

# PART 1

## SQLsign

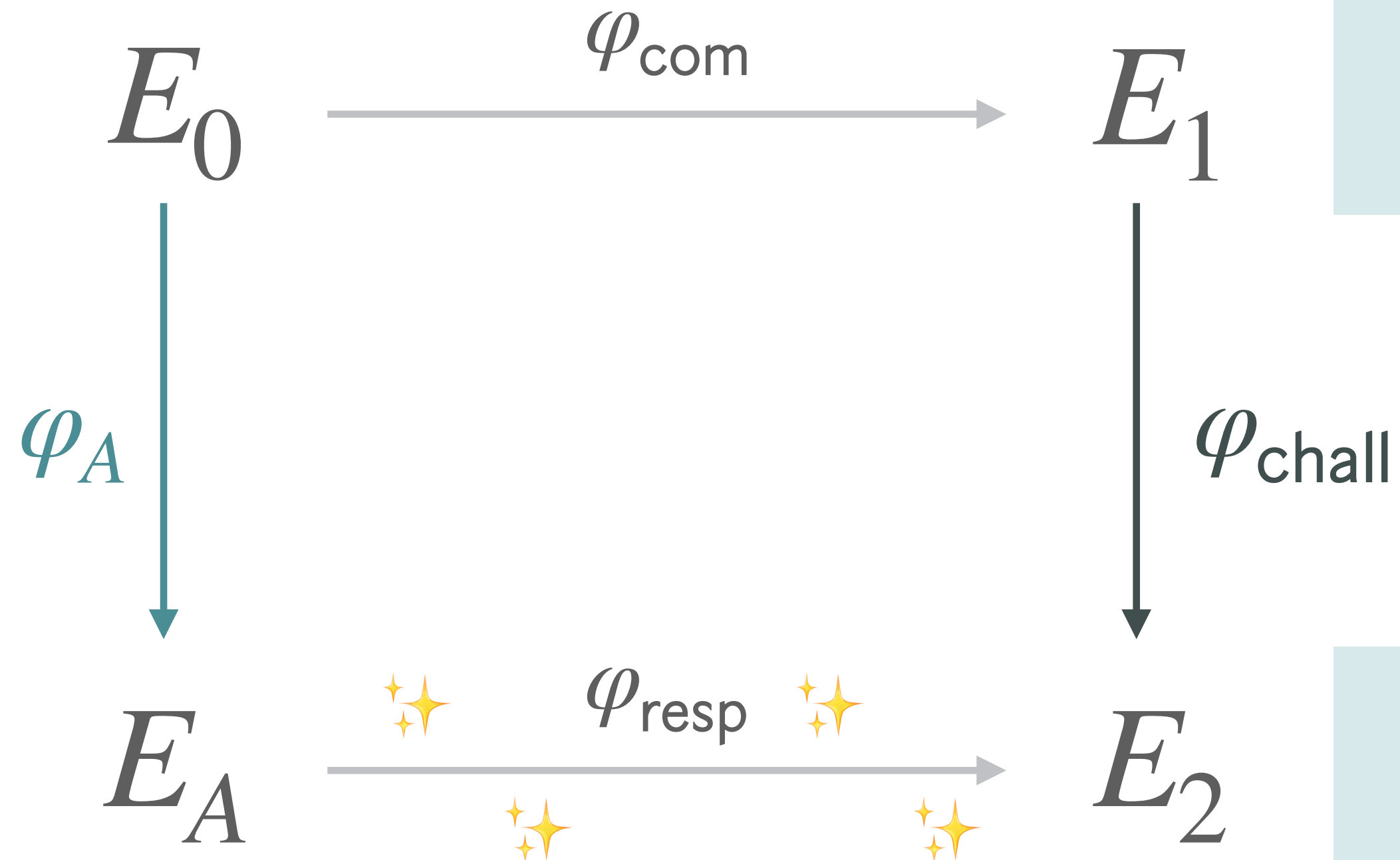
### Identification protocol

- **Commitment:** random isogeny  $\varphi_{\text{com}} : E_0 \rightarrow E_1$
- **Challenge:** semi-random isogeny  $\varphi_{\text{chall}} : E_1 \rightarrow E_2$
- **Response:** “matching” isogeny  $\varphi_{\text{resp}} : E_A \rightarrow E_2$

