

Lecture 11 : October 9, 1997

*Lecturer: Ron Rivest**Scribe: Ching Law*

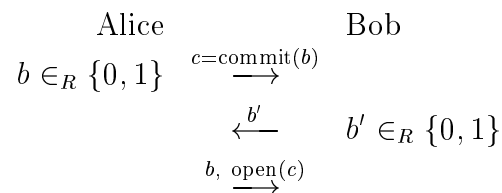
Take-home Midterm: Oct 30.

Topics Covered:

- Coin-flipping
- Proof of knowledge:
 - as identification protocol
 - definition
 - of discrete log
 - in zero knowledge

1 Coin-flipping

Alice and Bob wants to decide something on the phone. Can you flip coins on the phone?



Result = $b \oplus b'$. Both Alice and Bob cannot influence the result. They can also play other games with similar protocols.

2 Proof of Knowledge

Alice (Prover) knows x such that $y = g^x \pmod{p}$, (x, g, p are public). Alice wants to prove that she knows x to Bob (Verifier). For example, in a login system, Bob

is the computer and x is the key for identification. How to prove knowledge of x without revealing x , or any information about x ?

3 Interactive Protocol

An interactive protocol is a specification of a back and forth dialogue between a Prover and a Verifier. At the end of which the Verifier either “accepts” or “rejects”. The Verifier accepts if he is convinced that the Prover knows x . The interactive protocol has multiple rounds of ‘proofs’, compared with a single one-way statement in an ordinary proof.

Completeness If P knows x , then V accepts.

Soundness If V accepts, then P knows x .

Zero-knowledge V learns nothing (zero), except that P knows x . (V learns nothing about x .) (The protocol does not leak any information about x .)

Protocol for proving knowledge of Discrete Logarithm:

Prover knows x such that $y = g^x \pmod{p}$.

Repeat the following round t times:

Prover		Verifier
$k \in_R Z_{p-1}$	$\xrightarrow{s=g^k \pmod{p}}$	
	\xleftarrow{c}	$c \in_R \{0, 1\}$
	$\xrightarrow{r=k+cx \pmod{p-1}}$	$sy^c \stackrel{?}{=} g^r$

Verifier accepts if the check $sy^c \stackrel{?}{=} g^r$ always succeeds.

3.1 Completeness

If $c = 0$, $g^r = g^k = s = sy^0 = sy^c$.

If $c = 1$, $g^r = g^{k+x} = g^k g^x = sy^1 = sy^c$.

3.2 Soundness

If Verifier accepts, then Prover knows x .

Definition 1 *A program P knows x in a given state if it is possible to easily extract x by examining P 's outputs to several different inputs (from a same given starting state).*

In this case, we have:

$$\begin{array}{ll} \text{input } c = 0 & \text{output } k \\ \text{input } c = 1 & \text{output } k + x \pmod{p-1} \end{array}$$

Thus $x \pmod{p-1}$ is known by P . Since P has demonstrated her ability to respond to challenges, she must be prepared to respond either way.

Can she cheat?

$$\begin{array}{ll} P \text{ guesses } c = 0 & : \text{ pick } k \in_R Z_{p-1} \\ & \text{output } s = g^k \pmod{p} \\ & r = k \\ \\ P \text{ guesses } c = 1 & : \text{ pick } \overbrace{k+x}^r \in_R Z_{p-1} \\ & \text{output } s = \frac{g^{k+x}}{g^x} = \frac{g^{k+x}}{y} \pmod{p} \\ & r = k + x \end{array}$$

She can only succeed with probability 2^{-t} by guessing the challenges (c). Therefore, P must 'know' x .

4 Zero-knowledge

We will show that the Verifier learns nothing about x (except that P knows x).

Definition 2 *A transcript is a record of all messages, all coin flips and all public information seen by the Verifier. It is what the Verifier "takes home" when the protocol is over.*

A transcript has a probability distribution depending on the random coin flips of P and V . The Verifier gets a sample of transcript according to the probability distribution. However, we claim that V can sample transcripts with the same probability distribution (perfect Zero-knowledge!) on his own. This implies that P has not given V anything that V cannot get on his own, and thus P has released zero knowledge about x by engaging in the protocol (she is not worse off).

The Transcripts of the Interactive Protocol

s is uniform in Z_p^*
 c is uniform in $\{0, 1\}$
 r is uniform in Z_{p-1}

Such that $sy^c = g^r$

Construction of sample transcripts without knowing x

$$\left[\begin{array}{l} \text{pick } c \in_R \{0, 1\} \\ \text{pick } r \in_R Z_{p-1} \\ \text{compute } s = \frac{g^r}{y^c} \pmod{p} \\ \text{output } (s, c, r) \end{array} \right] \text{ repeat for } t \text{ rounds}$$

Since we have an identical distribution here, an honest verifier gains zero knowledge.