Does Secure Time-Stamping Imply Collision-Free Hash Functions

Ahto Buldas, Aivo Jürgenson

aivo.jurgenson@eesti.ee

Tallinn University of Technology, Estonia.

Elion Enterprises Ltd, Estonia.

Topics

- background about hash functions and their security
- timestamping and backdating attack
- what is blackbox reduction
- how to prove that blackbox reduction is not possible
- show that time-stamping doesn't require CHFH

Hash functions

- $X \in \{0,1\}^*$, $x = h(X)$, $x \in \{0,1\}^m$

Hash functions

- $X \in \{0, 1\}^*$, $x = h(X)$, $x \in \{0, 1\}^m$
- $X_1 \neq X_2$, $h(X_1) = h(X_2)$

Hash functions

- $X \in \{0,1\}^*$, $x = h(X)$, $x \in \{0,1\}^m$
- $X_1 \neq X_2$, $h(X_1) = h(X_2)$
- attacks against collision resistance of MD5, SHA-1, SHA-256

Hash functions

- $X \in \{0,1\}^*$, $x = h(X)$, $x \in \{0,1\}^m$
- $X_1 \neq X_2$, $h(X_1) = h(X_2)$
- attacks against collision resistance of MD5, SHA-1, SHA-256
- is this *collision freedom* really required in applications (for example in timestamping)?

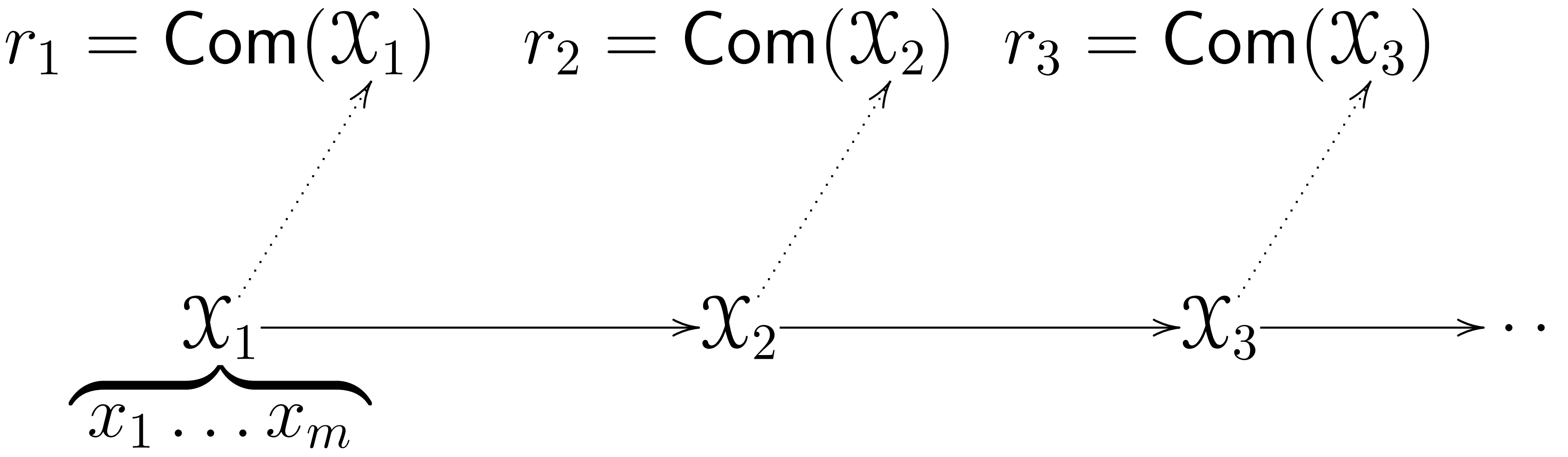
Hash functions

- $X \in \{0,1\}^*$, $x = h(X)$, $x \in \{0,1\}^m$
- $X_1 \neq X_2$, $h(X_1) = h(X_2)$
- attacks against collision resistance of MD5, SHA-1, SHA-256
- is this *collision freedom* really required in applications (for example in timestamping)?
- Buldas and Saarepera in 2004: collision freedom is *insufficient*.
- Buldas and Laur in 2006: collision freedom is *unneccessary*.

