COIN FLIPPING BY TELEPHONE

„A protocol for solving impossible problems“

Based on Manuel Blum's Paper from 1981

Proved useful for:

- Mental poker
- Certified Mail
- Exchange of Secrets

Table of Contents

- Application scenario and applicability
- Goals
- Needed mathematical properties
- Base Assumptions
- Mathematical approach and implementation
- Actual discussion of protocol steps

Universität Augsburg University

Application scenario

- Alice and Bob have recently been divorced
- Who gets the car?
- How to decide over a distance?
- B doesn't want to guess, hear A flip coins, then lose => cheatproofing!

[1]

Applicability

- Two adversaries
- One of them generates RANDOM Bits
- It is in the interest of the picking side NOT to pick at random

[2]

Thus, a protocol is needed to ensure random picking

Universität Augsburg University

Goals

- Guarantee to B that A WILL pick a random sequence of Bits

  => Flip coins „to“ her
- Guarantee to A that B does not know the sequence himself
- Ensure and enforce that no cheating occurs

  => Judge's protocol

Universität Augsburg University

One-Way functions

- Two types: Completely and normally secure
- x => f(x) is easily computible, f(x) => x is not for the vast majority of values
- From knowing f(x), you have a maximal chance of 50% to guess a non-trivial property of x
- Both functions share the first property, the second is exlusive to the completely secure variation

Universität Augsburg University

Protocol using a CS-OWF

- A and B agree on a function f
- A picks an integer x, computes f(x) and sends it to B
- B guesses whether x is even or odd
- A reveals the result and sends x as confirmation
- End
- But: A CS-OWF is hard to come by, maybe it doesn't even exist [B'81]

2 – 1 function

- For the protocol, we'll use a normally secure one-way function with an additional property
- The 2 – 1 function maps two elements from its domain to each element of each range
- For x,y: f(x) = f(y)
- The values x and y shall be distinguished by a simple property => even and odd

Universität Augsburg University

Consequences

- If A computes f(x) and sends it to B, he has no idea if x or y was used for the computation
- B then guesses whether the original number was even or odd
- A then reveals the result and sends the number she used, in this case x
- The sending must be automatized, or else A may just send y without anyone being the wiser

Universität Augsburg University

Needed properties

- No cheating => probability is 50-50, guaranteed
- If one participant is caught cheating, it is provable to the judge
- After B's coin tosses, A knows the results. Until she reveals it to him, he has no idea.
- After the flips, A can verify to B how the coins fell

=> The protocol fulfills all goals while staying applicable

Additional properties

- Each adversary knows at each step if the other cheats => The court only provides justice, independent proof or force a participant to complete the protocol
- B can use his public key n, provided its correct construction, to flip coins
- B does not need new primes for each flip, for the needed computation time per flip is of the order of gcd(x,y)

Universität
Augsburg
University

Assumptions

- We assume that no procedure can efficiently factor a number n comprised of two large primes (1)
- We assume both A and B possess their own true random number generators (2)
- Signed messages via the secure signature proposed by Diffie and Hellman [DH'79], implemented by Rivest, Shamir and Adleman [RSA'78] (3)

Assumption (1)

- In 1980s Technology, a 1000 CRAY-1's would need over 5 years to factorize a 160 digit prime [B'81]
- The interesting property for computational speed is FLOPS (Floating point operations per second)
- 1000 CRAY-1A's: 80 GFLOPS [3]
- 1 Nvidia Geforce GTX 1080 Ti: 11.5 TFLOPS (SP) [4]
- A difference in speed of a factor 143,75

Universität Augsburg University

Assumption (1)

- If we assume the computation time of the CRAY-1's to be exactly 5 years, then the graphics card will take only about 12 and a half days to factorize the prime

  => In our time, the protocol has to be either completed in a much shorter timeframe, or the prime needs to be much larger
- If we use a modern supercomputer, the needed time can easily be pushed down into the hour range

Universität Augsburg University

Assumption (1)

- What about using a quantum computer instead of a regular one?
- Classical factorization algorithms scale with an exponential order
- Quantum factorization algorithms scale with a below exponential order (almost polynomial)

  => A quantum computer would be able to factorize the number in an even shorter timeframe

Universität Augsburg University

Assumption (2)

- A true random number generator may be impossible to construct [B'81]
- There are several decent generators [5]
- We assume that any modern random number generator will be sufficient for the application

Universität Augsburg University

Assumption (3)

- The basic idea of the encryption [DH'79]:
  - In advance, A and B agree on a shared secret
  - If need be, A and B mix their secret with the shared one and send it
  - When they mix the other's secret with their own, they both obtain the same, common secret

=> The method is set up this way! If there is a difference in outcome, the shared secret is not the same either

