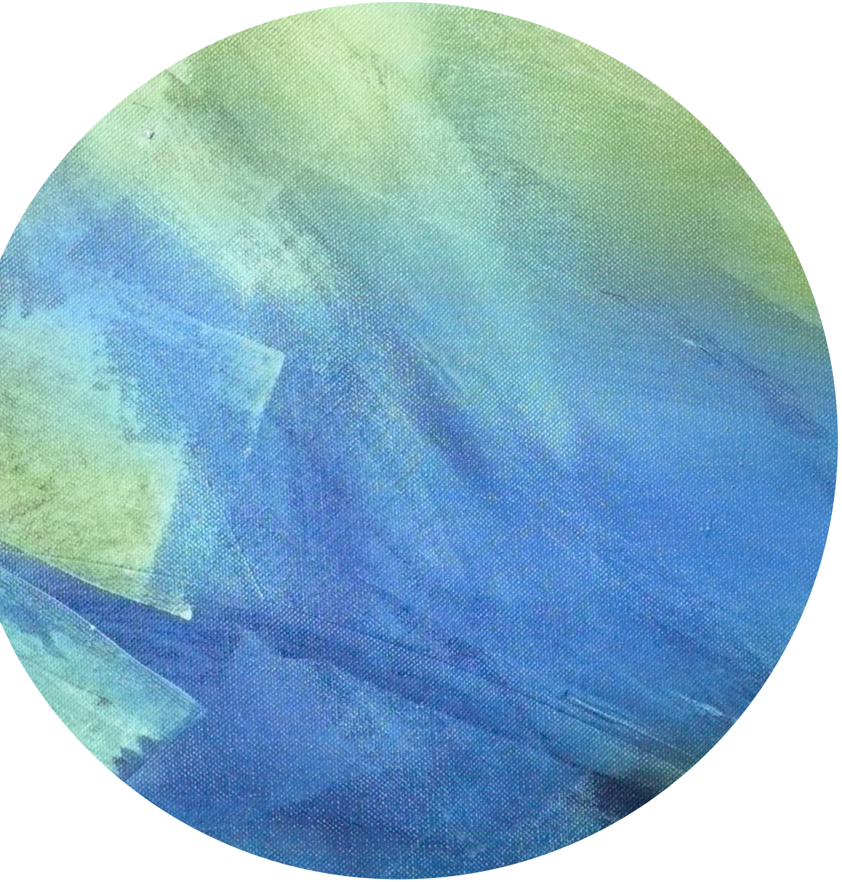
Zero-Knowledge Proofs.

