

# Alice (and Bob) in Wonderland

Henk Bierlee

Uppsala University

April 3, 2019

# Problem statement

## Abstract

Alice and Bob want to flip a coin by telephone. (They have just divorced, live in different cities, want to decide who gets the car.) Bob would not like to tell Alice HEADS and hear Alice (at the other end of the line) say "Here goes... I'm flipping the coin.... You lost!"

Figure: From *COIN FLIPPING BY TELEPHONE: A PROTOCOL FOR SOLVING IMPOSSIBLE PROBLEMS* by M. Blum (1981)

# Desired qualities

What are some desired qualities in this scenario?

- Fairness
  - Both parties have a 50/50 chance of winning
  - Consequently, neither party can cheat (or it's not in their best interest to attempt to do so)
- Non-repudiation: afterwards, either party can prove to a third party who won the game
- Cheat-detection: if cheating is attempted, the other party can find out

# An (im)practical solution

- 1 Alice writes down her call (heads/tails) on paper and locks it in a box
- 2 Alice sends the box (but not the key) to Bob
- 3 Bob flips a coin and reports the outcome to Alice
- 4 Alice reveals who won and sends her key to Bob so he can open the box and verify Alice's claim

The box holds Alice's *commitment* to her call. How do we achieve this with classical cryptography methods?

# Part I

## A classical solution [Blu81]

# One-way functions

A *normally secure* one-way function  $f$  is an efficiently computable function whose inverse cannot be computed efficiently: given  $f(x)$ , it is infeasible to find the input  $x$ .

Possible example:  $c = b^e \pmod{m}$  (modular exponentiation, solving for  $e$ )

A *completely secure* one-way function  $f$  has the additional property that given  $f(x)$ , one cannot determine some non-trivial property of  $x$  with  $> 50\%$  probability (for instance, the even or oddness of  $x$ ).

# An ideal solution

- 1 Alice and Bob agree on a completely secure one-way, one-to-one function  $f$
- 2 Alice (randomly) selects an integer  $x$  and computes  $f(x)$
- 3 Alice sends only  $f(x)$  to Bob
- 4 Bob (randomly) guesses whether  $x$  is even or odd
- 5 Alice reveals whether Bob was correct (and sends the original input  $x$  so Bob can verify that  $f(x) = x$ )

$f(x)$  constitutes Alice's *commitment* to her call of  $x$ . Since  $f$  is one-to-one, there is no other value that  $f$  maps to  $f(x)$ .

## Another approach

However, completely secure one-way functions are hard to find and may not exist at all! But maybe we can do it with a normally secure one-way function?

A *two-to-one* function  $f$  maps exactly two elements from its domain to each element of its range: for each input  $x$ , there always exists a second value  $y \neq x$  for which  $f(y) = f(x)$ .

Example:  $f(x) = |x|$ ?



## Another approach

However, completely secure one-way functions are hard to find and may not exist at all! But maybe we can do it with a normally secure one-way function?

A *two-to-one* function  $f$  maps exactly two elements from its domain to each element of its range: for each input  $x$ , there always exists a second value  $y \neq x$  for which  $f(y) = f(x)$ .

Example:  $f(x) = \frac{1}{|x|}$

# A classical solution

## Basic concept

Consider a normally secure one-way, two-to-one function  $f$  with two additional properties:

- Given  $x$ , it's hard to find  $y \neq x$ , such that  $f(y) = f(x)$
- The elements of an input pair  $x, y$  can be distinguished from each other
  - Example: when  $x$  is even,  $y$  is odd

# A classical solution

## Basic concept

Consider a normally secure one-way, two-to-one function  $f$  with two additional properties:

- Given  $x$ , it's hard to find  $y \neq x$ , such that  $f(y) = f(x)$
- The elements of an input pair  $x, y$  can be distinguished from each other
  - Example: when  $x$  is even,  $y$  is odd

Then, basically as before: Alice randomly selects  $x$ , sends  $f(x)$  to Bob, Bob guesses the property, and Alice sends  $x$  for verification

Alice's inability to find a second, distinct value  $y$  that  $f$  will also map to  $f(x)$  prevents her from cheating.

