

I) Zero knowledge proofs

(Goldwasser, Micali, Rackoff 1982)

Excellent technical introduction: chapter 13 in Boaz Barak's cryptography book

Some key concepts

- Proof system
- Interactive proof
- Roles: Prover (P), Verifier (V)
- Desirable properties
 - Completeness - everything works if everyone is honest & behaves
 - Soundness (\Leftarrow knowledge soundness) - a bad P will fail
 - Zero knowledge - V obtains no add'l info besides correctness

Example I (Prover) want to convince you (Verifier) that I can distinguish two colors that you see as identical

Prover

Here are two items
A & B different colors

Verifier

(V sees A & B as identical)

flips a coin secretly,
if head, swap A & B without
P looking

challenge: here are the
two items, Did I swap them?

answers the challenge

checks P's answer

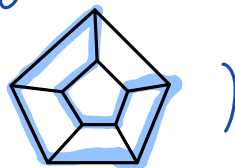
(repeat 100 times)

Hamilton cycle [Blum '87]

Common knowledge : a graph G

P wants to prove to V that he knows a **Hamilton cycle** of G with out revealing any additional information

(Hamilton cycle : a cycle going through every vertex of the graph exactly once & returns to the start

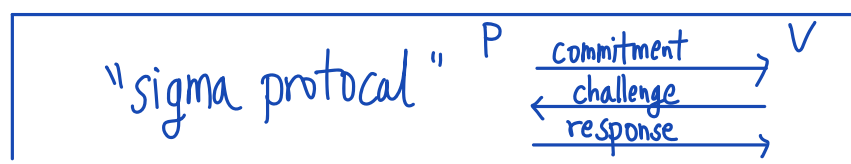
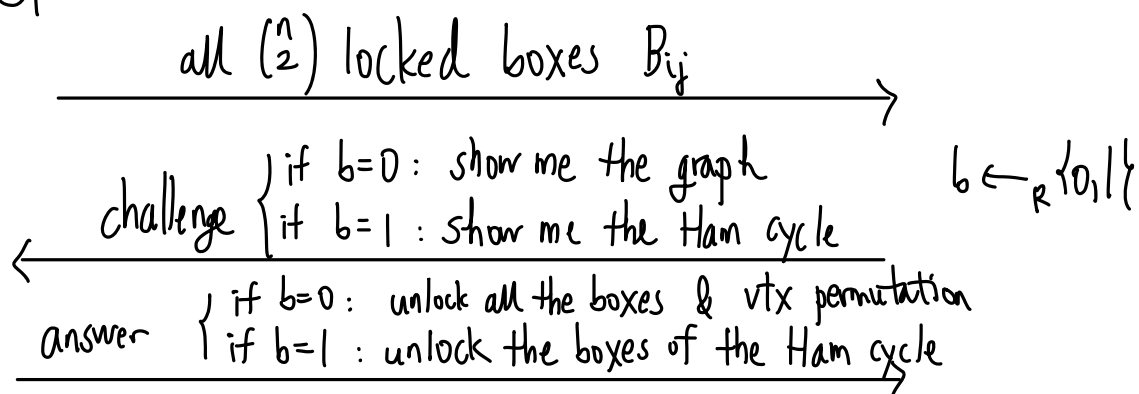


P

V

Label vtx $1, \dots, n$ according to some privately randomly chosen permutation (remember this permutation)

Btwn every pair ij of vtx, put in a locked box B_{ij} the bit indicating whether ij is an edge of G



check:
if $b=0$: same graph
if $b=1$: Ham cycle

Commitments

physical
(locked box)

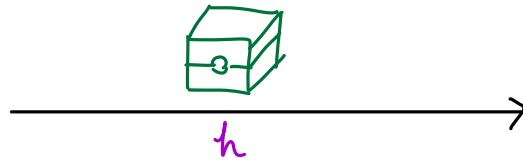
digital
(cryptographic)

P

secret data s

lock data
in a box

$r \leftarrow_R \{0,1\}^{256}$
 $h \leftarrow \text{hash}(s,r)$



open it



s, r

V

check $h = \text{hash}(s,r)$

Properties

Completeness If everyone behaves, then protocol accepts

Soundness If there is no Ham cycle, then no matter what P does, V rejects with prob $\geq \frac{1}{2}$

(There is a stronger requirement called "knowledge soundness" which says that even if the graph has a Ham but the prover doesn't "know" it, the protocol would still fail. The precise definition involves an extractor with rewinding abilities...)