Lecture 20

Scribe: Alan

Outline



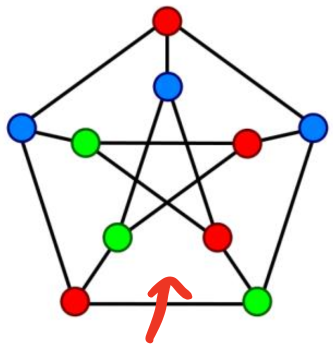
Zero-Knowledge for 3-coloring

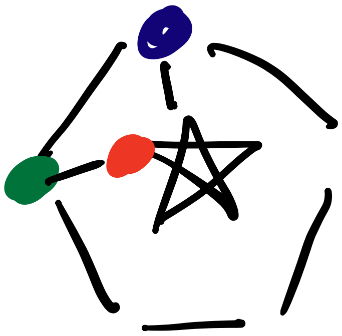
~~Garbled Circuits~~

NP-complete problem

ZK for 3-coloring

3-coloring

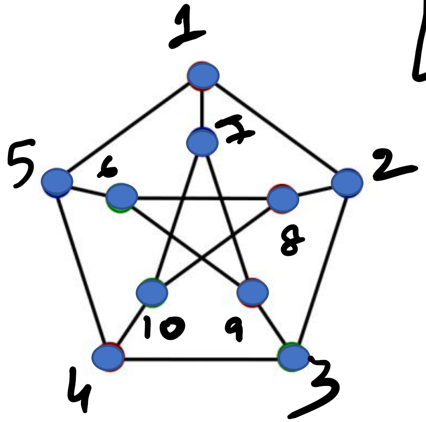




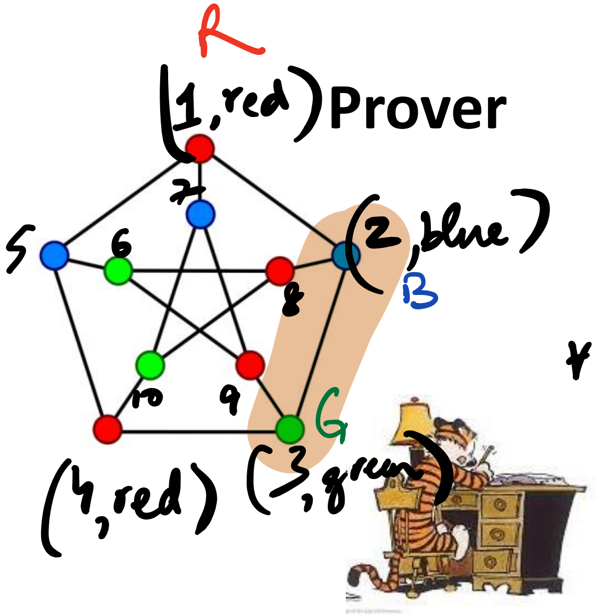
Color all vertices with only three colors (R, G, B) such that no edge should connects two vertices of the same color.

COMMITMENT. [Hiding] [Binding]

3-coloring



Prover



Verifier

$\forall i \in [n]$, com (i, color$_i$) →

← vertices (j, k)

decommit (2, blue), decommit (3, green) →

(color$_1$ above "blue", color$_2$ above "green")

Accept if color$_1$ ≠ color$_2$ and colors in RBG.

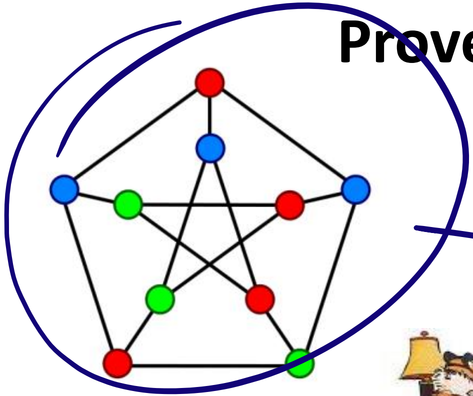
If G is not 3-col ⇒ ∃ at least 1 edge with same colors.

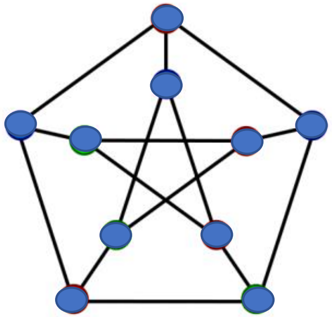
$\Pr[\text{ V picks } e \mid e \text{ has same colors }] = \frac{1}{\#\text{edges}}$

3-coloring

R → R
B   B
G   G

Prover





Verifier

repeat N times (n is security param):

com(i, $color_i$) →

← random edge (j, k)

(j, $color_j$), (k, $color_k$) →

SOUNDNESS

Let's s x ∉ L.

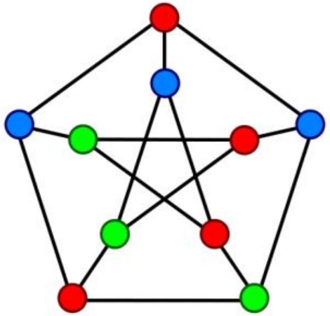
In each run, $\Pr[V \text{ accepts}] = 1 - \frac{}{\#\text{edges}}$.

