

Interactive and Zero Knowledge Proofs

Prof. Zeph Grunschlag

Interactive Proof Systems

- Prover *Peggy* wishes to prove knowledge of secret information to Verifier *Vera*. Communicate through a sequence of rounds $P-V-P-V\dots-P-V$
- Each round (except last): participant *receives* message from opposite, *computes* based on previous information, and *sends* message to opposite or ABORTs.
- Final round: *V* sends no message, and either ACCEPTs or REJECTs *P*'s proof.

3

Identification Schemes

- Peggy: Can you let me in please?
- Vera: Who are you?
- Peggy: I'm Peggy.
- Vera: What's your password?
- Peggy: None of your beeswax
- Vera: Get lost...

2

Completeness and Soundness

For Interactive Proof System to work, Peggy should usually be able to prove her knowledge while impostor Pernicia (P^*) should usually fail.

- Completeness: When possessor of secret P interacts, prob. of V accepting $\geq \frac{2}{3}$
- Soundness: When non-possessor of secret P^* interacts, prob. of V accepting $\leq \frac{1}{3}$

4

Password Identification Scheme

1. P : Sends password x
2. V : Checks x against stored password y .

ACCEPTs iff $x == y$

Transcript diagram:

x
P

 ← transcript
← action

CLAIM: System is Complete and Sound.

ISSUES

- If Eve sees transcript, can successfully pretend to be Peggy
- If Villain V^* replaces V , can pretend as well

5

ATM Scam

Swipes ATM card	Enters Pin	“Sorry. Out of order.”
P		V^* : Stores P 's info for later

Now V^* makes copy of P 's ATM card using stored info and harvests money from P 's account with card-copy and PIN.

6

PKE for Identification

First attempt to fix: use secret key to decrypt encrypted messages, thus not revealing secret key. First round omitted as Peggy does nothing.

Peggy proves that she knows sk :

$c = E(m, pk)$	$m' = D(c, sk)$
V : random message m	P

V accepts iff $m == m'$.

CLAIM: System is Complete and Sound.

ISSUE: V^* tricks P into decrypting ciphertext c with unknown plaintext.

7

Secrecy = Transcript Indistinguishability

If information is leaked, true transcripts inherently distinguishable from Simon's (S) simulations ignorant of P 's secret:

- Password Scheme Leakage:

Peggy:

andromeda
P

 Simon:

^%\$!@*%\$!
S

- PKE Scheme Leakage:

Peggy:

c^*	$m' = D(c^*, sk)$
V^*	P

 Simon:

c^*	$m''(c^*)$
V^*	S

8

Zero-Knowledge Interactive Proofs

DEF: An interactive proof system (P, V) is **(perfect) zero-knowledge** if there is a simulator S ignorant of P 's secret info. but able to reproduce the transcripts of any adversary V^* interacting with P with the same probability distribution as actual interactions.

Related notions. Replace “perfect” by...

- computationally secure - require slightly subtler complexity-based definitions
- statistical - differs for finitely many cases

9

Proof that Simple Fiat-Shamir is a ZKIP

1. Prove that protocol defines Interactive Proof: i.e. Sound and Complete
2. Prove that protocol reveals zero knowledge. Simon defined by:

$s = r^2 / y^{b'}$ mod n	b	r
S : guesses $b' \in_U \{0, 1\}$ generates: $r \in_U \mathbb{Z}_n^*$	V^*	S : Starts over if $b \neq b'$ Else: sends above message

Simon is a Las-Vegas algorithm.

11

Removing Information Leaks: Fiat-Shamir

Simplified version of Fiat-Shamir:

- Public Information: n - a product of *discarded* equal-length distinct primes $p, q \equiv 3 \pmod{4}$ and y - a quadratic residue mod n
- Peggy's secret: x - a square root of y mod n
- Protocol defined by:

$s = r^2$ mod n	b	$t = rx^b$ mod n
P : $r \in_U \mathbb{Z}_n^*$	V : $b \in_U \{0, 1\}$	P

- Victor ACCEPTs iff $t^2 \equiv_n sy^b$

10

Complexity Theory View

- Identification Protocols = Interactive Proof Systems = Languages
- Identification Protocols:
 1. P knows “password”
 2. Uses protocol to convince V
- Interactive Proof Systems:
 1. P knows proof to theorem
 2. Uses interactive proof to convince V
- Languages:
 1. P knows why string x is in language L
 2. Interactive proof shows membership

12

Complexity Theorems

1. Shamir proved: **IP = PSPACE**

2. If one-way functions exist: **CZK = IP**

COR: With the aid of a one-way function, any interactive proof system can be converted to a computational zero-knowledge proof system.

- **CZK**: computational zero knowledge lang's
- **IP**: interactive proof languages
- **PSPACE**: polynomial space languages
- Recall **NP PSPACE**