

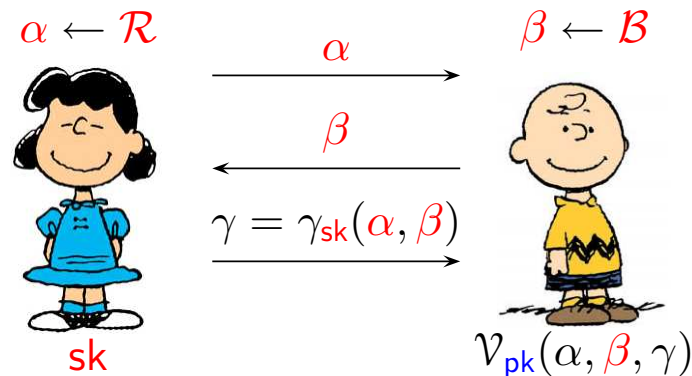
MTAT.07.003 CRYPTOLOGY II

## **Sigma Protocols**

Sven Laur  
University of Tartu

# Formal Syntax

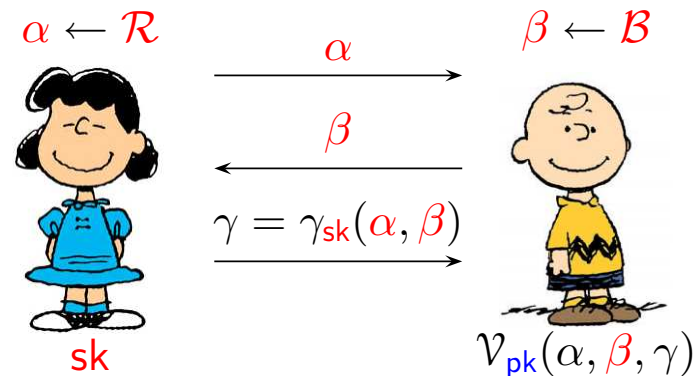
# Sigma protocols



A *sigma protocol* for an efficiently computable relation  $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$  is a three move protocol that satisfies the following properties.

- ▷  **$\Sigma$ -structure.** A prover first sends a commitment, next a verifier sends *varying* challenge, and then the prover must give a consistent response.
- ▷ **Functionality.** The protocol run between an honest prover  $\mathcal{P}(sk)$  and verifier  $\mathcal{V}(pk)$  is always accepting if  $(sk, pk) \in R$ .

## Security properties of sigma protocols



- ▷ **Perfect simulatability.** There exists an efficient *non-rewinding* simulator  $\mathcal{S}$  such that the output distribution of a semi-honest verifier  $\mathcal{V}_*$  in the real world and the output distribution of  $\mathcal{S}^{\mathcal{V}_*}$  in the ideal world coincide.
- ▷ **Special soundness.** There exists an efficient extraction algorithm  $\text{Extr}$  that, given two accepting protocol runs  $(\alpha, \beta_0, \gamma_0)$  and  $(\alpha, \beta_1, \gamma_1)$  with  $\beta_0 \neq \beta_1$  that correspond to  $pk$ , outputs  $sk_*$  such that  $(sk_*, pk) \in R$

# Soundness of Sigma Protocols

## Soundness in the standalone model

**Main Theorem.** Denote  $\kappa = |\mathcal{B}|^{-1}$ . Now if a  $t$ -time prover  $\mathcal{P}_*$  succeeds in the sigma protocol with probability at least  $\varepsilon > \kappa$ , there exists a knowledge-extraction algorithm  $\mathcal{K}^{\mathcal{P}_*}$  that always recovers a secret  $sk_*$  corresponding to  $pk$  and the expected running-time of  $\mathcal{K}^{\mathcal{P}_*}$  is

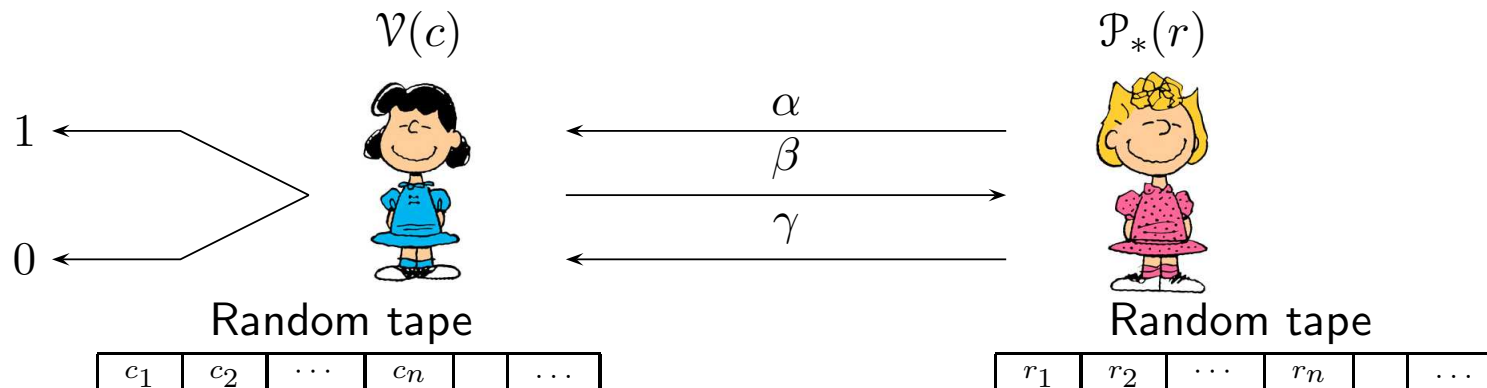
$$c_1 \cdot \frac{2}{\varepsilon - \kappa} + c_2$$

for some small constants  $c_1, c_2 \in \mathbb{R}$ .

