

Introduction to Modern Cryptography

Benny Chor

Hard Core Bits
Coin Flipping Over the Phone
Zero Knowledge

Lecture 10 (version 1.1)

Tel-Aviv University

18 March 2008. Slightly revised March 19.

Hard Core Bits for One Way Functions

- Let $F : D \rightarrow D$ be a one-way, **one-to-one** function.
- Suppose $B : D \rightarrow \{0, 1\}$ is an **efficiently computable** predicate on elements of D .
- We say that B is a **hard core bit** for F if it is computationally hard, given $F(x)$, to determine $B(x)$.
- This task cannot be harder than **inverting F** .
- We want to **formalize** this notion.

Hard Core Bits for One Way Functions (2)

- We want to **formalize** this notion.
- **Definition:** We say that B is a **hard core bit** for F if the following holds:
- There is an efficient (possibly randomized) procedure \mathcal{P} , which gets the input $z = F(y)$,
- for any algorithm $\mathcal{A}(\cdot)$ that, on input $F(x)$, outputs $B(x)$,
- The procedure \mathcal{P} can adaptively generate queries $z_1 = F(y_1)$, $z_2 = F(y_2), \dots, z_k = F(y_k)$, and feed them to $\mathcal{A}(\cdot)$. The response to the queries are the bits $B(y_1), B(y_2), \dots, B(y_k)$.
- The running time of \mathcal{P} and the number of queries it makes, k , are both **polynomial** in $\log D$.
- Last but not least: The procedure \mathcal{P} correctly inverts F , namely it outputs $y = F^{-1}(z)$. ♠

Hard Core Bits for One Way Functions: Intuition

If B is a **hard core bit** for F then the ability to efficiently infer $B(x)$ from $F(x)$ **enables to invert F** .

Any algorithm $\mathcal{A}(\cdot)$ that, on input $F(x)$, outputs $B(x)$, enables the inversion of $F(x)$ by employing the procedure \mathcal{P} .

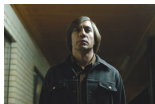
Thus if F is indeed hard (we assumed it is one-way) then $B(x)$ cannot be computed from $F(x)$.

Such “hard core bit” can serve as the basis for coin tossing over the phone.

Coin Tossing Over the Phone

Two non trusting parties wish to toss coins **over the phone**.

- The parties do not deviate from the **syntax** of the protocol.
- They may try to cheat in different ways, provided the messages they send have the right format.
- For example, Alice could send the string **00000** whenever the protocol calls for five **random bits**. (There is no way Bob can convincingly argue she is cheating here.)
- For a different example, Alice could send a **composite** number where the protocol calls for a **prime** number. (This specific attempt is not very smart, since Bob will easily detect it.)
- For yet a third example, Alice could send a product of three numbers **pqr** , where the protocol calls for a product of two, **pq** . (It is not clear if Bob could detect this.)



(Anton Shigur, the meanest coin tosser around.)



Coin Tossing Over the Phone: General Scheme

- Alice chooses an instance of a one-way, one-to-one function $F : D \rightarrow D$.
- Let $B : D \rightarrow \{0, 1\}$ be a hard core predicate for F .
- Alice starts by sending a description of F and of B to Bob.
- Alice picks an $x \in D$, computes $y = F(x)$ and $b = B(x)$.
- She sends y to Bob. This is supposed to create a **commitment** to the value x .
- Bob send to Alice his guess for $B(x)$, namely a bit $c \in \{0, 1\}$.
- The bit c could either be the result of a coin flip or the outcome of some efficient algorithm applied by Bob in an attempt to **guess** $B(x)$.
- After receiving c , Alice sends x to Bob, who can now compute $b = B(x)$ on his own.
- If $c = b$ then Bob wins the coin toss. Else ($c \neq b$) Alice wins.