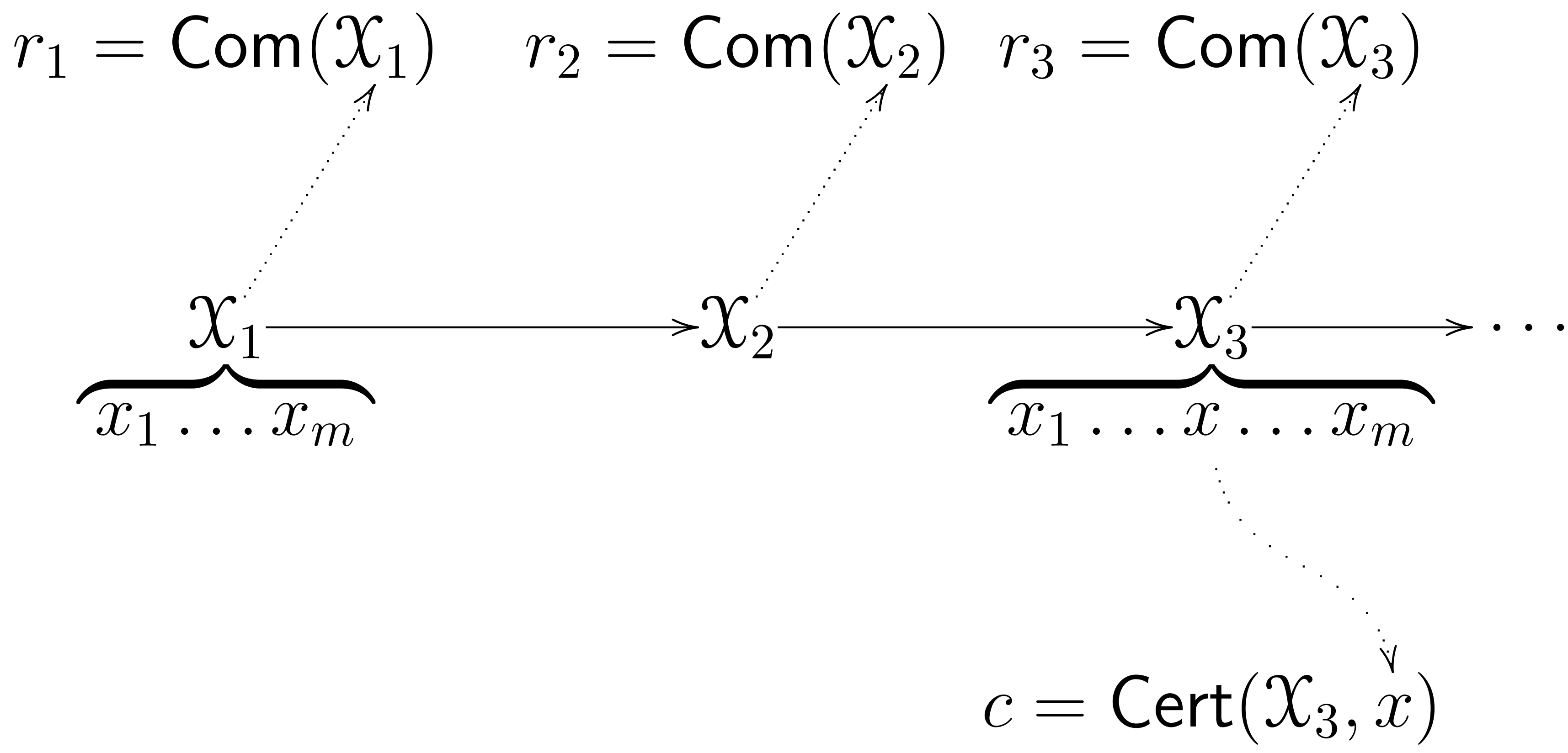
Timestamping scheme

$$\underbrace{\mathcal{X}_1}_{x_1 \ldots x_m} \longrightarrow \mathcal{X}_2 \longrightarrow \mathcal{X}_3 \longrightarrow \cdots$$