### Remark.

- ▷ The coefficient  $c_1$  depends on the total complexity of the protocol.
- ▷ The coefficient  $c_2$  depends on the complexity of the Extr algorithm.

## The corresponding matrix game



Let  $A(r, c)$  be the output of the honest verifier  $\mathcal{V}(c)$  that interacts with a potentially malicious prover  $\mathcal{P}_*(r)$ .

- ▷ Then all matrix elements in the same row  $A(r, \cdot)$  lead to same  $\alpha$  value.
- ▷ To extract the secret key **sk**, we must find two ones in the same row.
- ▷ We can compute the entries of the matrix on the fly.

## Classical algorithm

**Task:** Find two ones in a same row.

Rewind:

1. Probe random entries  $A(r, c)$  until  $A(r, c) = 1$ .
2. Store the matrix location  $(r, c)$ .
3. Probe random entries  $A(r, \bar{c})$  in the same row until  $A(r, \bar{c}) = 1$ .
4. Output the location triple  $(r, c, \bar{c})$ .

Rewind-Exp:

1. Repeat the procedure Rewind until  $c \neq \bar{c}$ .
2. Use the extraction algorithm Extr to extract **sk**.



## Average-case running time

**Theorem.** If a  $m \times n$  zero-one matrix  $A$  contains  $\varepsilon$ -fraction of nonzero entries, then the Rewind and Rewind-Exp algorithm make on average

$$\mathbf{E}[\text{probes}|\text{Rewind}] = \frac{2}{\varepsilon}$$

$$\mathbf{E}[\text{probes}|\text{Rewind-Exp}] = \frac{2}{\varepsilon - \kappa}$$

probes where  $\kappa = \frac{1}{n}$  is a *knowledge error*.

## Average case complexity I

Assume that the matrix contains  $\varepsilon$ -fraction of nonzero elements, i.e.,  $\mathcal{P}_*$  convinces  $\mathcal{V}$  with probability  $\varepsilon$ . Then on average we make

$$\mathbf{E}[\text{probes}_1] = \varepsilon + 2(1 - \varepsilon)\varepsilon + 3(1 - \varepsilon)^2\varepsilon + \dots = \frac{1}{\varepsilon}$$

matrix probes to find the first non-zero entry. Analogously, we make

$$\mathbf{E}[\text{probes}_2|r] = \frac{1}{\varepsilon_r}$$

probes to find the second non-zero entry. Also, note that

$$\mathbf{E}[\text{probes}_2] = \sum_r \Pr[r] \cdot \mathbf{E}[\text{probes}_2|r] = \sum_r \frac{\varepsilon_r}{\sum_{r'} \varepsilon_{r'}} \cdot \frac{1}{\varepsilon_r} = \frac{1}{\varepsilon},$$

where  $\varepsilon_r$  is the fraction of non-zero entries in the  $r^{\text{th}}$  row.

## Average case complexity II

As a result we obtain that the Rewind algorithm does on average

$$\mathbf{E}[\text{probes}] = \frac{2}{\varepsilon}$$

probes. Since the Rewind algorithm fails with probability

$$\Pr[\text{failure}] = \sum_r \Pr[c = \bar{c} | \text{halting}] \leq \frac{\kappa}{\varepsilon}$$

we make on average

$$\mathbf{E}[\text{probes}^*] = \frac{1}{\Pr[\text{success}]} \cdot \mathbf{E}[\text{probes}] \leq \frac{\varepsilon}{\varepsilon - \kappa} \cdot \frac{2}{\varepsilon} = \frac{2}{\varepsilon - \kappa}$$

probes.

## Soundness of sequential compositions

**Main Theorem.** Consider a setting where a prover  $\mathcal{P}_*$  and honest verifier  $\mathcal{V}$  sequentially execute the same sigma protocol  $\ell$  times. Denote  $\kappa = |\mathcal{B}|^{-1}$ . Also let  $\mathcal{P}_*$  be successful if  $\mathcal{P}_*$  succeeds at least in one protocol instance. Now if a  $t$ -time prover  $\mathcal{P}_*$  succeeds with probability at least  $\varepsilon > \ell\kappa$ , there exists a knowledge-extraction algorithm  $\mathcal{K}^{\mathcal{P}_*}$  that always recovers a secret  $sk_*$  corresponding to  $pk$  and the expected running-time of  $\mathcal{K}^{\mathcal{P}_*}$  is