Zero-knowledge If V accepts, then it learns no add'l info from the interaction, because V could have simulated the entire dialog by itself

Universal ZKP

Hamiltonicity is NP-complete

(every NP problem can be polynomially reduced to HAM)

P

Common: x, f

V

I know s s.t.
 $f(x, s) = 1$

P knows $s : f(x, s) = 1$

\Downarrow NP reduction

\Uparrow

I know a Ham
cycle in graph G

$G, \text{ZKP of HAM}$

\longleftrightarrow

convinced that P knows
Ham of G

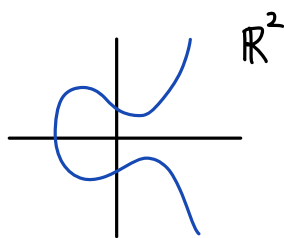
Remark Can turn an interactive proof into
a non interactive proof via the Fiat-Shamir heuristic

P can simulate V : whenever V picks a random value,
 P can simulate V 's randomness by running
a cryptographic hash function on the transcript
(so that P can't cheat by choosing favorable
"random" challenges)

Above construction of universal ZKP is not succinct
(also not practical). Later in the course: zk-SNARK

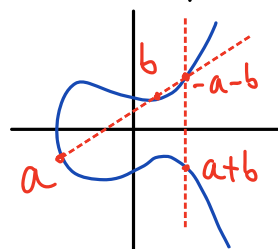
II) Elliptic curves

$$y^2 = x^3 + Ax + B$$



In cryptography,
work over \mathbb{F}_p instead of \mathbb{R}

Group law: addition of points on elliptic curve



$\bullet 0 = \text{pt at infinity}$

\leadsto abelian (commutative) group G whose elements are points on the curve

We write group operation additively $(G, +)$

Fix a generator $g \in G$ of prime order $q \approx 2^{256}$

In practice, (G, g) chosen from some standard (eg. NIST)

Discrete log: secret $x \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, public $xg \in G$
recovering x from xg is hard "Discrete Log Assumption (DLA)"

Diffie-Hellman key exchange

Alice
 $\alpha \leftarrow_{\mathbb{R}} \mathbb{Z}_q$

αg

Bob
 $\beta \leftarrow_{\mathbb{R}} \mathbb{Z}_q$

βg

shared secret: $\alpha\beta g$ (hard to compute given $\alpha g, \beta g$)

"Computational Diffie-Hellman (CDH) assumption"

Also "Decisional Diffie Hellman (DDH) assumption": hard to distinguish
 $(\alpha g, \beta g, \alpha\beta g)$ $\alpha, \beta \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ vs. $(\alpha g, \beta g, \gamma g)$ $\alpha, \beta, \gamma \leftarrow_{\mathbb{R}} \mathbb{Z}_q$

Schnorr protocol

P wants to convince V that

P knows a secret $s \in \mathbb{Z}_q$ s.t. $\underbrace{x = sg}_{\text{public}} \in \mathbb{G}$

$\overset{P}{s \in \mathbb{Z}_q, x = sg \in \mathbb{G}}$

$r \leftarrow_R \mathbb{Z}_q$

$\xrightarrow{u = rg}$

$\overset{V}{x \in \mathbb{G}}$

\xleftarrow{c}

$c \leftarrow \mathbb{Z}_q$

$\xrightarrow{z = r + cs}$

check $zg = u + cx$

Claim The above is a ZKP of discrete log.
(proof omitted)

Remarks

(I) Can be turned into a NIZK (noninteractive) via Fiat-Shamir, by setting $c = \text{Hash}(x, u)$

(II) Can be furthermore turned into a digital signature scheme
To sign message m , set $c = \text{Hash}(x, u, m)$ and publishing (c, z) as sig for m
(secret key = s , public key = $sg = x$)

Pairing based cryptography

Given cyclic groups G_0, G_1, G_T all same prime order q ,

a **pairing** is a nondegenerate bilinear map

$$e: G_0 \times G_1 \rightarrow G_T$$

(A) bilinear : $e(x+x', y) = e(x, y) + e(x', y)$
& $e(x, y+y') = e(x, y) + e(x, y')$ $\forall x, x', y, y'$
($\Rightarrow e(ax, y) = ae(x, y) = e(x, ay) \quad \forall a \in \mathbb{Z}_q$)

(B) nondegenerate : with generators $g_0 \in G_0$ & $g_1 \in G_1$
 $g_T := e(g_0, g_1) \in G_T$ is a generator

(Weil, Tate, ...)

Certain elliptic curves have useful pairings

- efficiently computable
- cryptographic hardness assumptions

An application : BLS signature scheme (Boneh-Lynn-Shacham)

Keygen : $sk := s \leftarrow_R \mathbb{Z}_q$ $pk := sg_0 \in G_0$

Sign(sk, m) $\rightarrow \sigma := s H(m) \in G_1$ $H(m) \in G_1$ hash of message

Verify(pk, m, σ) \rightarrow check $e(pk, H(m)) \stackrel{?}{=} e(g_0, \sigma)$

• Can be extended to allow signature aggregation