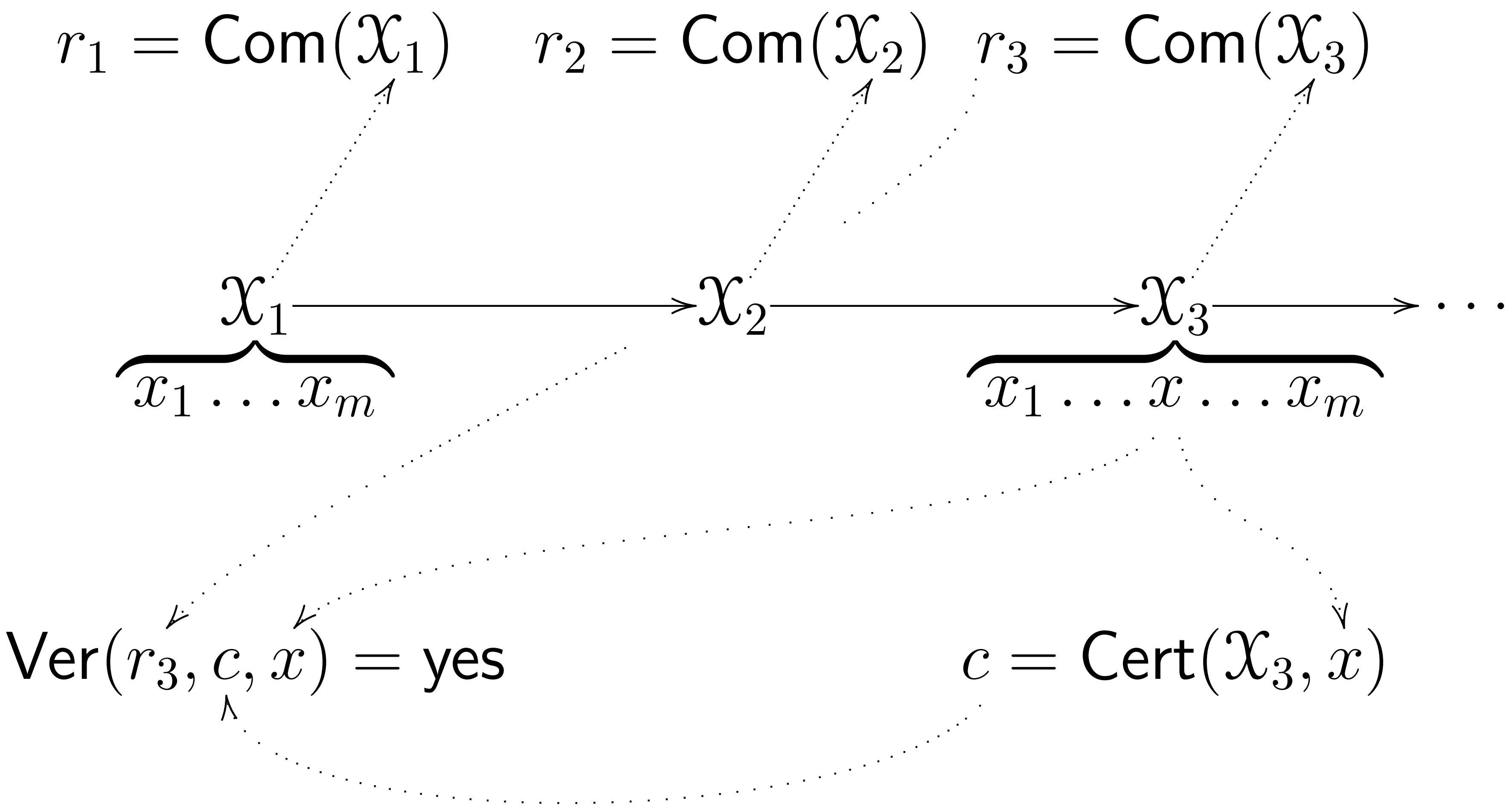
Timestamping scheme



Timestamping scheme



Timestamping scheme



Backdating attack

Backdating attack

- Adversary publishes commitment r.

Backdating attack

- Adversary publishes commitment r.
- Alice invents something $\mathcal{D}_{\mathrm{A}} \in \{0,1\}^*$.

Backdating attack

- Adversary publishes commitment r.
- Alice invents something $\mathcal{D}_{\mathrm{A}} \in \{0,1\}^*$.
- Adversary creates a modified description of the Alice's invention $\mathcal{D}'_{\mathrm{A}} \in \{0,1\}^*$ and claims that this was timestamped by himself long before Alice invented it.