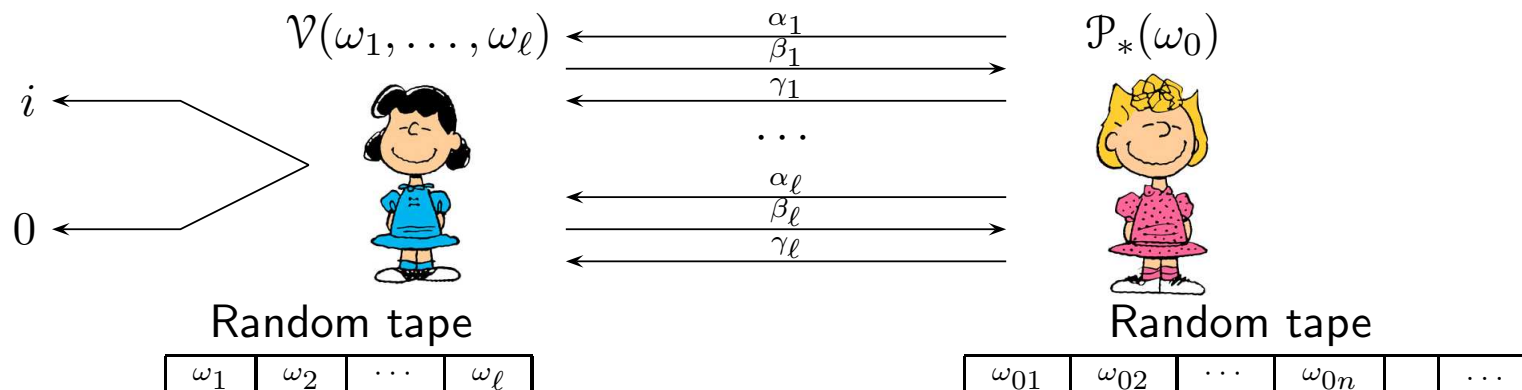
$$c_1 \cdot \frac{\ell + 1}{\varepsilon - \ell\kappa} + c_2$$

for some small constants  $c_1, c_2 \in \mathbb{R}$ .

### Remark.

- ▷ The coefficient  $c_1$  depends on the total complexity of the protocol.
- ▷ The coefficient  $c_2$  depends on the complexity of the Extr algorithm.

## The corresponding matrix game



Let  $A(\omega_0, \omega_1, \dots, \omega_\ell)$  denote the index of the first successful protocol between honest verifier  $\mathcal{V}(\omega_1, \dots, \omega_\ell)$  and a prover  $\mathcal{P}_*(\omega_0)$ .

- ▷ Then a randomness prefix  $\omega_0, \dots, \omega_{i-1}$  leads to the same  $\alpha_i$  value.
- ▷ To extract the secret key **sk**, we must find two  $i$ -s with the same prefix.
- ▷ We can compute the entries of the array on the fly.

## Classical algorithm

### Rewind:

1. Probe random entries  $A(\omega)$  until  $A(\omega) \neq 0$ .
2. Store the matrix location  $\omega$  and the rewinding point  $i \leftarrow A(\omega)$ .
3. Probe random entries  $A(\bar{\omega})$  with the prefix  $\omega_0, \dots, \omega_{i-1}$  until  $A(\bar{\omega}) = i$ .
4. Output the location tuple  $(\omega, \bar{\omega})$ .

### Rewind-Exp:

1. Repeat the procedure Rewind until  $\omega_i \neq \bar{\omega}_i$ .
2. Use the extraction algorithm Extr to extract **sk**.

## Average-case running time

**Theorem.** If a array  $A(\omega)$  with entries in  $\{0, \dots, \ell\}$  contains  $\varepsilon$ -fraction of nonzero entries, then Rewind and Rewind-Exp make on average

$$\mathbf{E}[\text{probes}|\text{Rewind}] = \frac{2}{\varepsilon}$$

$$\mathbf{E}[\text{probes}|\text{Rewind-Exp}] = \frac{\ell + 1}{\varepsilon - \kappa}$$

probes where the *knowledge error*

$$\kappa = \sum_{i=1}^{\ell} \Pr [\omega_i = \bar{\omega}_i] .$$

## Average case complexity I

Assume that  $\mathcal{A}$  succeeds with probability  $\varepsilon$ . Then the results proved for the zero-one matrix with fixed dimensions imply

$$\mathbf{E}[\text{probes}_1] = \frac{1}{\varepsilon} \quad \text{and} \quad \mathbf{E}[\text{probes}_2 | A(\omega) = i] = \frac{1}{\varepsilon_i}$$

where  $\varepsilon_i$  is the fraction of entries labelled with  $i$ . Thus

$$\mathbf{E}[\text{probes}_2] = \sum_{i=1}^{\ell} \Pr [A(\omega) = i] \cdot \mathbf{E}[\text{probes}_2 | A(\omega) = i]$$

$$\mathbf{E}[\text{probes}_2] = \sum_{i=1}^{\ell} \frac{\varepsilon_i}{\varepsilon} \cdot \frac{1}{\varepsilon_i} = \frac{\ell}{\varepsilon} .$$