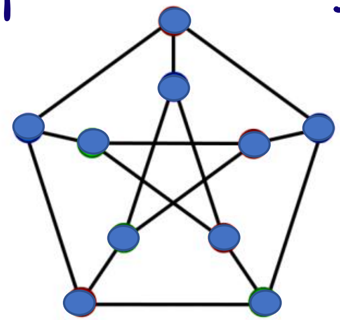
3-coloring

$\Pr[V \text{ accepts all runs}] = \left(1 - \frac{1}{\#\text{edges}}\right)^N$.

If $N = (\#\text{edges})^2$. $= \text{negl}(\#\text{edges})$

Prover





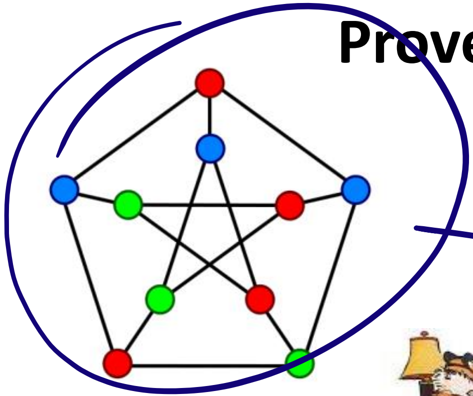
Verifier

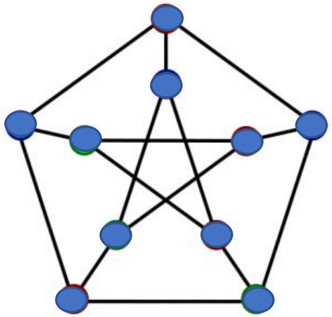
$\left(\left(1 - \frac{1}{\#\text{edges}}\right)^{\#\text{edges}}\right)^{\#\text{edges}}$.

$\left(1 - \frac{1}{\#\text{edges}}\right)^{\#\text{edges}} \approx \frac{1}{e}$

3-coloring

Prover





Verifier

repeat N times (n is security param): (each time with permuted colors)

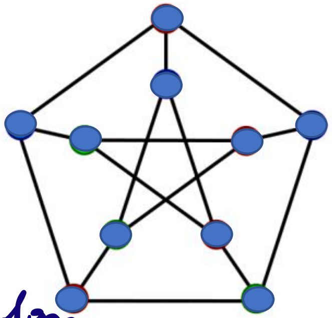
com $(i, color_i)$ →

random edge (j,k) ←

$(j, color_j), (k, color_k)$ →

3-coloring

Simulator



(color$_{j'}$, color$_{k'}$) are diff. random colors

G ∈ 3col.

com(j', color$_{j'}$), com(k', color$_{k'}$), com(0_s) everywhere else

edge(j, k)

Guess(j', k')

if $(j,k) = (j',k')$ → decommit (j', col$_{j'}$), decom(k', col$_{k'}$)

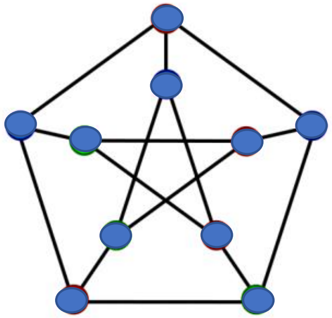
com( )

edge($\tilde{j}$, $\tilde{k}$)

Verifier

3-coloring

Simulator

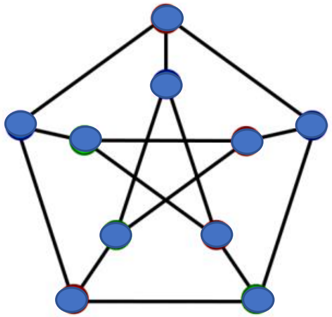


Verifier

Running time of Simulator = N. (# attempts needed so that $(j', k') = (j, k)$)

3-coloring

Simulator



Verifier

$Enc(\underline{s}\,; r)$

Proofs on Encrypted Data

Chaum-Pedersen

- Public parameters: (p,g,h)
  - p: large prime (1024 bit)
  - g: generator

- Proof of a given triple being of the following form

  $(u, v, w) = (g^a, g^b, g^{ab})$

- NP relation $\mathcal{R} = \{ (u, v, w) : \exists\ (a, b) \text{ s.t. } u = g^a, v = g^b, w = g^c \}$

