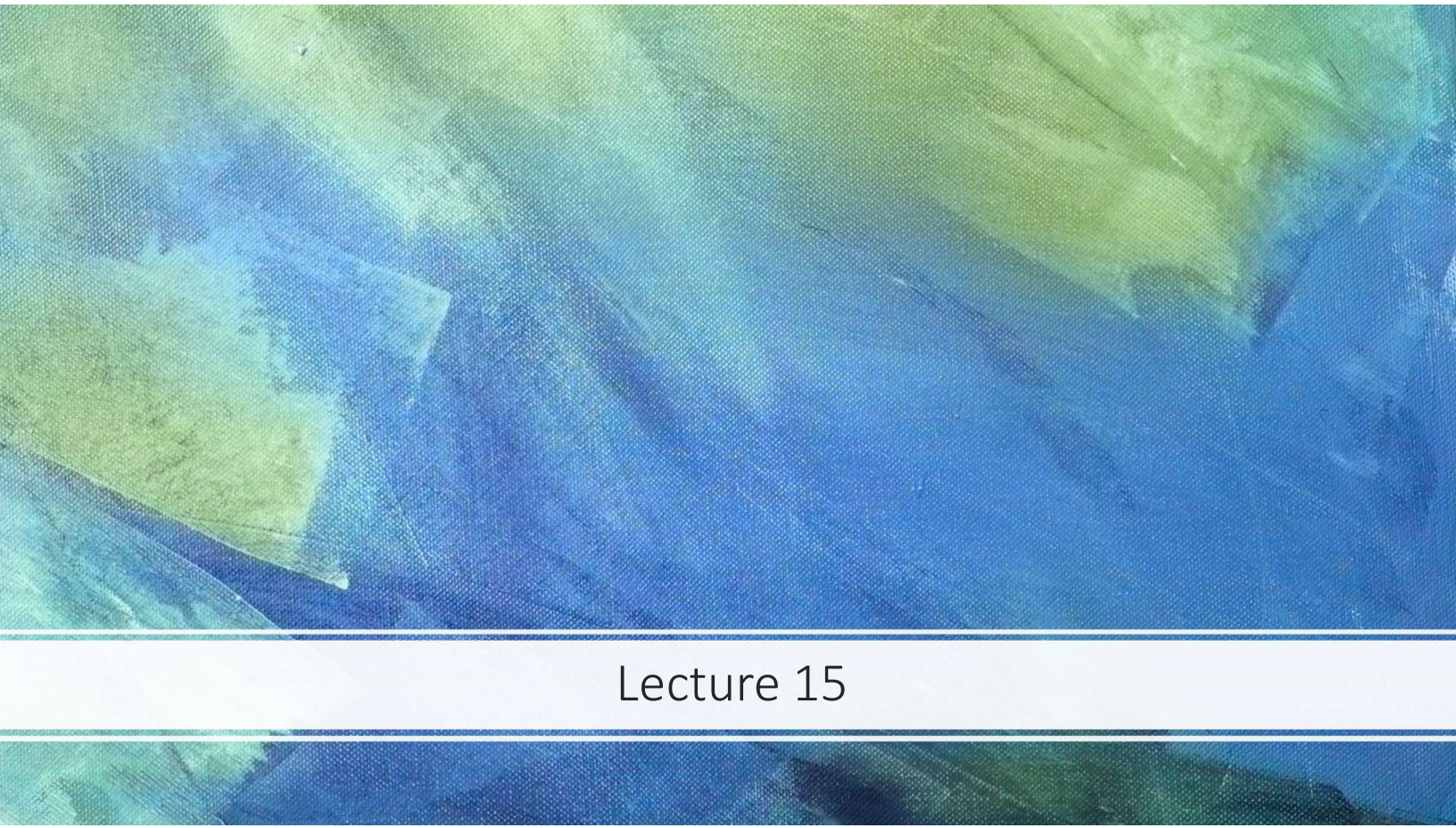
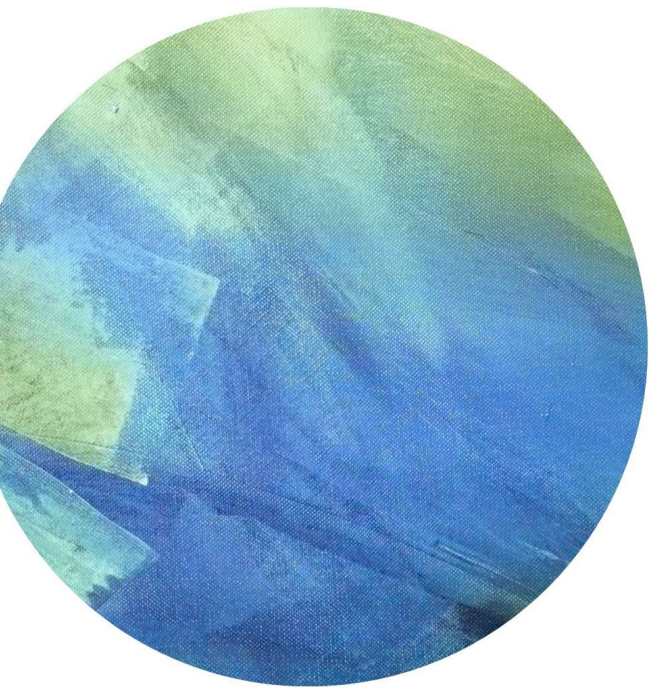