# A classical solution

- 1 We can efficiently compute an integer  $n = p_1 * p_2$ , where
  - $p_1, p_2$  are two random, large (80-digit) primes
  - $p_1 \equiv p_2 \equiv 3 \pmod{4}$
- 2 The Jacobi <sup>1</sup> symbol  $\left(\frac{x}{n}\right)$  (for odd positive integers  $n$  and arbitrary integers  $x$ ) has values 0, +1 or -1 and is efficiently computable
- 3 The group  $\mathbb{Z}_n^*$  holds all integers less than  $n$  which are relatively prime to  $n$  (meaning that numbers in  $\mathbb{Z}_n^*$  have no common divisors with  $n$  except 1)

---

<sup>1</sup>Generalisation of the Legendre symbol

# A classical solution

- ① We can efficiently compute an integer  $n = p_1 * p_2$ , where
  - $p_1, p_2$  are two random, large (80-digit) primes
  - $p_1 \equiv p_2 \equiv 3 \pmod{4}$
- ② The Jacobi <sup>1</sup> symbol  $\left(\frac{x}{n}\right)$  (for odd positive integers  $n$  and arbitrary integers  $x$ ) has values 0, +1 or -1 and is efficiently computable
- ③ The group  $\mathbb{Z}_n^*$  holds all integers less than  $n$  which are relatively prime to  $n$  (meaning that numbers in  $\mathbb{Z}_n^*$  have no common divisors with  $n$  except 1)

Roughly speaking,  $n$  (from point 2) can provide a one-way function  $f_n$  with inputs from  $\mathbb{Z}_n^*$  of which exactly half the solutions of the inverse will have  $\left(\frac{x}{n}\right) = +1$  (and the other half  $\left(\frac{x}{n}\right) = -1$ ).

---

<sup>1</sup>Generalisation of the Legendre symbol

# A classical solution

## Pros and cons

- + Non-repudiation: if all messages are signed, then the outcome of the protocol can be proven to an outside source (ie. a judge)
- + Cheat-detection: one party (or a trusted third party) constructs  $n$ , which can be tested for the desired properties
- The security of the algorithm depends on the prime factors of  $n$  being secret
  - In other words, the assumption that prime factorisation is *computationally hard*
  - But advances in algorithm design makes this an unsure assumption
  - For example, Shor's algorithm can factorise primes efficiently on quantum hardware [BL17]

## Part II

### A quantum solution [BB14]

# Behaviour of polarised photons

## Encoding bits as photons

- Light exists of photon particles which can be polarised at any angle with a polarising filter (up, right, diagonal, ..)
- This means we can encode a bit of information (0/1) into a photon in multiple ways, or *bases* (rectilinear basis, diagonal basis, ..)
- A photon's polarisation can be measured with a filter, but..
  - Rectilinear photon + rectilinear filter = deterministic outcome
  - Rectilinear photon + diagonal filter = random outcome
- Since a photon cannot be duplicated ('no-cloning theorem') and it loses its original polarisation after measurement, you only get one chance to measure it.



# Behaviour of polarised photons

## Example

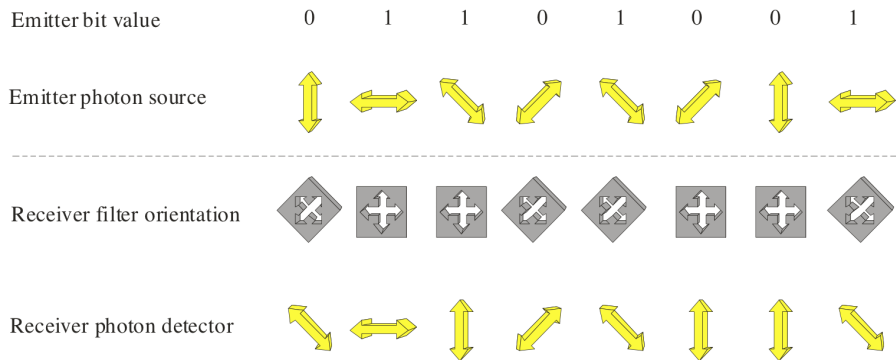


Figure: From the Cryptology lecture slides (Feb. 27, 2019)

# A quantum solution

Alice's bit string .....	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0
Alice's random basis .....	Rectilinear														
Photons Alice sends .....	$\updownarrow$	$\leftrightarrow$	$\updownarrow$	$\leftrightarrow$	$\leftrightarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\leftrightarrow$	$\updownarrow$	$\leftrightarrow$	$\updownarrow$	$\updownarrow$	$\leftrightarrow$	$\leftrightarrow$
Bob's random bases .....	R	D	D	D	R	R	D	R	R	D	R	R	D	D	R
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		
Bob's guess.....	'Rectilinear' 'You win'														
Alice's reply .....															
Alice sends her original bit string to certify	'1	0	1	0	0	1	1	1	0	1	0	1	1	0	0'
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		