Backdating attack

- Adversary publishes commitment r.
- Alice invents something $\mathcal{D}_{\mathrm{A}} \in \{0,1\}^*$.
- Adversary creates a modified description of the Alice's invention $\mathcal{D}'_{\mathrm{A}} \in \{0,1\}^*$ and claims that this was timestamped by himself long before Alice invented it.
- $x = H(\mathcal{D}'_{\mathrm{A}})$, $\mathsf{Ver}(r, x, c) =$ yes

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$(r, a) \leftarrow \mathsf{A}_1(1^k)$$

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$(r, a) \leftarrow \mathsf{A}_1(1^k), (x, c) \leftarrow \mathsf{A}_2(r, a)$$

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$(r, a) \leftarrow \mathsf{A}_1(1^k), (x, c) \leftarrow \mathsf{A}_2(r, a) :$$

$$\mathsf{Ver}(x, c, r) = \mathsf{yes}$$

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$\Pr\Big[(r, a) \leftarrow \mathsf{A}_1(1^k), (x, c) \leftarrow \mathsf{A}_2(r, a) : \mathsf{Ver}(x, c, r) = \mathsf{yes}\Big] = k^{-\omega(1)}$$

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$\Pr\Big[(r,a) \leftarrow \mathsf{A}_1(1^k), (x,c) \leftarrow \mathsf{A}_2(r,a) : \mathsf{Ver}(x,c,r) = \mathsf{yes}\Big] = k^{-\omega(1)}$$

- $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2) \in \mathsf{FPU}$ when

$$(r,a) \leftarrow \mathsf{A}_1(1^k),$$

$$(x,c) \leftarrow \mathsf{A}_2(r,a)$$

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$\Pr\Big[(r,a) \leftarrow \mathsf{A}_1(1^k), (x,c) \leftarrow \mathsf{A}_2(r,a) : \mathsf{Ver}(x,c,r) = \mathsf{yes}\Big] = k^{-\omega(1)}$$

- $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2) \in \mathsf{FPU}$ when

$$(r,a) \leftarrow \mathsf{A}_1(1^k), x' \leftarrow \Pi(r,a),$$

$$(x,c) \leftarrow \mathsf{A}_2(r,a) \colon x' = x$$

Formalized attack

- Two-staged adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$.
- Security condition:

$$\Pr\Big[(r,a) \leftarrow \mathsf{A}_1(1^k), (x,c) \leftarrow \mathsf{A}_2(r,a) : \mathsf{Ver}(x,c,r) = \mathsf{yes}\Big] = k^{-\omega(1)}$$

- $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2) \in$ FPU when

$$\Pr\Big[(r,a) \leftarrow \mathsf{A}_1(1^k), x' \leftarrow \Pi(r,a), (x,c) \leftarrow \mathsf{A}_2(r,a) \colon x' = x\Big] = k^{-\omega(1)}$$