Universität Augsburg University

Jacobi-Symbol

$$(x/n) = \begin{cases} 1, & \text{if } x \text{ is quadratic residue to } a, \\ 0, & \text{if } x \text{ is factor of } n, \\ -1, & \text{if } x \text{ is no quadratic residue to } a. \end{cases}$$

- Quadratic residue:

$$x := a^2$$
$$x = a^2 + t * n$$

Definition by code

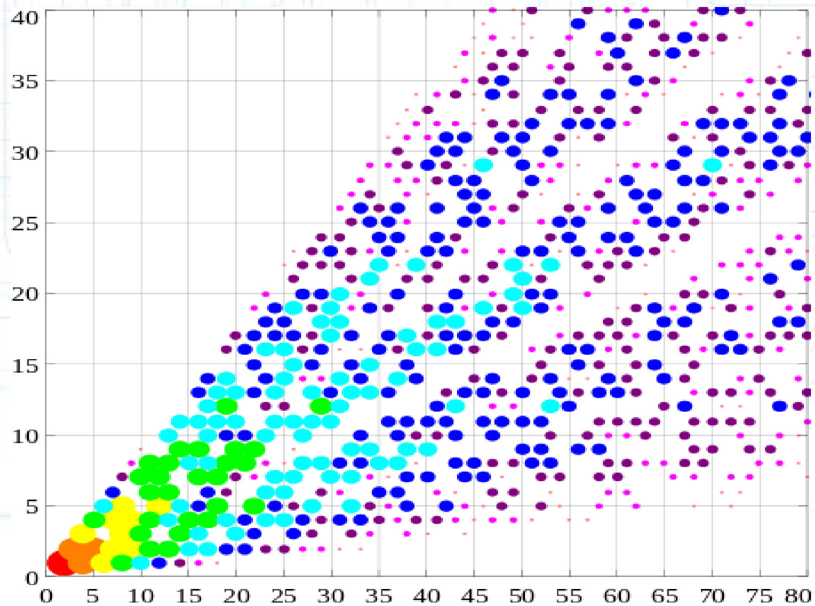
- x_i: arbitrary integers
- n_i: positive odd integers
- $(x/n) = 0$ if $\gcd(x,n) =/= 1$
- $(1/n) = 1$
- $((x_1*x_2)/n) = (x_1/n)*(x_2/n)$
- $(x/(n_1*n_2)) = (x/n_1)*(x/n_2)$
- $(x_1/n) = (x_2/n)$ if $x_1 = x_2 \mod n$

Definition by code

- $(-1/n) = +1$ if $n = 1 \bmod 4$, or
- $(-1/n) = -1$ if $n = 3 \bmod 4$
- $(2/n) = +1$ if $n = 1$ or $7 \bmod 8$, or
- $(2/n) = -1$ if $n = 3$ or $5 \bmod 8$
- $(n_1/n_2) = (n_2/n_1)$ if $\gcd(n_1,n_2) = 1$ and [n_1 or $n_2 = 1 \bmod 4$], or
- $(n_1/n_2) = -(n_2/n_1)$ if $\gcd(n_1,n_2) = 1$ and [n_1 and $n_2 = 3 \bmod 4$]

Zn*

- For an integer n > 1:
- Zn* = {0 < n_i < n; n_i coprime to n}
- Coprime := No shared common positive factors except 1



[6]

Lemma

$$n = p_1^{e_1} * \ \ldots \ * p_k^{e_k}$$

$$k := \text{ integer } > 1$$

$$p_i := \text{ distinct odd primes}$$

$$e_i := \text{ positive integers}$$

$$a \in Zn^* \text{ is quadratic residue mod n}$$

$$x^2 = a \bmod n$$

Lemma

$$x = [\pm(x_1 v_1 p_2^{e_2} * \ \dots \ * p_k^{e_k}) \pm (x_2 v_2 p_1^{e_1} * \ \dots \ * p_k^{e_k}) \pm \\ \pm \dots \pm (x_k v_k p_1^{e_1} * \ \dots \ * p_{k-1}^{e_{k-1}})] \mod n$$

v_1 is any integer defined so $(v_1 p_2^{e_2} * \ \dots \ * p_k^{e_k}) \mod p_1^{e_1} = 1$

- The other values of v are similarily defined
- Proof by LeVeque, Theorems 3.21 & 5.2
  [L'77]

Theorem 1

- For any odd integer n which can be expressed in the way of the lemma, except that $k = 1$ is also permitted, then the following statements are equivalent:
- There are x,y in Z_n^* with $x^2 = y^2 \bmod n$ and $(x/n) \neq (y/n)$
- $p_i^{(e_i)} = 3 \bmod 4$ for some i
- Let a from Z_n^* be a quadratic residue to mod n; Then exactly half the roots in Z_n^* of the equation $a = x^2 \bmod n$ have $(x/n) = 1$, the other half $(x/n) = -1$