Chaum-Pedersen

Prover | **Verifier**

(u, v, w)

$d \leftarrow Z_p$, $(v' = g^d, w' = g^{ad})$ → (v', w') →

$c \leftarrow Z_p$

← c ←

$e = d + b.c$ → e →

$(u, v, w) = (g^a, g^b, g^{ab})$

Check$_1$: $g^e = v'.(v)^c$

Check$_2$: $u^e = w'.(w)^c$

Chaum-Pedersen

Prover **Verifier**

(u, v, w)

$d \leftarrow Z_p$, $v' = f(d)$, $w' = f(ad)$

(v', w') →

$c \leftarrow Z_p$

← c

$e = d + b.c$

e →

$(u, v, w) = (f(a), f(b), f(ab))$

$Check_1$: $f(e) = v'.(v)^c = f(d). (f(b))^c$

$Check_2$: $f(ae) = w'.(w)^c = f(ad). (f(ab))^c$

Chaum-Pedersen: Zero-Knowledge

Simulator | **Verifier**

(u, v, w), guess c

$e \leftarrow Z_p$, $(v' = g^e/v^c, w' = u^e/w^c)$

Simulator → Verifier: (v', w')

Verifier → Simulator: c

Simulator → Verifier: e

~~$(u, v, w) = (g^a, g^b, g^{ab})$~~

Chaum-Pedersen: Soundness

Prover

(u, v, w)

d ← Z_p, (v' = g^d, w' = g^{ad})

(v', w') → (v', w') →

c ← c' ←

e= d + b.c

e → e' →

(u, v, w) = (g^a, g^b, g^{ab})

e' = d + b.c'

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R}$ = $\{$ P, $(u_1, u_2, ...u_n)$: $\exists$ $(a_1, a_2, ...a_n)$ s.t. $u_i = \Pi g^a$, and $P(a_1, a_2, ...a_n)$ = true$\}$

P, $(u_1, u_2, ...u_n)$

P, $(u_1, u_2, ...u_n)$

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R}$ = { P, $(u_1, u_2, \ldots.u_n)$: ∃ $(a_1, a_2, \ldots.a_n)$ s.t. $u_i = \Pi g^{a}$, and $P(a_1, a_2, \ldots.a_n)$ = true}

P, $(u_1, u_2, \ldots.u_n)$

P, $(u_1, u_2, \ldots.u_n)$

d ← Z_p, $(u'_i = \Pi g \quad)$

$(u'_1, u'_2, \ldots.u'_n)$ →

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R} = \{ P, (u_1, u_2, \ldots u_n) : \exists\ (a_1, a_2, \ldots a_n) \text{ s.t. } u_i = \Pi\, g^{a}\ \ , \text{and } P(a_1, a_2, \ldots a_n) = \text{true} \}$

$P, (u_1, u_2, \ldots u_n)$

$P, (u_1, u_2, \ldots u_n)$

$d \leftarrow Z_p, (u'_i = \Pi\, g\ \ )$

$(u'_1, u'_2, \ldots u'_n)$ →

c ←

$c \leftarrow Z_p$

General Linear Relations on Exponents

Prover | **Verifier**

NP relation $\mathcal{R}$ = $\{$ P, $(u_1, u_2, \ldots .u_n)$: $\exists$ $(a_1, a_2, \ldots .a_n)$ s.t. $u_i = \Pi\, g^a$, and $P(a_1, a_2, \ldots .a_n)$ = true$\}$

P, $(u_1, u_2, \ldots .u_n)$

$d \leftarrow Z_p$, $(u'_i = \Pi\, g \quad)$

$e_j = d + a_j.c$

$(u'_1, u'_2, \ldots .u'_n)$ →

c ←

P, $(u_1, u_2, \ldots .u_n)$

$c \leftarrow Z_p$

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R}$ = { P, $(u_1, u_2, \ldots. u_n)$: ∃ $(a_1, a_2, \ldots. a_n)$ s.t. $u_i = \Pi\, g^{a}$, and P$(a_1, a_2, \ldots. a_n)$ = true}