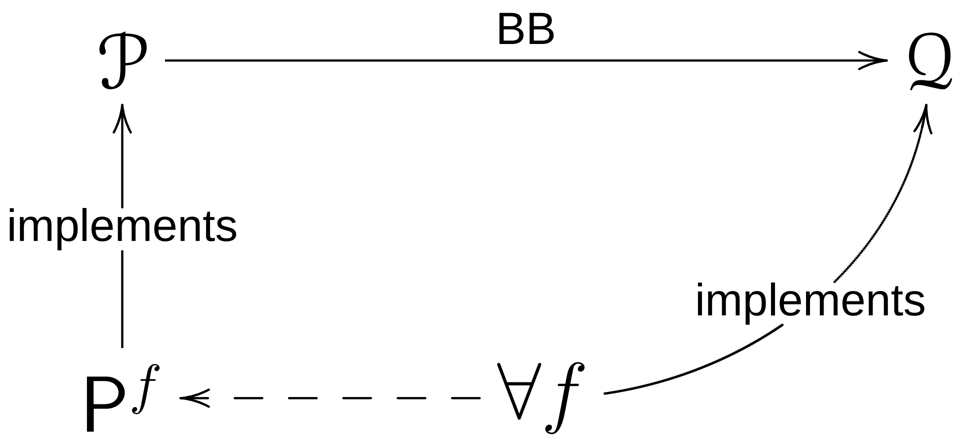
BlackBox reduction

$$\mathcal{P} \xrightarrow{\text{BB}} \mathcal{Q}$$

P

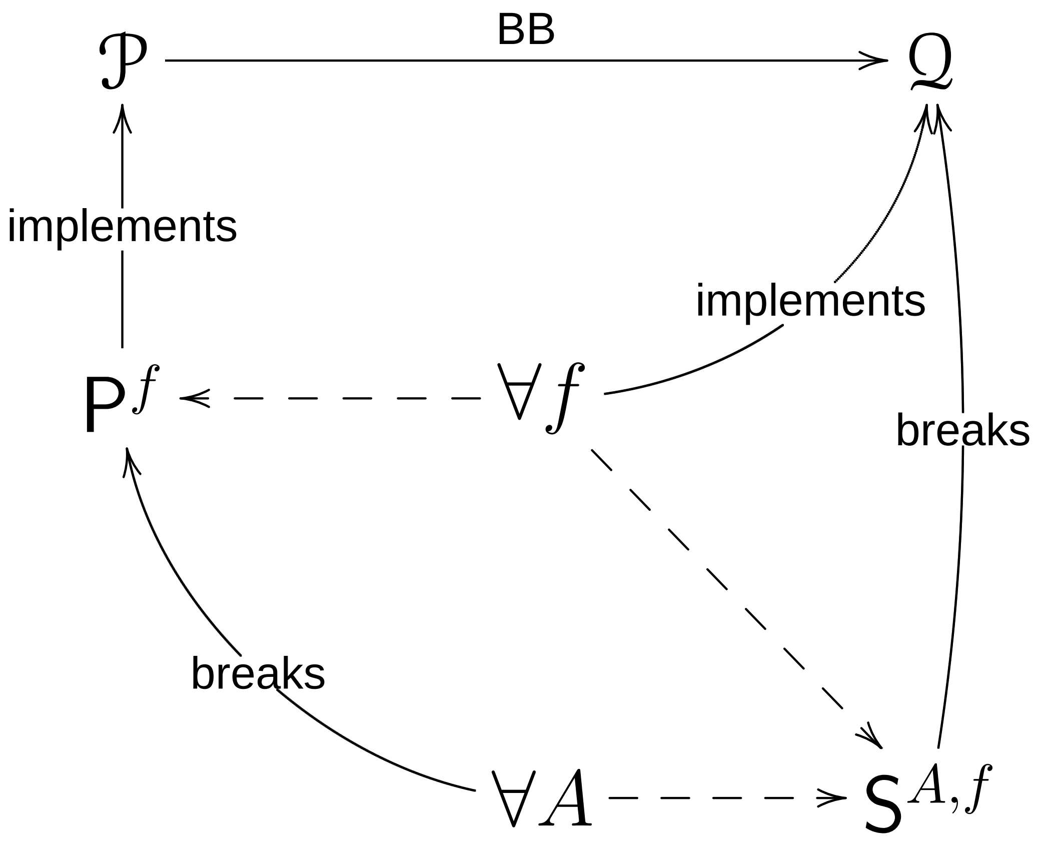
S

BlackBox reduction



S