Specific Examples (Good and Bad)

- $D = Z_p^*$, $F(x) = g^x \bmod p$ where p is a prime number, and g is a primitive element in Z_p^* . Let $B(x) = \text{Half}_p(x)$ (namely 0 if $1 \leq x < (p-1)/2$, and 1 if $(p-1)/2 \leq x \leq p-1$). **Good**, provided g is a primitive element.
- $D = Z_p^*$, $F(x) = g^x \bmod p$ where p is a prime number, the factorization of $p-1$ is known, and g is a primitive element in Z_p^* . Let $B(x)$ be the least significant bit of x . **Bad**.
- Let $D = Z_N^*$, $F(x) = x^e \bmod N$ where $N = pq$, both p and q are prime numbers, and e is relatively prime to $(p-1)(q-1)$. The numbers e and N are sent to Bob, but N 's factorization is not given. Let $B(x)$ be the least significant bit of x . **Good**, provided e is relatively prime to $\phi(N)$.
- Let $F : D \rightarrow D$ be any one-way, one-to-one function, let n be the number of bits of elements in D . Define a new one to one, one way function $G : D \times D \rightarrow D \times D$ by $G(x, r) = (F(x), r)$. Define $B(x, r) = \sum_{i=1}^n x_i \cdot r_i \pmod{2}$. **Good** – this is the generic hard core bit of Goldreich and Levin, 1989.

Coin Tossing Over the Phone and Hard Core Bits

Question: Does it suffice that B is a hard core bit for F in order to guarantee a **fair** coin toss? Or should the definition be **strengthened**?

In assignment 3 you are asked to think about this problem, and provide an educated answer.

Coin Tossing Over the Phone: Can Alice Cheat?

- Alice chooses an instance of a one-way, **one-to-one** function $F : D \rightarrow D$.
- Suppose Alice managed to choose a **two-to-one** function $F : D \rightarrow D$, but poor Bob could not tell the difference.
- Furthermore, suppose there are $x_0, x_1 \in D$ such that $F(x_0) = F(x_1)$ but $B(x_0) = 0, B(x_1) = 1$.
- In such case, Alice can win the coin toss with certainty.

- Bob is justified in being worried.
- But what can he do? He has neither time nor patience to go exhaustively over D and verify that F is **one-to-one** over it.

Coin Tossing Over the Phone: Can Alice Cheat?

- Bob is justified in being worried.
- But what can he do? He has neither time nor patience to go exhaustively over D and verify that F is **one-to-one** over it.
- In the specific examples we considered, there is a short proof that F is **one-to-one**.
- For example, $D = Z_N^*$, $F(x) = x^e \bmod N$ where $N = pq$, both p and q are prime numbers, and e is relatively prime to $\phi(N)$.
- If e is **not** relatively prime to $\phi(N)$, then $F(x) = x^e \bmod N$ is **k -to-one** for some $k \geq 2$.
- This means that for **every** $y \in Z_N^*$ in the range of F there are distinct $x_1, x_2, \dots, x_k \in Z_N^*$ such that $F(x_1) = F(x_2) = \dots = F(x_k) = y$.

Coin Tossing Over the Phone: Calming Down Bob

- Bob is justified in being worried.
- But what can he do? He has neither time nor patience to go exhaustively over D and verify that $F(x) = x^e \bmod N$ is **one-to-one** over it.
- If F is not one-to-one, then for **every** $y \in Z_N^*$ in the range of F there are distinct $z_1, z_2, \dots, z_k \in Z_N^*$ such that $F(z_1) = F(z_2) = \dots = F(z_k) = y$.
- Just after receiving the description of F , Bob picks one hundred elements in Z_N^* at random, z_1, z_2, \dots, z_{100} . He computes $y_i = F(z_i)$ ($i = 1, \dots, 100$), and sends y_1, y_2, \dots, y_{100} to Alice.
- Alice computes $F^{-1}(y_1), F^{-1}(y_2), \dots, F^{-1}(y_{100})$ and sends them to Bob. If for all of them $F^{-1}(y_i) = z_i$, then Bob is convinced that F is **one-to-one**. Otherwise, he **knows** that Alice is cheating, and quits the game.

Some Observations on Verifying F Is One-To-One

- It was crucial that Bob makes his queries **before** Alice committed to x by sending $F(x)$ (why?).
- The “protocol for convincing Bob” relied on two **very specific** properties of $F(x) = x^e \bmod N$.
- First, we used the fact that this is a **trapdoor function** and Alice should have the trapdoor information.
- Second, the **k -to-one** property is very specific to exponentiation modulo N .
- In general, we may want to convince a suspicious **verifier** that a certain claim holds, without giving away any additional information.
- Leads to **zero knowledge proofs** (Goldwasser Micali Rackoff 1985).