## Average case complexity II

Consequently, the Rewind algorithm does on average

$$\mathbf{E}[\text{probes}] = \frac{\ell + 1}{\varepsilon}$$

probes. Since the Rewind algorithm fails with probability

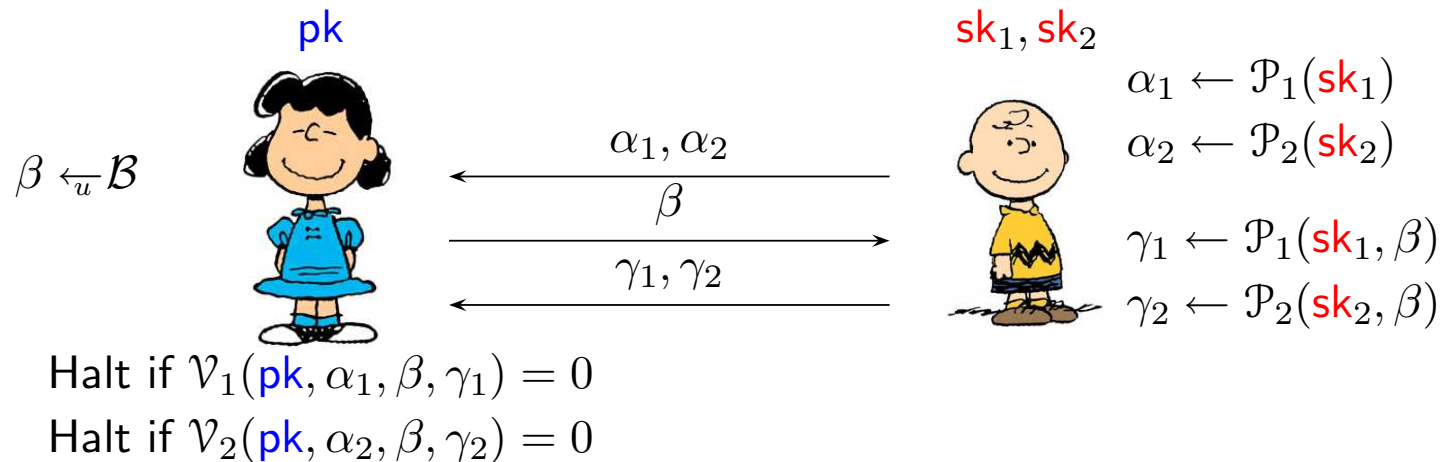
$$\Pr[\text{failure}] = \sum_{i=1}^{\ell} \Pr[A(\omega) = i] \Pr[\omega_i = \bar{\omega}_i | \text{halting}] \leq \frac{\kappa_1 + \dots + \kappa_\ell}{\varepsilon}$$

we make on average

$$\mathbf{E}[\text{probes}^*] = \frac{1}{\Pr[\text{success}]} \cdot \mathbf{E}[\text{probes}] \leq \frac{\varepsilon}{\varepsilon - \kappa} \cdot \frac{\ell + 1}{\varepsilon} = \frac{\ell + 1}{\varepsilon - \kappa} .$$

# Various Parallel Compositions

## Conjunctive proofs



If we run two sigma protocols for different relations  $R_1$  and  $R_2$  in parallel, we get a sigma protocol for new relation  $R_1 \wedge R_2$

$$(sk_1, sk_2, pk) \in R_1 \wedge R_2 \iff (sk_1, pk) \in R_1 \wedge (sk_2, pk) \in R_2 .$$

provided that both sigma protocols have the same challenge space  $\mathcal{B}$  and it a *perfect simulation of transcripts*  $(\alpha_i, \beta, \gamma_i)$  is *efficient* for any  $\beta$ .

## The corresponding proof

**Perfect simulatability.** Let  $\mathcal{S}_1$  be a canonical simulator for  $\mathcal{V}_1$ . Now if  $\mathcal{S}_1$  outputs a properly distributed triple  $(\alpha_1, \beta, \gamma_1)$ , then we can augment this with properly distributed  $(\alpha_2, \beta, \gamma_2)$  and thus we get a properly distributed protocol transcript  $(\alpha_1, \alpha_2, \beta, \gamma_1, \gamma_2)$ .