Lecture 15



Outline

Zero Knowledge

Proofs on Encrypted Data

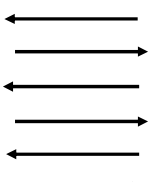
Zero-Knowledge

Real World

Prover

NP Statement **x**

Witness that **x** is true



Verifier

Didn't learn witness

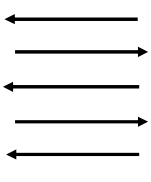
Outputs view

Ideal World (Proof)

Simulator

NP Statement **x**

No witness



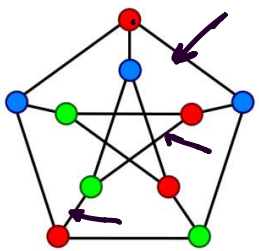
Verifier

Didn't learn witness

Outputs similar view

Cannot distinguish the two

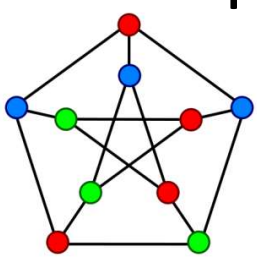
3-coloring

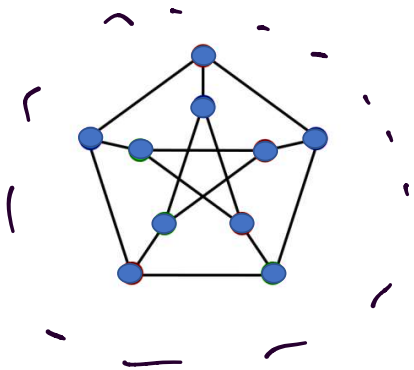


Color all vertices with only three colors (R, G, B) such that no edge should connect two vertices of the same color.

3-coloring

Prover



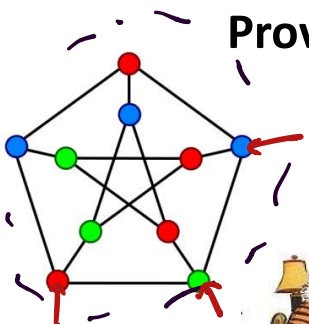


Verifier

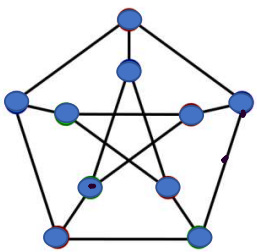
If V checks too few: P has noticeable prob. of cheating

If V checks too many: leaks inf. about colors

3-coloring



Prover



Verifier

$c_1, c_2 \ldots\ldots c_n$

$\forall i \in [v]$, com(i, color_i) →

← random (j,k) s.t. vertices (j,k) connected by an edge

R → B

B → G

G → R

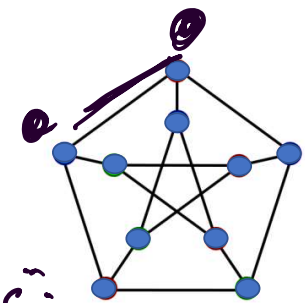
Decommit c_j and c_k →

check that vertices j and k had diff. colors

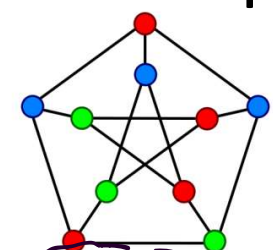
Soundness: in 3 message subprotocol

$\Pr[V \text{ accepts} \mid \text{Graph is not 3-colorable}] \geq 1 - \frac{1}{\#\text{edges}}$