### signature scheme

replace semi-random  $\varphi_{\text{chall}}$   
by a challenge isogeny generated  
from  $\text{SHAKE256}(\text{msg} || E_1)$


everyone knows  
 $\text{End}(E_0)$



only **you** know  
 $\varphi_{\text{com}}$  and  $\text{End}(E_1)$

only **you** know  
 $\varphi_A$  and  $\text{End}(E_A)$

only **you** know  
 $\text{End}(E_2)$

 **Fact:** ONLY, given  $\text{End}(E_A)$  and  $\text{End}(E_2)$   
you can compute a proper response

## PART 1

# SQLsign

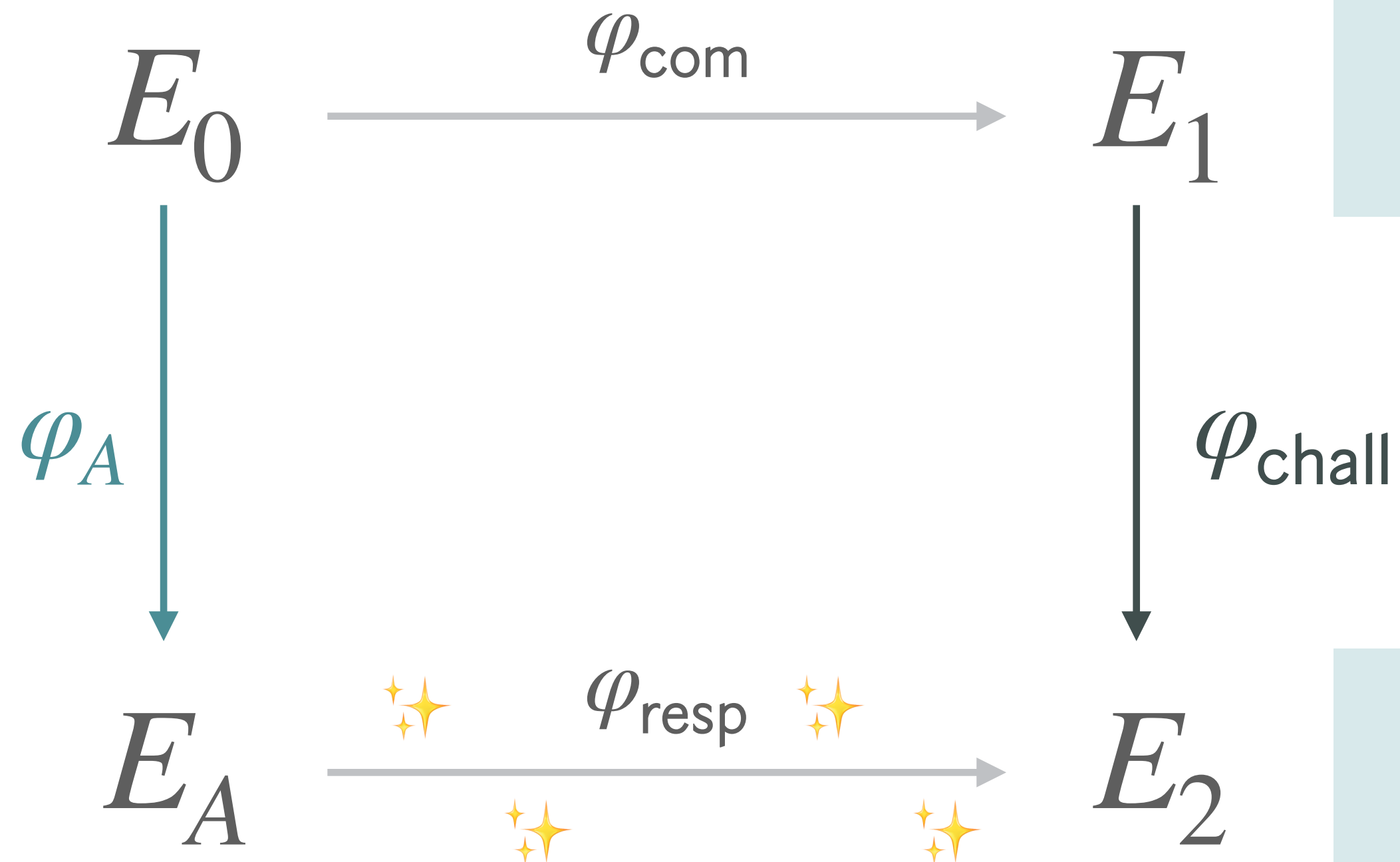
### Identification protocol

- **Commitment:** random isogeny  $\varphi_{\text{com}} : E_0 \rightarrow E_1$
- **Challenge:** semi-random isogeny  $\varphi_{\text{chall}} : E_1 \rightarrow E_2$
- **Response:** “matching” isogeny  $\varphi_{\text{resp}} : E_A \rightarrow E_2$

### signature scheme

replace semi-random  $\varphi_{\text{chall}}$   
by a challenge isogeny generated  
from  $\text{SHAKE256}(\text{msg} || E_1)$

everyone knows  
 $\text{End}(E_0)$



only **you** know  
 $\varphi_{\text{com}}$  and  $\text{End}(E_1)$

only **you** know  
 $\varphi_A$  and  $\text{End}(E_A)$

only **you** know  
 $\text{End}(E_2)$

### WARNING!

with this approach  
the response will  
be **large**, degree  $2^{1000}$



**Fact:** ONLY, given  $\text{End}(E_A)$  and  $\text{End}(E_2)$   
you can compute a proper response