

Zero Knowledge Interactive Proof

Eerke Boiten, University of Kent

21 Nov 2006

Abstract

- There are two parties in this seminar: the presenter and the audience. The presenter knows a secret (namely, what *zero knowledge* and *interactive proof* mean in the context of cryptography), and we'll assume that the audience doesn't know it. If, by the end of the seminar, the audience is convinced that the presenter knows the secret, without the audience having learned anything about the secret other than that fact, then this will have been a zero-knowledge seminar.

What and why?

- Looking at crypto/FM interface
- Interesting subject?
- Deceptive terminology!
- Example of common structure in this area

Introduction

- Purpose: identification (entity authentication)
(Sometimes combined with key exchange)
- Two parties: “prover”/“claimant” P , “verifier” V
- pre: P knows a “secret” (e.g. a key, pin ...)
- post: V knows that P knows
- and a whole lot more as this is security!

Basic schemes (1)

P:

say!P!V!password

V:

say?P!V?pw →

([pw = pwtable(P)] →
accept

□

[pw ≠ pwtable(P)] →
reject

)

Basic schemes (2)

P:

say!P!V!H(password)

V:

say?P!V?h →

([h = hptable(P)] →
accept

□

[h ≠ hptable(P)] → reject

)

Basic schemes (3)

P:

say!P!V→

say!V!P?m →

say!P!V!decrypt(sk_P,m)

V:

say?P!V→

generate?c →

say!V!P!encrypt(pk(P),c) →

say!P!V?d →

([d =c] → accept

□

[d≠ c] → reject

)

Interactive Proof: properties

- Soundness: if the verifier accepts, then the prover knows the secret.
[If the prover doesn't know the secret and the verifier accepts, then the secret isn't.]
- Completeness: if the prover knows the secret, the verifier will “always” accept.
- Security properties galore.
- Complexity. Extra: mutual? key exchange?

Intermezzo: number theory

Take two large primes p and q , let $n=pq$.

- \mathbb{Z}_n^* is the set of all numbers in \mathbb{Z}_n prime to both p and q , a multiplicative group.
- Computing square roots in \mathbb{Z}_n^* is as hard as factoring.
- All arithmetic modulo n for now.

Fiat-Shamir protocol (simplified)

P's secret: y (is root of x)

P:

$\oplus_{r \in \mathbb{Z}^*_n}$ say! $r^2 \rightarrow$

say? $e \rightarrow$

say! $(r * y^e)$

V:

say? $a \rightarrow$

$\oplus_{e \in \{0,1\}}$ say! $e \rightarrow$

say? $b \rightarrow$

$[b^2 = ax^e] \rightarrow$ accept

□

$[b^2 \neq ax^e] \rightarrow$ reject

)

$\oplus_{x \in S}$ is probabilistic
uniform choice

Analysis of Fiat-Shamir

- Completeness is obvious: $b^2 = r^2y^{2e} = r^2x^e = ax^e$ is the correct test to perform.
- Soundness is not quite there yet. A “dishonest prover” can pre-guess e , and send out r^2/x^e and r of his own choice with 50% chance of success. Improve by iterating the procedure. (“Full version” has a vector of secrets + iteration.)

Security properties (attacks)

- successful identification not transferable
- impersonation
- even if large number of previous successful runs observed, or participated in, or several runs are interleaved

Zero-knowledge

- Informally: observation/participation in protocol doesn't convey information not already polytime computable from public information.
- Formally: there's a polytime simulator which can simulate "accepting transcripts"
- The proof is construction of a simulator (later!)

Small step back: soundness

- Soundness (officially) = a dishonest prover that has a “too large” chance of succeeding can be used as a subroutine by a simulator that extracts (information equivalent to) P’s secret
- P’s secret being Hard To Compute completes reductio ad absurdum

Fiat-Shamir: 50%

>50% is too large (i.e.: “impossible”) for our example.

If >50% then for some a can produce both root of a and root of ax , divide to get ...

Proof of zero knowledge

- Construct the simulator! Unsatisfactory!
- Fortunately, it's already mentioned: pre-guess e .
- General strategy: try (only poly often) until you succeed. Here: expected twice.
- For version which iterates t times, need 2^t tries which can be kept within poly if ...

Simulator: final details

- Need to produce not only accepting transcripts but also duplicate their distribution.
- Fortunately \mathbb{Z}_n^* gives guarantees that distributions of random*fixed not distinguished from random

Constructive cryptography

- Universal quantifications over all polynomial probabilistic algorithms: what other chances than ad absurdum?
- Protocols are inventions (trial and error). Incremental approach can be made somewhat respectable (e.g. CSP, Paulson)
- Cry for compositionality?
- Never ascribe to malice that which can be adequately explained by non-determinism?