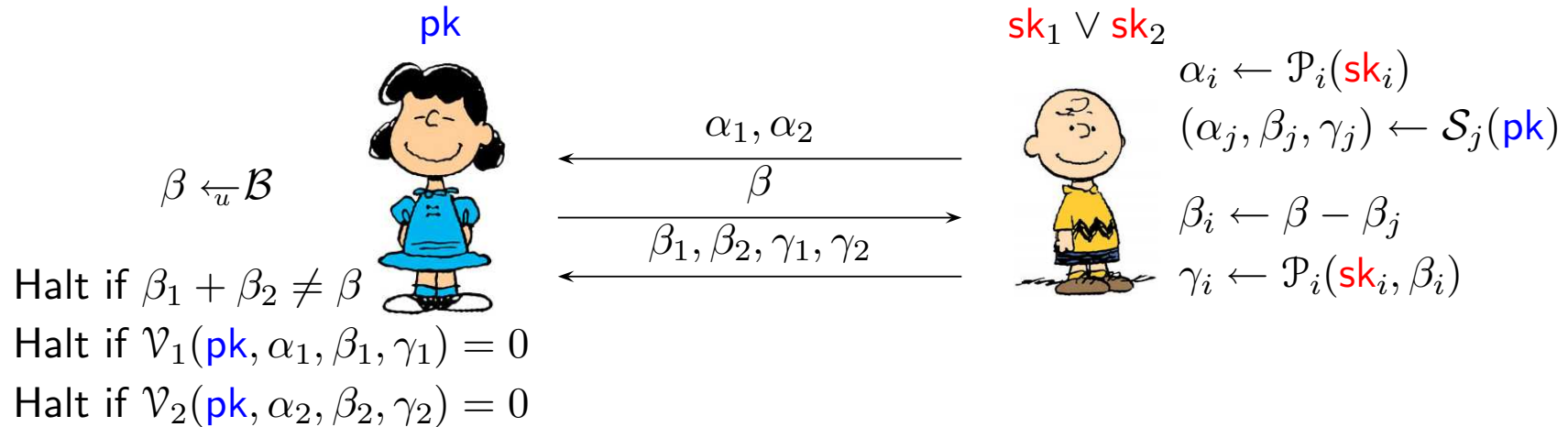
**Special soundness.** Given two accepting transcripts

$$(\alpha_1, \alpha_2, \beta^0, \gamma_1^0, \gamma_2^0), (\alpha_1, \alpha_2, \beta^1, \beta_2^1, \gamma_1^1, \gamma_2^1), \quad \text{with } \beta^0 \neq \beta^1 ,$$

we can decompose them into original colliding transcripts

$$\begin{aligned} &(\alpha_1, \beta^0, \gamma_1^0), (\alpha_1, \beta^1, \gamma_1^1), & \beta^0 \neq \beta^1 , \\ &(\alpha_2, \beta^0, \gamma_2^0), (\alpha_2, \beta^1, \gamma_2^1), & \beta^0 \neq \beta^1 . \end{aligned}$$

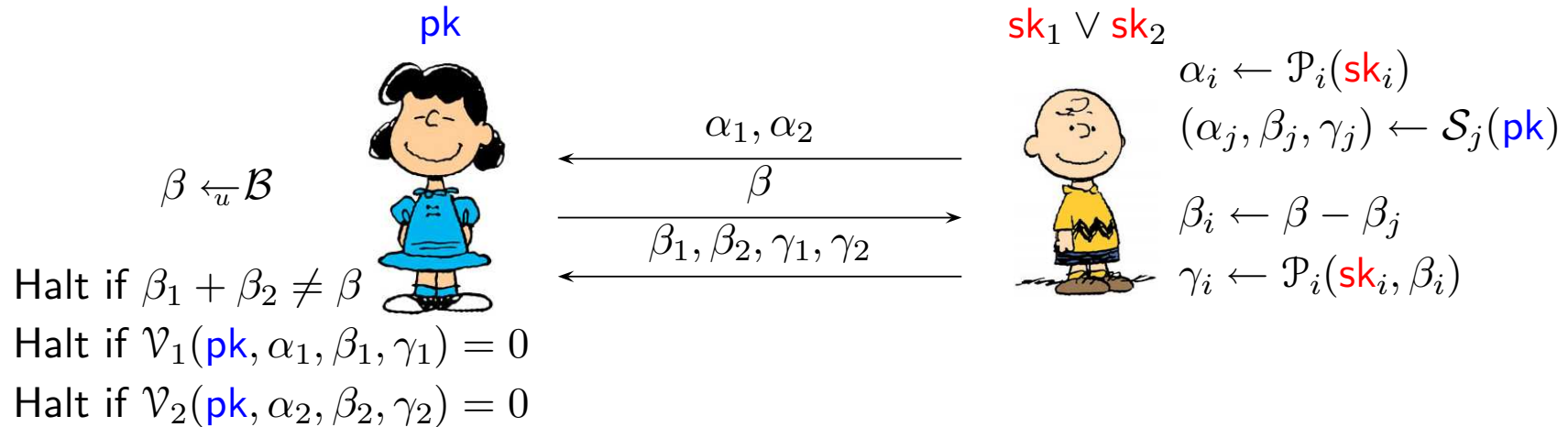
## Disjunctive proofs



Assume that we have two sigma protocols for relations  $R_1$  and  $R_2$  such that the challenge is chosen uniformly from a commutative group  $(\mathcal{B}; +)$ .

Then a prover can use a simulator  $\mathcal{S}_j$  to create the transcript for missing secret  $sk_j$  and then create response using the known secret  $sk_i$ .