3-coloring



Prover



R B G
color 2 color 1 color 3

$c_i = com(i, color_i)$

edge = (j,k)

opens c_j, c_k

repeat many times

K B G
color 3 color 1 color 2

$c'_i = com[i, color_i)$

edge = (j', k')

open $c_{j'}, c_{k'}$

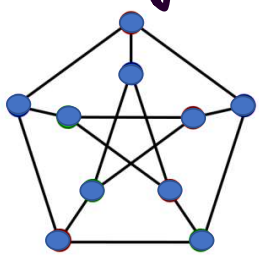
Verifier

Soundness in n-fold repetitions:

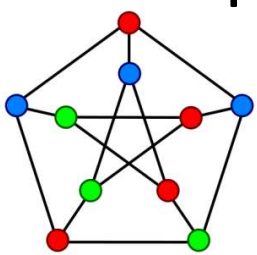
$\Pr[V \text{ accepts ALL } n \text{ repetitions} \mid G \text{ is not 3-colorable}] \leq \left(1 - \frac{1}{\#\text{edges}}\right)^n$.

Set $n = (\#\text{edges})^2$ to get negligible

3-coloring



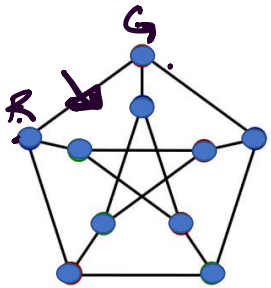
Prover



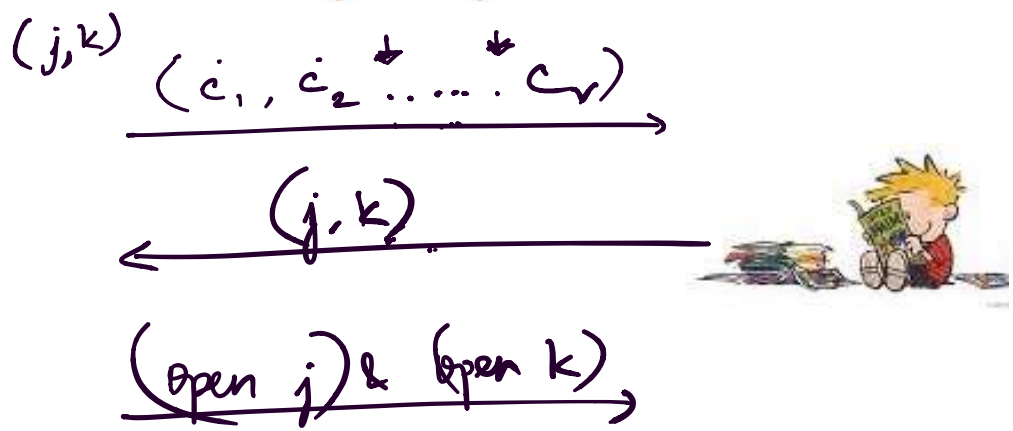
Verifier

3-coloring

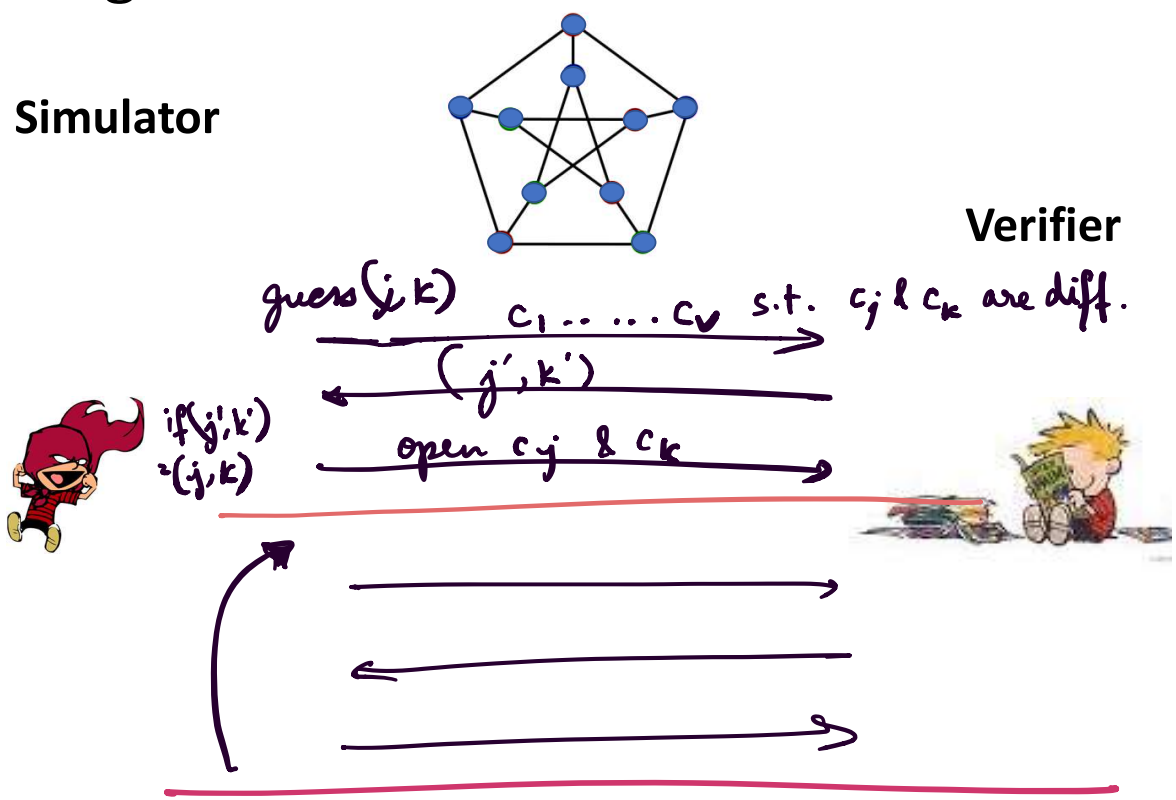
Simulator



Verifier

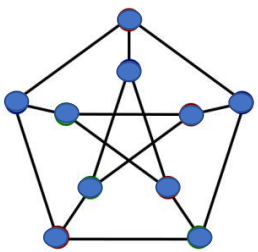


3-coloring


Simulator
