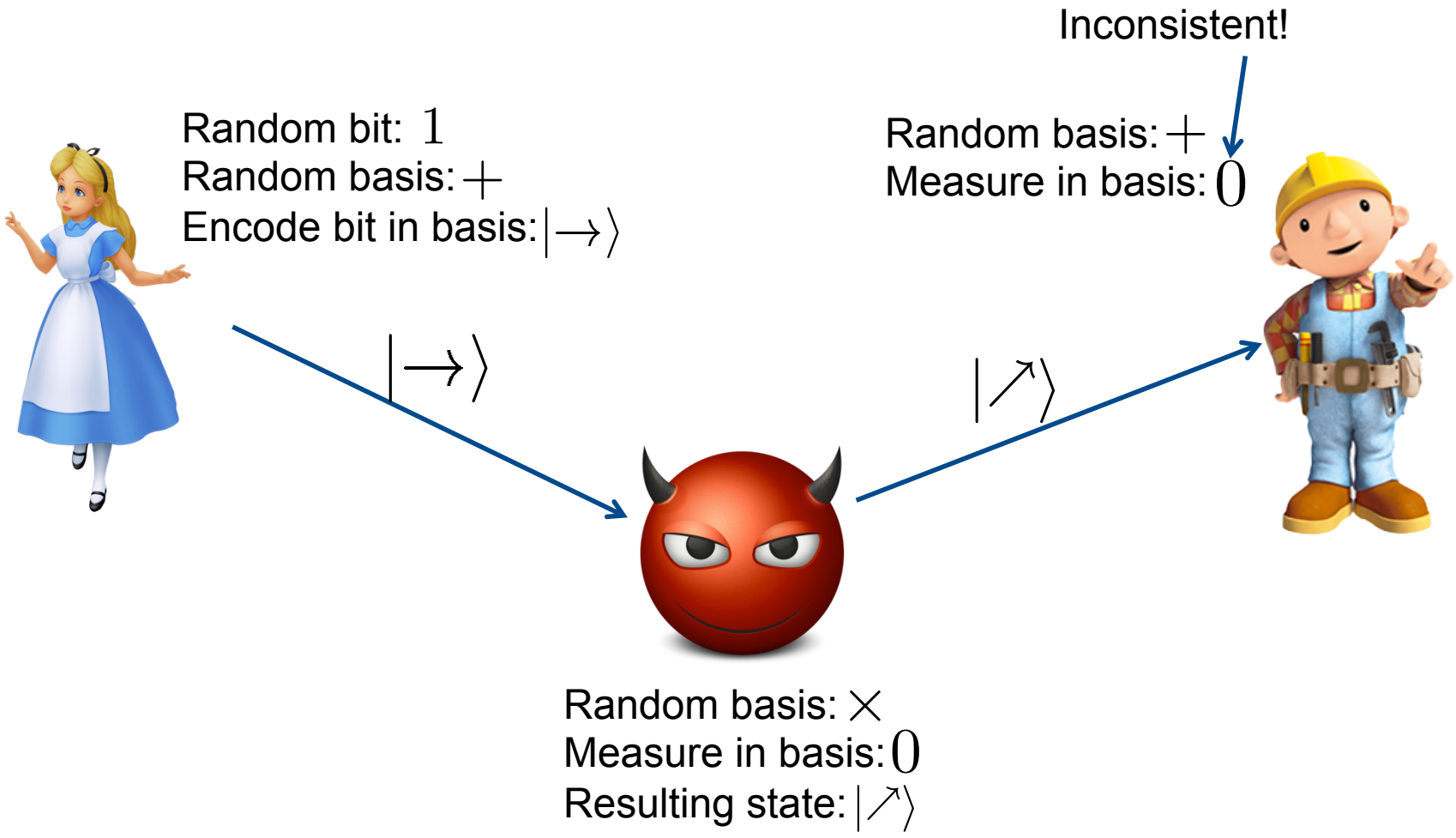
What happens if someone evesdrops?



Evesdropper cannot tell basis. What if guesses incorrectly?

→ Introduce errors, even if $c = c'$

Example


Inconsistent!
Random bit: 1
Random basis: $+$
Encode bit in basis: $|\rightarrow\rangle$
Random basis: $+$
Measure in basis: 0
$|\rightarrow\rangle$
$|\nearrow\rangle$
Random basis: $\times$
Measure in basis: 0
Resulting state: $|\nearrow\rangle$

Quantum Key Distribution

Phase 3:

Random fraction of bits

If too many errors, start over on new channel

More Details

Errors may occur even if no eavesdropping

- Impossible to tell if errors result from bad channel or an eavesdropper

Information reconciliation:

- Make sure Alice and Bob have same key
- Eavesdropper will have some partial information about key

Privacy Amplification:

- Reduce information to negligible
- Basically apply universal hash to key

Conclusion

Quantum computer will have a large impact on crypto

- Additional computational power → many systems broken
- Quantum channel attacks
- Quantum protocols → security without assumptions

A lot of work remains:

- Cryptanalysis of existing post-quantum schemes
- Reduce key sizes
- Increase distance for QKD