Publication date of n

- At the beginning of the protocol, B sends n and the publication date of n to A – why?
- B cannot be sure that the factorization of n remains secret for longer than a fixed period of time
- It is unreasonable to expect indefinite contact between A and B

Universität
Augsburg
University

The protocol – Step 1

- Bob selects n:
  - He picks two random exactly 80-digit primes p_1 and p_2, both congruent to 3 mod 4
  - Then n = p_1 * p_2

B => A: „n, published YYYY-MM-DD“

The protocol – step 2

- A tests n (if A trusts B, skippable)
  - Check that n is 160 digits and $n = 1 \bmod 4$

    $\Rightarrow$ n is odd and $(-1/n) = 1$
  - Check for some x that there exists a y so $x^2 = y^2 \bmod n$ and $(x/n) \neq (y/n)$

Universität Augsburg University

The protocol – step 2

- Testing procedure:
  - B => A: Select 80 random numbers x_i from Zn* and send x_i^2 mod n to A
  - A => B: Send a sequence of 80 random bits b_i to B, where b_i is either 1 or -1
  - B => A: For each i, send back x_i if b_i is 1 or y_i if b_i is -1
- This convinces A that the first condition of Theorem 1 holds. It fails with a probability of 2^(-80) < 1/(N_A)

Universität Augsburg University

The protocol – step 3

- B flips coins to A:
  - Mandatory signing of messages!
  - A checks the publication date
  - A => B: A selects 80 x_i from Zn* at random; then sends „n, publication date of n, x_i^2 mod n – signed by A"
- Delicate point for A: B might not respond, then claim A did not want to continue. The judge is needed then (Termination or forced completion of the protocol)

The protocol – step 3

- B checks n and it's publication date
  - B => A: „n, x_i^2 mod n, b_i, signed by B“
- A computes the Jacobi symbols of x_i and saves the results as Js_i
  - Js_i = b_i => r_i =1, else r_i = -1
- Now A knows what B flipped her, he has no idea

Universität
Augsburg
University

The protocol – step 3

- A => B: „x_i, signed by A“
  - Signature is not needed if B is confident A does not know the factorization of n
- B now confirms his flips by computing (x_i/n) and matching it to his guesses b_i => r_i
- If more flips are needed, the first two steps may be skipped as long as the same n is used
- END

Universität Augsburg University

Judge's protocol

- May be programmed in an ironclad fashion
- I case of dispute:
  - Subpoena all signed messages; if the case has exceeded the statute of limitations, throw it out
  - If A produces a signed messages relating to a signed message from B, he must present said message or be found guilty of cheating

Universität Augsburg University

Judge's protocol

- Test n as in step 2
- If no messages have been provably exchanged during step 3, terminate the protocol by signed message (even if A send her first message and B did not care)
- Otherwise, force completion of protocol
- Check that x_i^2 mod n in A's first signed message matches with quadratic residue mod n of x_i in the later A-signed message

=> if not, find A guilty of cheating

Universität Augsburg University

Judge's protocol

  - Determine r_i
  - A and B are given a signed message showing the judge's findings
- END

[7]

Universität Augsburg University

Literature

- Blum, Manuel. (1982). Coin Flipping by Telephone - A Protocol for Solving Impossible Problems.. Sigact News - SIGACT. 15. 133-137. 10.1145/1008908.1008911. [B'81]
- W. Diffie and M.E. Hellman. (1979). Privacy and Authentication: An Introduction to Cryptography.. Proc. IEEE, vol. 67 no. 3, 397-427 [DH'79]
- W.J. LeVeque. (1977). Fundamentals of Number Theory. Addison-Wesley Pub. [L'77]
- R.L. Rivest, A. Shamir, L.L. Adleman. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM, vol. 21, 120-126 [RSA'78]

Universität Augsburg University

Sources and pictures

- https://tvtropes.org/pmwiki/pmwiki.php/Main/AliceAndBob [1]
- http://www.magic-factory-essen.de/zubehoer/muenzen/doppelkopfmuenze-half-dollar-kopf.html [2]
- https://www.bernd-leitenberger.de/cray-1.shtml [3]
- https://www.heise.de/newsticker/meldung/Grafikkarte-Nvidia-GeForce-GTX-1080-Ti-angetestet-Hoechs [4]
- https://en.wikipedia.org/wiki/List_of_random_number_generators [5]
- https://en.wikipedia.org/wiki/Coprime_integers#/media/File:Coprime8.svg [6]
- https://www.amazon.com/Judges-Gavel-with-Striking-Block/dp/B01B3Z0R5S [7]

All sites last accessed on 23.7.18, 12:21

Universität Augsburg University

Thank you!

Source:

www.xkcd.com

Universität Augsburg University