Verifier
guess (j,k)
$c_1 \ldots \ldots c_v$ s.t. c_j & c_k are diff.
(j',k')
if (j',k') = (j,k)
open c_j & c_k

3-coloring

Simulator

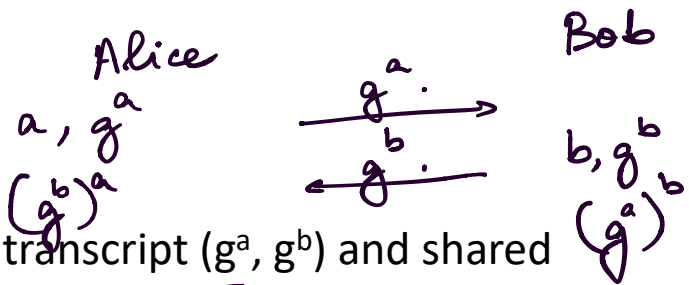


Verifier

Proofs on Encrypted Data

Chaum-Pedersen

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



- Recall that an El-Gamal encryption has transcript (g^a, g^b) and shared secret key g^{ab} is used to hide the message
- Suppose I wanted to prove correctness of ciphertexts
- Lets simplify the problem

Chaum-Pedersen

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

- Proof of a given triple being of the following form

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

  $(\log_g u) \cdot (\log_g v) = (\log_g w)$

  $a \quad b \quad ab$

- 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)