## Disjunctive proofs



As a result, we get a sigma protocol for new relation  $R_1 \vee R_2$

$$(sk_1, sk_2, pk) \in R_1 \vee R_2 \iff (sk_1, pk) \in R_1 \vee (sk_2, pk) \in R_2 .$$

## The corresponding proof

**Perfect simulatability.** Note that  $\beta_1$  and  $\beta_2$  are independent and have a uniform distribution over  $\mathcal{B}$ . Consequently, we can run the canonical simulators  $\mathcal{S}_1$  and  $\mathcal{S}_2$  for  $\mathcal{V}_1$  and  $\mathcal{V}_2$  in parallel to create the properly distributed transcript  $(\alpha_1, \alpha_2, \beta_1 + \beta_2, \beta_1, \beta_2, \gamma_1, \gamma_2)$ .

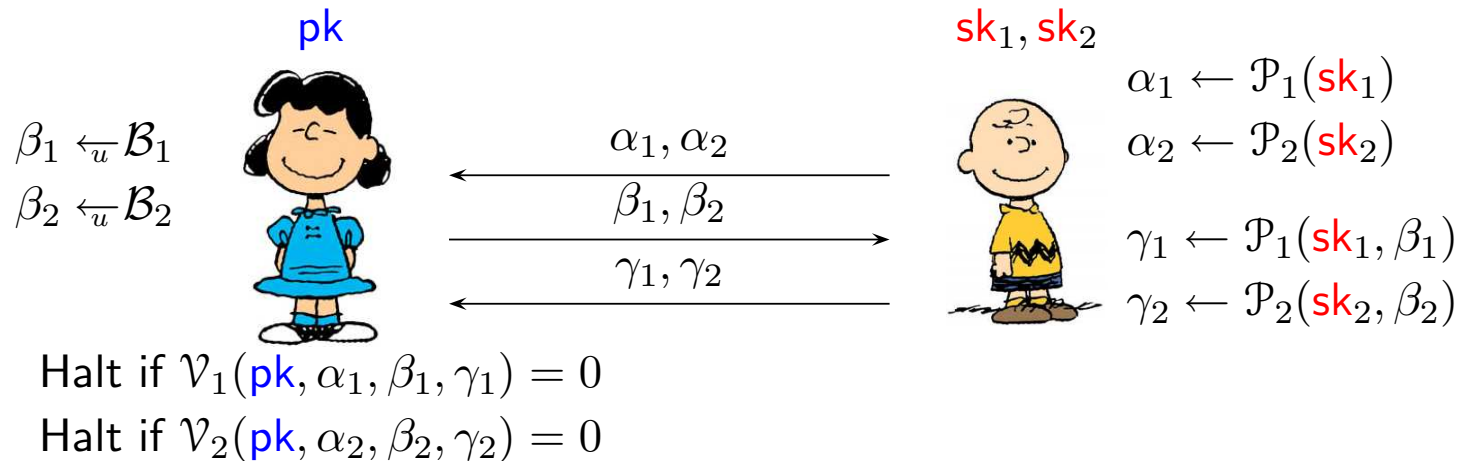
**Special soundness.** Given two transcripts

$$(\alpha_1, \alpha_2, \beta_1^0 + \beta_2^0, \beta_1^0, \beta_2^0, \gamma_1^0, \gamma_2^0), (\alpha_1, \alpha_2, \beta_1^1 + \beta_2^1, \beta_1^1, \beta_2^1, \gamma_1^1, \gamma_2^1)$$

such that  $\beta_1^0 + \beta_2^0 \neq \beta_1^1 + \beta_2^1$ , we can extract a colliding sub-transcript

$$\begin{cases} (\alpha_1, \beta_1^0, \gamma_1^0), (\alpha_1, \beta_1^1, \gamma_1^1), & \text{if } \beta_1^0 \neq \beta_1^1, \\ (\alpha_2, \beta_2^0, \gamma_2^0), (\alpha_2, \beta_2^1, \gamma_2^1), & \text{if } \beta_2^0 \neq \beta_2^1. \end{cases}$$

## Parallel composition



If we run two sigma protocols for different relations  $R_1$  and  $R_2$  in parallel, we get a sigma protocol\* for new relation  $R_1 \wedge R_2$

$$(sk_1, sk_2, pk) \in R_1 \wedge R_2 \iff (sk_1, pk) \in R_1 \wedge (sk_2, pk) \in R_2 .$$