P, $(u_1, u_2, \ldots. u_n)$

P, $(u_1, u_2, \ldots. u_n)$

$d \leftarrow Z_p$, $(u'_i = \Pi\, g \;\;)$

$(u'_1, u'_2, \ldots. u'_n)$ →

c ←

$c \leftarrow Z_p$

$e_j = d + a_j.c$

$e_1, e_2, \ldots, e_n$ →

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R}$ = { P, $(u_1, u_2, \ldots. u_n)$: ∃ $(a_1, a_2, \ldots. a_n)$ s.t. $u_i = \Pi\, g^{a}$, and $P(a_1, a_2, \ldots. a_n)$ = true}

P, $(u_1, u_2, \ldots. u_n)$

P, $(u_1, u_2, \ldots. u_n)$

$d \leftarrow Z_p$, $(u'_i = \Pi\, g \quad)$

$(u'_1, u'_2, \ldots. u'_n)$ →

c ←

$c \leftarrow Z_p$

$e_j = d + a_j.c$

$e_1, e_2, \ldots, e_n$ →

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R}$ = { P, $(u_1, u_2, \ldots . u_n)$: ∃ $(a_1, a_2, \ldots . a_n)$ s.t. $u_i = \Pi g^{a}$, and P$(a_1, a_2, \ldots . a_n)$ = true}

P, $(u_1, u_2, \ldots . u_n)$

P, $(u_1, u_2, \ldots . u_n)$

$d \leftarrow Z_p$, $(u'_i = \Pi g \;\;)$

$(u'_1, u'_2, \ldots . u'_n)$ →

c ←

$c \leftarrow Z_p$

$e_j = d + a_j.c$

$e_1, e_2, \ldots, e_n$ →

Equality of Ciphertexts

Recall: El-Gamal Encryption

PK = (g, h), SK = a s.t. $g^a = h$, $Enc_{PK}(m; r) = g^r, h^r.m$

Equality of Ciphertexts

Recall: El-Gamal Encryption

PK = (g, h), *SK* = a s.t. $g^a = h$, $Enc_{PK}(m; r) = g^r, h^r.m$

$PK1 = (g, h_1)$, $PK2 = (g, h_2)$

$ct_1 = Enc_{PK1}(m; r_1) = (g^{r1}, h_1^{r1}.m)$
$ct_2 = Enc_{PK2}(m; r_2) = (g^{r2}, h_2^{r2}.m)$

ct_1 , ct_2 and a proof that
both encrypt the same message

Equality of Ciphertexts

Recall: El-Gamal Encryption

PK = (g, h), *SK* = a s.t. $g^a = h$, $Enc_{PK}(m; r) = g^r, h^r.m$

PK1 = (g, h_1), *PK2* = (g, h_2)

$ct_1 = Enc_{PK1}(m; r_1) = (g^{r1}, h_1^{r1}.m) = (p_1, q_1)$

$ct_2 = Enc_{PK2}(m; r_2) = (g^{r2}, h_2^{r2}.m) = (p_2, q_2)$

ct_1 , ct_2 and a proof that
both encrypt the same message

There exist r1, r2 such that:
$p_1 = g^{r1}$, $p_2 = g^{r2}$, $q_1/q_2 = h_1^{r1}/h_2^{r2}$

General Linear Relations on Exponents

Prover **Verifier**

NP relation $\mathcal{R}$ = { P, $(u_1, u_2, \ldots .u_n)$: ∃ $(a_1, a_2, \ldots .a_n)$ s.t. $u_i = \Pi g^a$, and P$(a_1, a_2, \ldots .a_n)$ = true}

P, $(u_1, u_2, \ldots .u_n)$

d ← Z_p, $(u'_i = \Pi g\ \ )$

$(u'_1, u'_2, \ldots .u'_n)$ →

c ←

$e_j = d + a_j.c$

$e_1, e_2, \ldots, e_n$ →

P, $(u_1, u_2, \ldots .u_n)$

c ← Z_p