Zero Knowledge Proofs

Can I **convince** you that some statement \mathcal{S} holds, without giving you **any hint** about its proof?

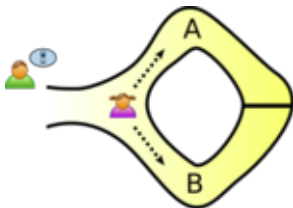
- **Not** by authority or intimidation!
- By some **proof system**, possibly randomized, where
- If \mathcal{S} is false, the probability that I convince you is smaller than ε (a small parameter).
- If \mathcal{S} is true, the probability that I convince you is larger than $1 - \varepsilon$.
- The text of the conversations, assuming \mathcal{S} is true, gives you **nothing more** – you could have generated very similar text **on your own**.

- Will now demonstrate this non intuitive notion using two examples: A light one (Ali Baba) and a heavier one (3 coloring).

Zero Knowledge Proofs: Ali Baba's Secret Passage

This zero knowledge version of the classic story is due to Quisquater and Guillou. It was taken (with the figures) from Wikipedia.

Tectonic activity had created a cave in the shape of a closed circle under a mountain side near Bagdad. The cave forks to two paths, **A** and **B**, soon after the entrance. If one follows any of the paths and walks (in the dark!) for an hour, s/he reaches a solid wall and must return the same way s/he came.

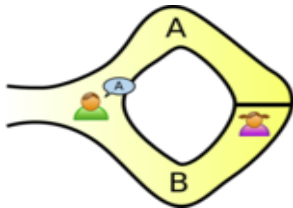


Rumor has it that the paths actually merge on the two sides of the solid wall, and that there is a **secret passage** through that wall, which can be opened after chanting a **secret password**.

Zero Knowledge Proofs: Ali Baba's Secret Passage (2)

Ali Baba (in purple) claims he knows the secret password. Babar (in green) is willing to pay him a small fortune for that information (he plans to open a "tour the cave" startup, and must guarantee a steady, fast flow).

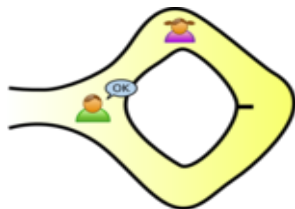
First, he want to be convinced that indeed Ali Baba has the secret. Ali Baba suggests the following protocol: He will enter the cave and proceed a few hundred meters along a path he (Ali Baba) **picks arbitrarily**. Ali Baba is very careful about his own privacy and does not agree to Babar getting close to him at this stage.



Babar then comes to the entrance and shouts either "A" or "B" (he picks one option **at random**).

Zero Knowledge Proofs: Ali Baba's Secret Passage (3)

Ali Baba returns to the entrance through the requested path, and Babar, sitting at the fork, can verify that.



What is Ali Baba's chance to succeed in case he **does not hold** the secret password? **Exactly 1/2**.

This by itself is not good enough to convince Babar – Ali Baba could just have been lucky in guessing what path Babar is going to call.

But if they repeat this experiment for thirty days, and Ali Baba succeeds in each and every one of them, Babar is convinced that Ali Baba **knows** the secret password.

Zero Knowledge Proofs: Ali Baba's Secret Passage (4)

- What did Babar learn from this lengthy test?
- He did learn that, with very high probability, Ali Baba knows the password for the secret passage.
- But other than that, Babar **learns nothing** – neither a single bit of the password, nor its quadratic residuosity, etc.
- This is a very simple example of a **zero knowledge protocol**.
- In such protocols there are two parties: The **prover** (Ali Baba in our case) and the **verifier** (Babar).
- They do not trust each other, yet engage in a protocol at the end of which the **prover** should **convince** the **verifier** that some statement is true **without supplying any additional information**.
- The formal definition takes a **simulator** that enables the verifier to generate a distribution of convincing conversations by himself.
- This seemingly contradictory notion was suggested by Goldwasser, Micali, and Rackoff in 1985.
- It was met initially with much skepticism, but is now widely accepted.