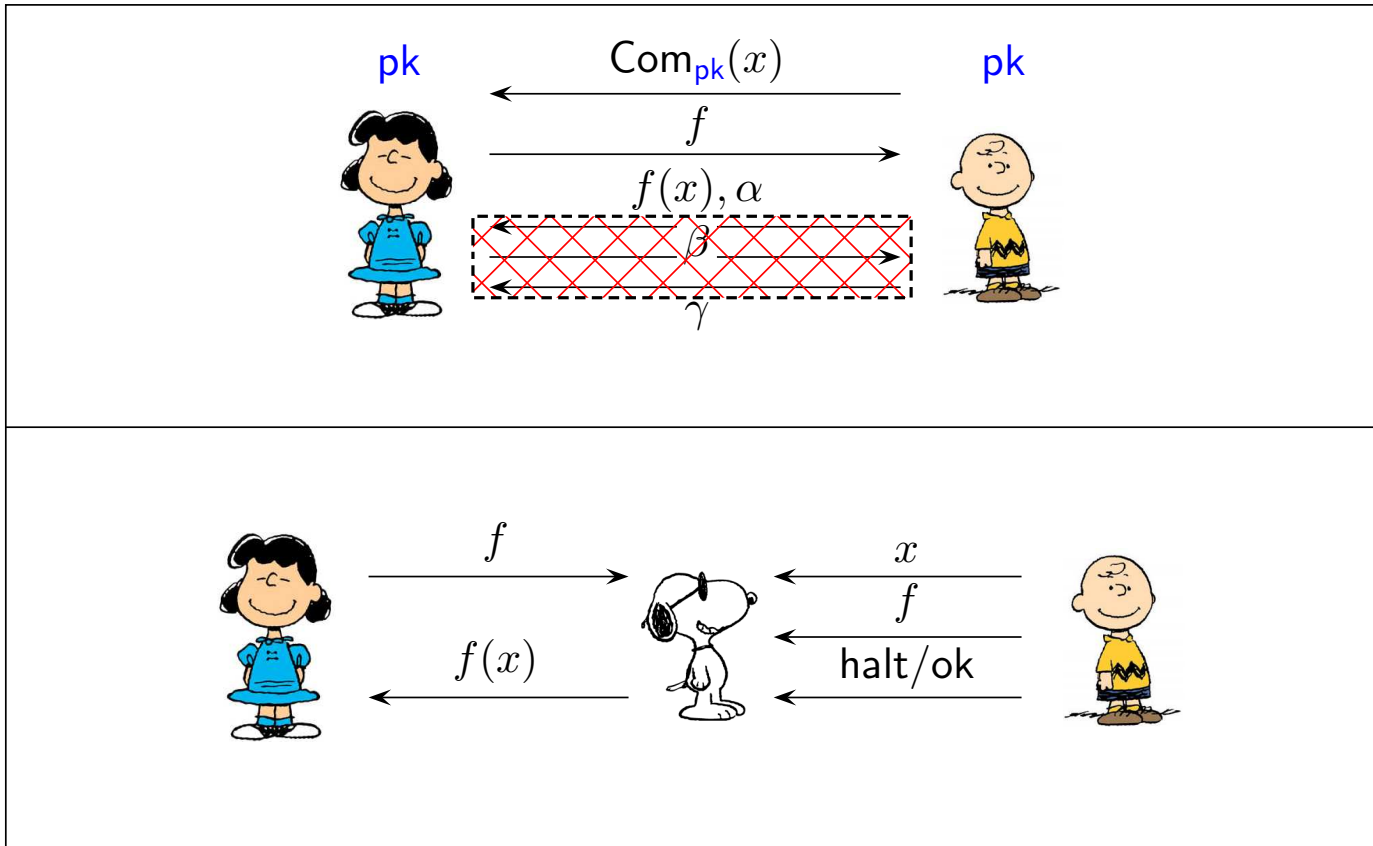
\* Modulo small details—not all colliding transcripts reveal both secrets.



# Certified Computations

Semihonest case

# The concept



Lucy should learn  $f(x)$  and nothing more even if Charlie is malicious.

## Basic tools

There are many ways how to build protocols for certified computations. Here, we consider one of the simplest protocols that is based DL group.

- ▷ We use Pedersen commitments with a public parameter  $y \leftarrow_u \mathbb{G}$

$$(y^x g^r, (x, r)) \leftarrow \text{Com}(x; r)$$

- ▷ We use proofs of knowledge for various relations about discrete logarithms

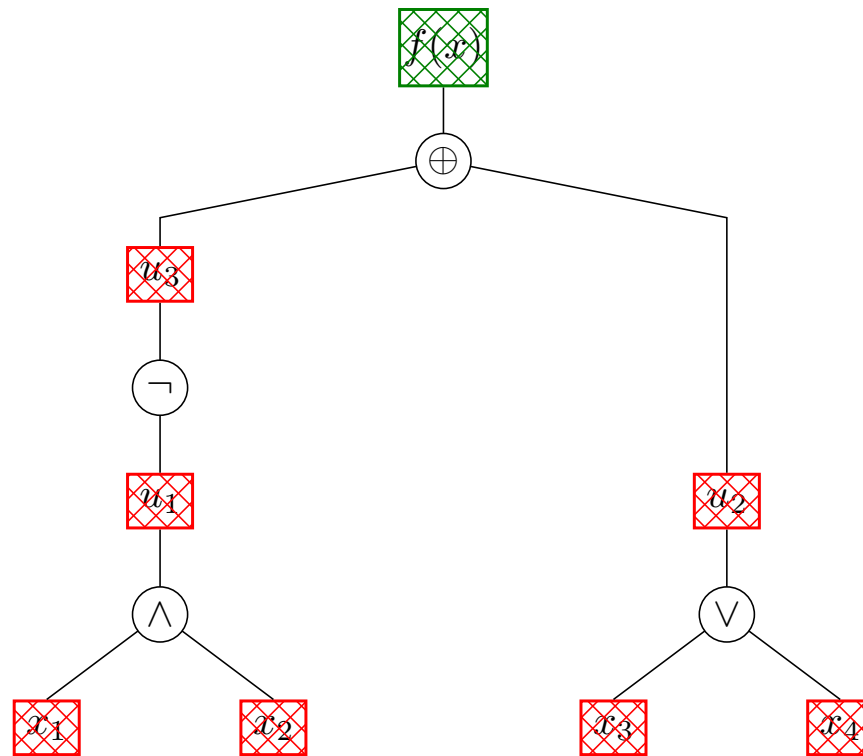
$$\text{POK}_{z,g} [\exists x : g^x = z]$$

$$\text{POK}_{g_1, g_2, z} [\exists x_1, x_2 : g_1^{x_1} g_2^{x_2} = z]$$

to prove more complex properties about Pedersen commitments.

