

### 5.1 Analysis

1. Completeness
2. Soundness
3. Zero-Knowledge

#### 5.1.1 Completeness

If  $G_0 \cong G_1$  then  $V$  always accepts

#### 5.1.2 Soundness

If  $G_0 \not\cong G_1$  then  $V$  rejects with probability at least  $\frac{1}{2}$ . (But cheating prover may also succeed with probability at most  $\frac{1}{2}$ . This is called the *soundness error* of the scheme. To reduce it to  $2^{-k}$ , then repeat the protocol  $k$  times.)

#### 5.1.3 Intuition here

- If  $P$  could answer both challenges, then statement must be true!
- If model both as machines (Turing Machines or VMs), then one can take a snapshot of both after  $P$  has sent first message. Then run protocol to completion; Then restart protocol from snapshot and run it until  $V$  picks a different challenge (b).

This gives  $V$  both permutations  $\rho_0$  and  $\rho_1$

$$G_0 \stackrel{\cong}{=} \rho_0 \mathbb{H} \text{ and } G_1 \stackrel{\cong}{=} \rho_1 \mathbb{H}$$

so,  $V$  could extract the isomorphism between  $G_0$  and  $G_1$

- Gedankenexperiment

#### 5.1.4 Zero-Knowledge

What is "computational knowledge"?

If a party can generate a random variable  $T$  with exactly the same (or an indistinguishable) distribution, then this party gains no (useful) information from  $T$ .

Here,  $V$  can generate (simulate) a transcript  $T$  of an accepting protocol execution.

same distribution  $\Leftrightarrow$  perfect zero knowledge

indistinguishable distribution  $\Leftrightarrow$  computational zero knowledge.

$V$  can simulate transcript  $T$ :

1.  $\tilde{b} \leftarrow \{0, 1\}$
2.  $\tilde{\rho} \leftarrow$  permutation of  $V$
3.  $\tilde{H} = \tilde{\rho}(G_{\tilde{b}})$ , i.e. isomorph to  $G_{\tilde{b}}$

Distribution  $(H, b, \rho)$  in real protocol same as  $(\tilde{H}, \tilde{b}, \tilde{\rho})$  in simulated execution.

$\Rightarrow V$  leaves no information through ZKP

$\Rightarrow V$  cannot transfer this to any third party

#### 5.1.5 What can be proved "in zero-knowledge"?

- GI problem  $\notin P$   
GI is believed to be between  $P$  and  $NP$  (like factoring, or DL)
- If one  $NP$ -complete problem has a ZKP, then any problem in  $NP$  has a ZKP (polynomial time)  
3-Colorability of a graph  $G$  is  $NP$ -complete and has ZKP

### 5.1.6 Can one use these protocols for online authentication?

In principle yes, BUT in practice more efficient schemes exist.

## 5.2 Zero-Knowledge Proofs of Knowledge (ZKPK)

Want to prove knowledge about secrets  $\alpha, \beta, \gamma, \dots$  such that  $\Psi(\alpha, \beta, \gamma, \dots)$  holds

### Example: Prover $\mathbb{P}$

Prover  $\mathbb{P}$  knows that it knows  $\alpha$  s.t.  $g^\alpha = y$ .

### Notation

$PK\{(\alpha, \beta, \gamma, \dots) : \Psi(\alpha, \beta, \gamma, \dots)\}$ , where  $\alpha, \beta$  are known to  $\mathbb{P}$  e.g.  $PR\{(\alpha) : y = g^\alpha\}$

### 5.2.1 Formalizing ZKPK (3-Move Protocol or $\Sigma$ -Protocol)

To convince  $\mathbb{V}$ ,  $\mathbb{P}$  should demonstrate that it knows such a secret  $(\alpha, \beta, \gamma, \dots)$  s.t.  $\Psi(\alpha, \beta, \gamma, \dots)$ .  
Formalized using *extraction* of the secret  $\alpha, \beta, \dots$  from  $\mathbb{P}$ . Using an extractor  $\mathbb{E}$ , an efficient algorithm that extracts secrets  $\alpha, \beta, \dots$  from  $\mathbb{P}$  when given two protocol runs (transcripts) with same commitment

### 5.2.2 Definition

A zero-knowledge proof-of-knowledge (ZKPK) is a 3-Move protocol for a relation  $\Psi$  satisfies

#### Completeness

If  $\mathbb{P}$  has input  $x$  s.t.  $\Psi(x)$  then  $\mathbb{V}$  accepts.

#### Soundness

There is an efficient knowledge extractor  $\mathbb{E}$  s.t.  $\mathbb{E}((t, c, s), (t, c', s')) \rightarrow x$  when  $c \neq c'$  and  $\Psi(x)$  (both transcripts are from executions where  $\mathbb{V}$  accepts).

#### Zero-Knowledge

$\mathbb{V}$  can simulate transcripts  $(t, c, s)$  on its own with same (or indistinguishable) distribution

$\Leftrightarrow$

$\exists$  simulator  $\mathbb{S}$  that produces  $(t, c, s)$ ... but may use different order

### 5.2.3 ZKPK of a Discrete Logarithm ("Schnorr Proof")

Again,  $G = \langle g \rangle$  of order  $q$ .

$$\Psi(x) : g^x = y$$

#### Completeness?

If  $\mathbb{P}$  and  $\mathbb{V}$  honest then:

$$t = g^r = g^{r-cx+cx} = g^s \cdot g^{x \cdot c} = g^s \cdot y^c$$

Thus  $\mathbb{V}$  accepts.

## Soundness?

Two executions with  $(t, c, s)$  and  $(t, c', s')$  (Note  $c \neq c'$ ):

$$\begin{aligned} \Rightarrow t &= g^s \cdot y^c = g^{s'} \cdot y^{c'} \\ \Leftrightarrow g^{s-s'} &= y^{c'-c} = g^{x \cdot (c'-c)} \\ \Leftrightarrow s - s' &\equiv x(c' - c) \pmod{q} \\ \Leftrightarrow x &\equiv \frac{s - s'}{c' - c} \pmod{q} \end{aligned}$$

This  $x$  satisfies  $g^x = y$ .

## Zero-Knowledge?

$\forall$  chooses triples  $(t, c, s)$  on its own:

$$\begin{aligned} c &\leftarrow \mathbb{Z}_q \\ s &\leftarrow \mathbb{Z}_q \\ t &\leftarrow g^s \cdot y^c \quad (\text{in } G) \end{aligned}$$

A triple  $(t, c, s)$  has same distribution as a transcript of an accepting execution

## 5.3 Commitment Schemes

- How to pick a uniformly random bit among two parties s.t. no single party can bias this bit
- Cryptographic primitive for a sender  $\mathbb{S}$  and a receiver  $\mathbb{R}$

### 5.3.1 Definition

A commitment scheme has 3 algorithms: KEYGEN(), COM(), VER().

1. KEYGEN()  $\rightarrow pk$   
probabilistic
2. COM(PK, X, R)  $\rightarrow c$   
deterministic  
outputs commitment  $c \in \{0, 1\}^*$   
 $x \in \{0, 1\}^*$   
 $r \in R$ , randomness, chosen  $r \leftarrow R$  by  $\mathbb{S}$
3. VER(PK, X, R, C)  $\rightarrow \text{TRUE/FALSE}$   
deterministic, outputs boolean indicating whether  $x$  and  $r$  correctly "open" commitment  $c$   
Run by receiver  $\mathbb{R}$

## Completeness

...

## Binding

...

## Hiding

...