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

Chaum-Pedersen

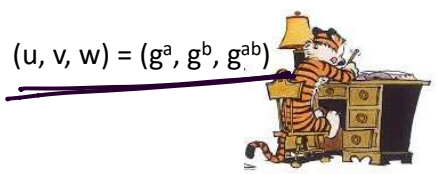
Prover

Verifier

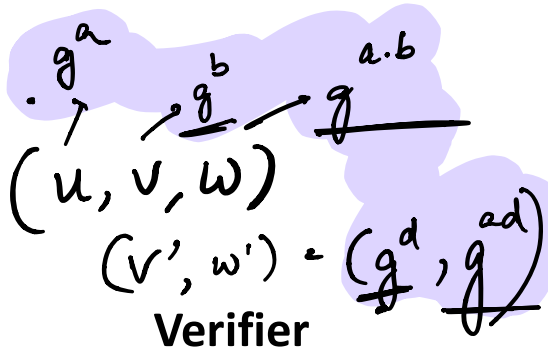
(u, v, w)

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

(v', w')



Chaum-Pedersen

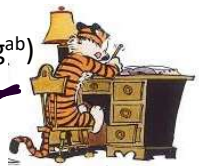


Prover

(u, v, w)

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

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



$e = d + b.c$

$g^d \; g^{ad} = u^d$

(v', w')

c

$e = (d + b.c)$

e'

Verifier

$c \leftarrow Z_p$

Check 1: $g^{bc} \cdot g^d = g^e$

Check 2: $(g^{abc})^a \cdot g^{ad} = g^{ae}$

Check 1: $g^e = g^d (g^b)^c$

$= v' \cdot (v)^c$

guarantees that $e = (d + b.c)$

Check 2: $u^e = (u^d)(u^b)^c$

$= w' \cdot (w)^c$

Suppose $(u, v, w) = g^a, g^b, g^z$ for $z \neq ab$.

$u^{e'} = w' \cdot w^c$

$g^{de'} = g^{ad} \cdot g^{ze}$ $u^e \neq w' \cdot w^c$ because $w \neq u^b$

$\neq u^b$

Chaum-Pedersen

Prover **Verifier**

(u, v, w)

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

<—— c —— $c \leftarrow Z_p$

$e = d + b.c$ —— e ——>

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

$g^e = (v' \cdot v^c)$

$u^e = (w' \cdot w^c)$

Chaum-Pedersen: Zero-Knowledge

Honest Prover's messages would've looked like the following:

Simulator **Verifier**

(u, v, w), guess c

$v' = g^e \cdot (v^c)^{-1}$, $w' = u^e \cdot (w^c)^{-1}$

c

e

(u, v, ... a, b)

* Guess c
* Pick e
* Pick $v' = g^e / v^c$
* Pick $w' = u^e / w^c$

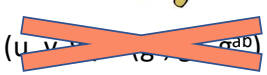
$g^e = (v' \cdot v^c)$, $u^e = w' \cdot (w^c)$

Chaum-Pedersen: Zero-Knowledge

Simulator **Verifier**

(u, v, w), guess c

e ← Z_p, ($v' = g^e/v^c$, $w' = u^e/w^c$) $\xrightarrow{(v', w')}$



Chaum-Pedersen: Zero-Knowledge

Simulator **Verifier**

(u, v, w), guess c

e ← Z_p, (v' = g^e/v^c, w' = u^e/w^c)

(v', w') →

← c

Chaum-Pedersen: Zero-Knowledge

Simulator **Verifier**

(u, v, w), guess c

e ← Z_p, (v' = g^e/v^c, w' = u^e/w^c)

(v', w')

c

e

Chaum-Pedersen: Soundness

Prover **Verifier**

(u, v, w)

