Cryptographic Protocols

Spring 2017

Part 2

Polynomial, Negligible, Noticeable

Function $f : \mathbb{N} \rightarrow \mathbb{R}$

- f is polynomial $\Leftrightarrow \quad \exists c \;\; \exists n_0 \forall n \geq n_0 : \;\; f(n) \leq n^c$
- f is negligible $\Leftrightarrow \quad \forall c \;\; \exists n_0 \forall n \geq n_0 : \;\; f(n) \leq \frac{1}{n^c}$
- f is noticeable $\Leftrightarrow \quad \exists c \;\; \exists n_0 \forall n \geq n_0 : \;\; f(n) \geq \frac{1}{n^c}$

Implications

- poly $\times$ poly = poly
- poly $\times$ negligible = negligible (cannot be amplified)
- poly $\times$ noticeable = „large enough“ (can be amplified)

P, NP, PSPACE, etc

Running Time of a fixed given TM (aka algorithm)

- for input x: number of steps $s(x)$
- for n-bit input: $t(n) :=$ max$\{s(x) : x \in L, |x| \leq n\}$ (worst-case)
- TM is polynomial iff $t(n)$ is a polynomial function

Complexity Classes

- P $= \{L : \exists$ polytime TM that accepts $L\}$
- NP $= \{L : \exists$ non-det. polytime TM that accepts $L\}$ (German script)
  NP $= \{L : \exists$ poly TM s.t. $(x \in L \Leftrightarrow \exists w :$ TM$(x, w) = 1)\}$ (Engl. scribe)
  $\rightarrow$ Thm 1.8: These two definitions are equivalent!
- NP-hard $= \{L : \forall L' \in NP : L'$ can be reduced to L$\}$
- NP-Complete $=$ NP $\cap$ NP-hard
- PSPACE $= \{L : \exists$ TM that accepts L with poly memory (in any time)$\}$

Interactive Proofs of Statements

Def: TM accepts language L iff $x \in$ L $\Leftrightarrow$ TM(x) outputs 1

Def: An interactive proof for language L is a pair (P,V) of int. programs s.t.

i) $\forall x$: running time of V is polynomial in $|x|$
ii) $\forall x \in L$: Pr((P $\Leftrightarrow$ V) $\rightarrow$ “accept”) $\geq 3/4$ $[p = 3/4]$
iii) $\forall x \notin L, \forall$P' : Pr((P' $\Leftrightarrow$ V) $\rightarrow$ “accept”) $\leq 1/2$ $[q = 1/2]$

Remarks

- Constants p, q are arbitrary, could be $p = 1-2^{-|x|}$ and $q = 2^{-|x|}$
- However: only NP-languages have proofs with $q = 0$
- If iii) holds only for poly P' $\rightarrow$ interactive argument
- Probabilistic P is not more powerful than deterministic P

Examples: Sudoku, GI, GNI, Fiat-Shamir,

Zero-Knowledge

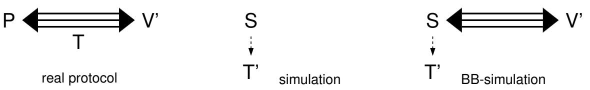
Idea: Protocol (P,V) has transcript T, Simulator S outputs similar T'.

Def: (P,V) is **zero-knowledge (ZK)** $\Leftrightarrow \forall$ V' $\exists$ S:

i) Transcript T of (P $\Leftrightarrow$ V') and output T' of S are indistinguishable,
ii) Running time of S is polynomially bounded in running time of V'.

Def: (P,V) is **black-box zero-knowledge (BB-ZK)** $\Leftrightarrow \exists$ S $\forall$ V':

i) Transcript T of (P $\Leftrightarrow$ V') and output T' of S in (S $\Leftrightarrow$ V') are indist.,
ii) Running time of S is polynomially bounded.



Def: (P,V) is **honest-verifier zero-knowledge (HVZK)** if S exists for V' = V.

Types of ZK: perfect, statistical, computational.

c-Simulatability

Definition: A three-move protocol (round) with challenge space C is **c-simulatable** if for any value $c \in C$ one can efficiently generate a triple (t, c, r) with the same distribution as occurring in the protocol (conditioned on the challenge being c).

Formally: The cond. distribution $P_{TR|C}$ is efficiently samplable.

Lemma: A 3-move c-simulatable protocol is HVZK.
(assumption: challenge is efficiently samplable)

Lemma: A sequence of HVZK protocols is a HVZK protocol.

Lemma: A sequence of ZK protocols is a ZK protocol.

Lemma: HVZK round with c uniform from C, $|C|$ small, is ZK.