# Boolean circuit of commitments

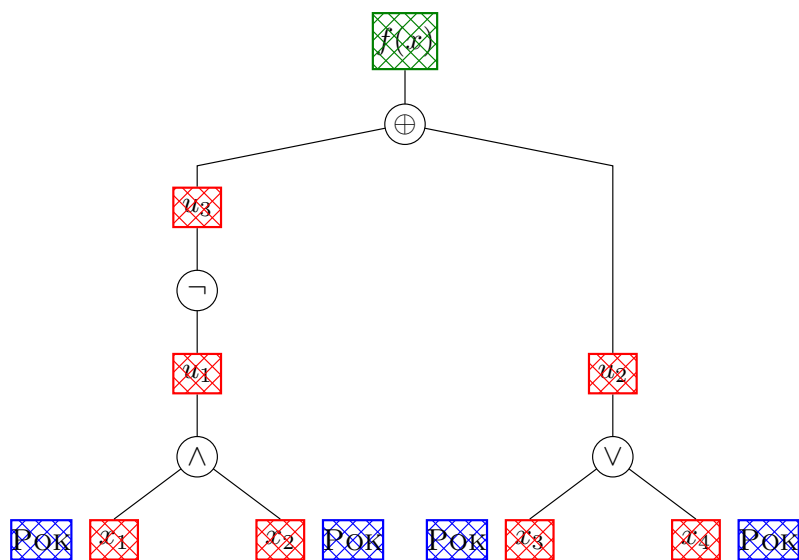
Charlie prepares a Boolean circuit for  $f$  and commits all intermediate values.



# Augmentation by proofs of knowledge I

Charlie proves that all commitments  $\text{Com}(x_i)$  are commitments of bits

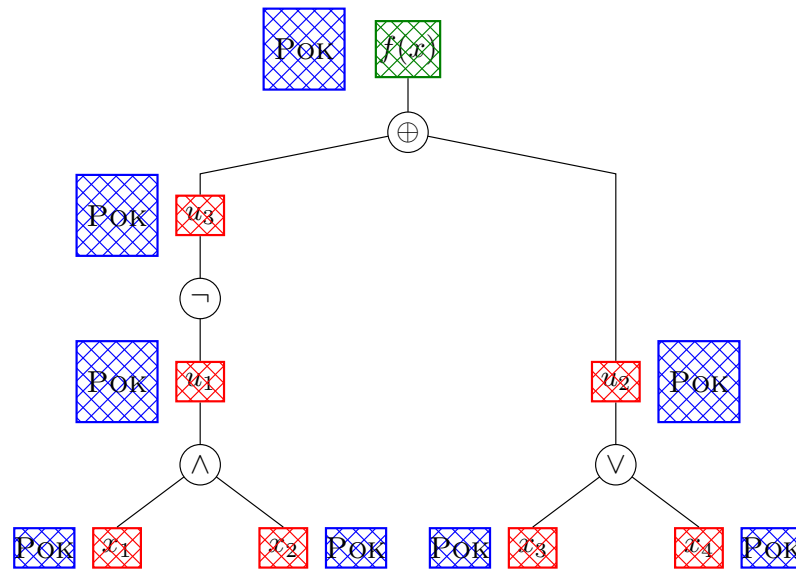
$$\text{POK}_{g,y,c} [\exists r : g^r = c \vee g^r = cy^{-1}]$$



# Augmentation by proofs of knowledge II

Charlie proves that all intermediate commitments are correct, e.g.

$$\text{POK}_{g,y,c_1,c_2}^{-1} \left[ \exists r_1 r_2 : g^{r_1} = c \wedge g^{r_2} = c_2 y^{-1} \vee \dots \right]$$



## Final protocol

Since we can use disjunctive composition to combine all sigma proofs, we get the following protocol for certified computations.

- ▷ Charlie commits his input bit by bit using Pedersen commitment.
- ▷ Lucy sends the description of a function  $f$ .
- ▷ Charlie creates Boolean circuit and commits all values.
- ▷ Both parties agree on the corresponding validity proof.
- ▷ Charlie decommits  $f(x)$  and sends the first proof message  $\alpha$ .
- ▷ Lucy sends the challenge message  $\beta \leftarrow \mathcal{B}$ .
- ▷ Charlie sends back the corresponding response  $\gamma$ .
- ▷ Lucy accepts  $f(x)$  only if the sigma protocol succeeds.