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

(v', w') →

$c \leftarrow Z_p$

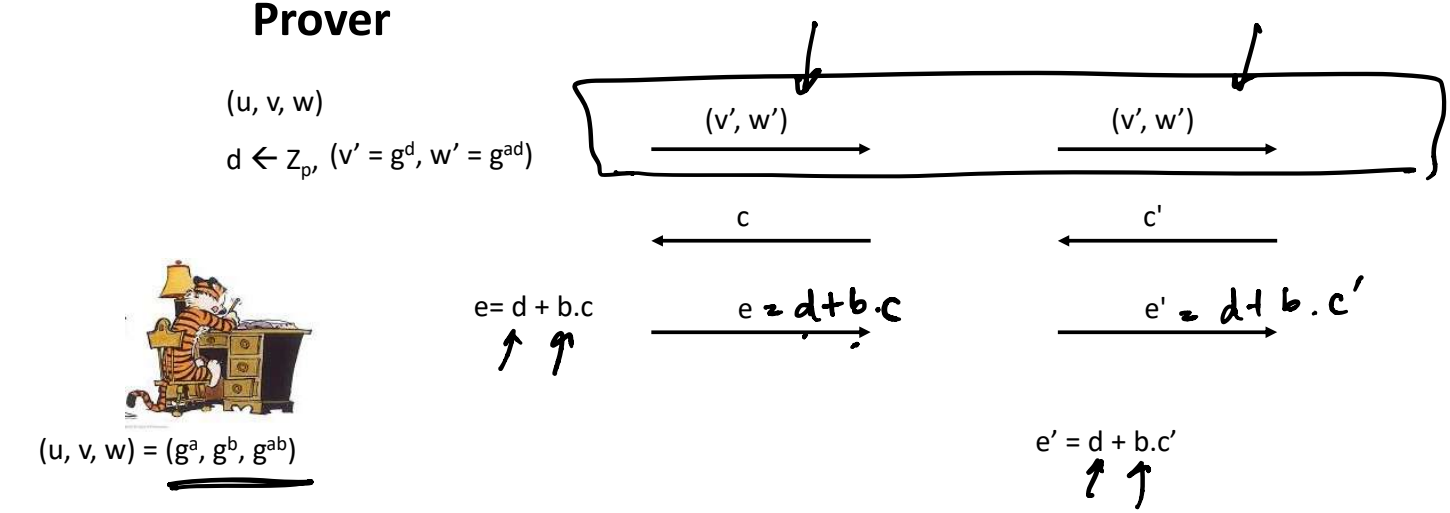
← c

$e = d + b.c$

e →

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

Chaum-Pedersen: Soundness



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. $u_i = \prod_j g_{ij}^{x_j}$, and $P(a_1, a_2, \ldots a_n)$ = true}

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

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

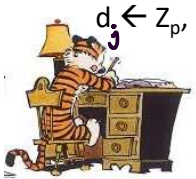
General Linear Relations on Exponents

Prover **Verifier**

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

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

$d_j \leftarrow Z_p$, $(u'_i = \prod_j g_{ij}^{d_j})$



$(u'_1, u'_2, \ldots u'_n)$

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

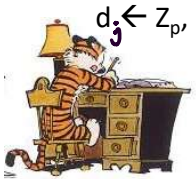
General Linear Relations on Exponents

Prover **Verifier**

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

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

$d_j \leftarrow Z_p$, $(u'_i = \prod_j g_{ij}^{d_j})$



$(u'_1, u'_2, \ldots u'_n)$

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. $u_i = \prod g_{ij}^{z_{ij}}$, and $P(a_1, a_2, \ldots a_n)$ = true}

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

$d_j \leftarrow Z_p$, $(u'_i = \prod g_{ij}^{d_j})$

$(u'_1, u'_2, \ldots u'_n)$

c

$e_j = d_j + a_j.c$

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

$c \leftarrow Z_p$

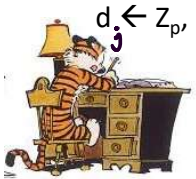
General Linear Relations on Exponents

Prover **Verifier**

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

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

$d_j \leftarrow Z_p$, $(u'_i = \prod g_{ij}^{d_j})$



$(u'_1, u'_2, \ldots u'_n)$

c

$e_1, e_2, \ldots, e_n$

$c \leftarrow Z_p$

$e_j = d_j + a_j.c$

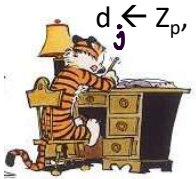
General Linear Relations on Exponents

Prover **Verifier**

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

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

$d_j \leftarrow Z_p$, $(u'_i = \prod g_{ij}^{d_j})$



$(u'_1, u'_2, \ldots u'_n)$

c

$e_1, e_2, \ldots, e_n$

$c \leftarrow Z_p$

$e_j = d_j + a_j.c$

$\forall i \in [n], \prod g_{ij}^{e_j} \stackrel{?}{=} u'_i \cdot u_i^c$