**Figure:** Table from *Quantum cryptography: Public key distribution and coin tossing*

1. Alice encodes a random bit-string with a randomly chosen basis and sends them to Bob

# A quantum solution

Alice's bit string .....	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0
Alice's random basis .....	Rectilinear														
Photons Alice sends .....	↓	↔	↓	↔	↔	↓	↓	↓	↔	↓	↔	↓	↓	↔	↔
Bob's random bases .....	R	D	D	D	R	R	D	R	R	D	R	R	D	D	R
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		
Bob's guess.....	'Rectilinear' 'You win'														
Alice's reply.....															
Alice sends her original bit string to certify	'1	0	1	0	0	1	1	1	0	1	0	1	1	0	0'
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		

**Figure:** Table from *Quantum cryptography: Public key distribution and coin tossing*

2. Bob randomly chooses a basis for each photon and fills corresponding measurement table and makes a guess

# A quantum solution

Alice's bit string .....	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0
Alice's random basis .....	Rectilinear														
Photons Alice sends .....	$\updownarrow$	$\leftrightarrow$	$\updownarrow$	$\leftrightarrow$	$\leftrightarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\leftrightarrow$	$\updownarrow$	$\leftrightarrow$	$\updownarrow$	$\updownarrow$	$\leftrightarrow$	$\leftrightarrow$
Bob's random bases .....	R	D	D	D	R	R	D	R	R	D	R	R	D	D	R
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		
Bob's guess.....	'Rectilinear' 'You win'														
Alice's reply .....															
Alice sends her original bit string to certify	'1	0	1	0	0	1	1	1	0	1	0	1	1	0	0'
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		

**Figure:** Table from *Quantum cryptography: Public key distribution and coin tossing*

3. Alice reveals the correct basis and sends original bit-string over classic communication channel

# A quantum solution

Alice's bit string .....	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0
Alice's random basis .....	Rectilinear														
Photons Alice sends .....	↓	↔	↓	↔	↔	↓	↓	↓	↔	↓	↔	↓	↓	↔	↔
Bob's random bases .....	R	D	D	D	R	R	D	R	R	D	R	R	D	D	R
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		
Bob's guess.....	'Rectilinear' 'You win'														
Alice's reply.....															
Alice sends her original bit string to certify	'1	0	1	0	0	1	1	1	0	1	0	1	1	0	0'
Bob's rectilinear table .....	1					1					0				0
Bob's diagonal table .....		0		1						1			0		

**Figure:** Table from *Quantum cryptography: Public key distribution and coin tossing*

4. Bob verifies that Alice didn't cheat by comparing bit-string to corresponding measurement table

More volunteers..?

# A quantum solution

## Practical application

- We need *good* quantum hardware..
  - Otherwise Alice might accidentally send multiple photons, allowing Bob to measure multiple times and guess the filter with  $> 50\%$  probability
- ..but we don't want *perfect* quantum hardware.
  - Otherwise Alice could cheat by 'entangling' photons with the so-called Einstein-Podolsky-Rosen effect
  - In practice likely impossible to achieve

First experimental quantum coin-toss already performed at the Laboratory for Communication and Processing of Information (LTCI) in Paris!

# A quantum solution

## Pros and cons

- + By the properties of quantum mechanics, it is impossible to cheat:
  - Alice cannot cheat, because she can predict only one table with 100% accuracy, but not the other
  - Bob cannot cheat, because the photons give no information as to the basis he has to guess
- No non-repudiation: quantum cryptography does not provide digital signatures

So we rely on the *laws of physics* instead of on *computational complexity*





*Questions..?*

*Thank you for listening!*

Report, slides and demo source code available at:

<https://github.com/hbierlee/quantum-coin-toss>

# References

-  Charles H Bennett and Gilles Brassard, *Quantum cryptography: public key distribution and coin tossing.*, Theor. Comput. Sci. **560** (2014), no. 12, 7–11.
-  Daniel J Bernstein and Tanja Lange, *Post-quantum cryptography-dealing with the fallout of physics success.*, IACR Cryptology ePrint Archive **2017** (2017), 314.
-  Manuel Blum, *Coin flipping per telephone: A protocol for solving problems impossible*, SIGACT News **15** (1981), 23–27.