BlackBox reduction



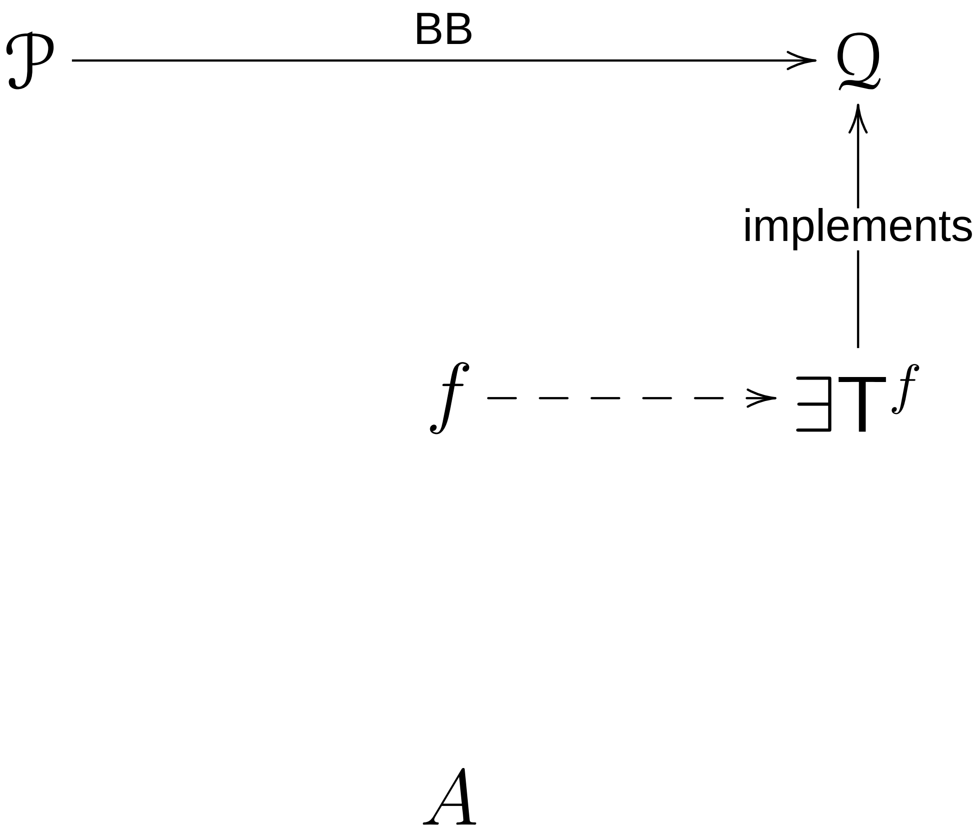
Oracle separation

$$\mathcal{P} \xrightarrow{\text{BB}} \mathcal{Q}$$

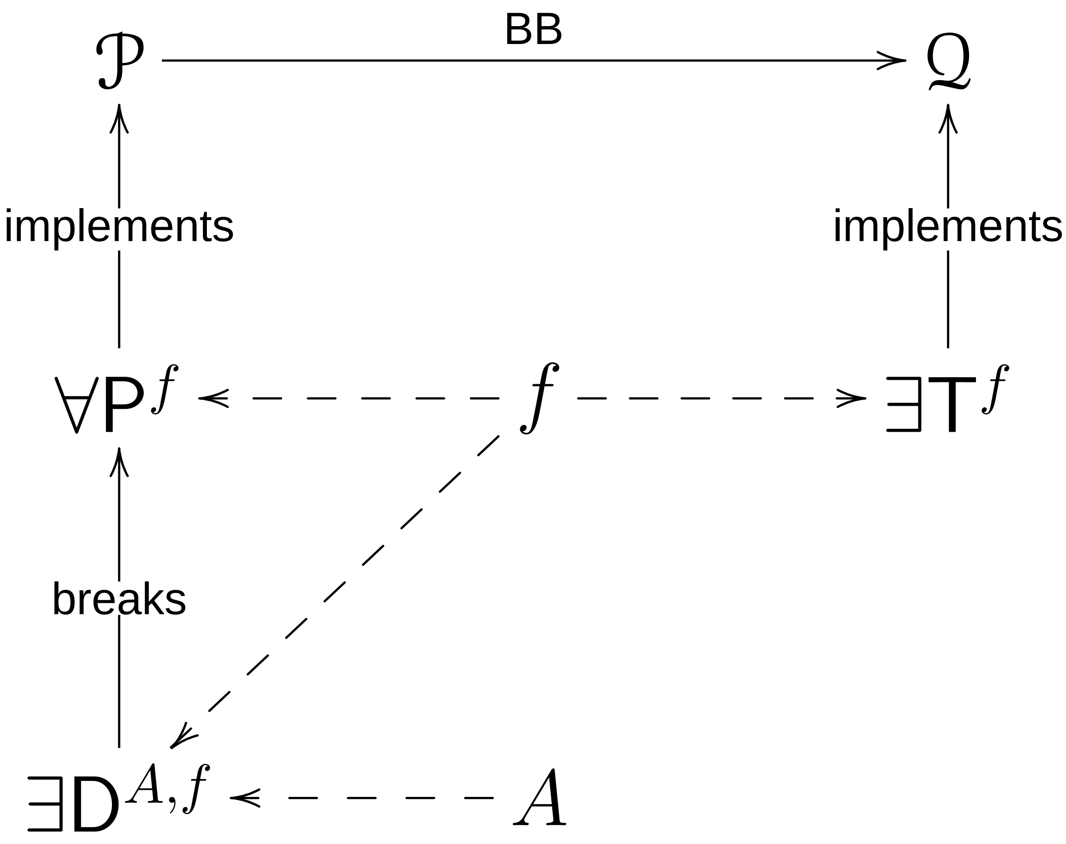
$$f$$

$$A$$

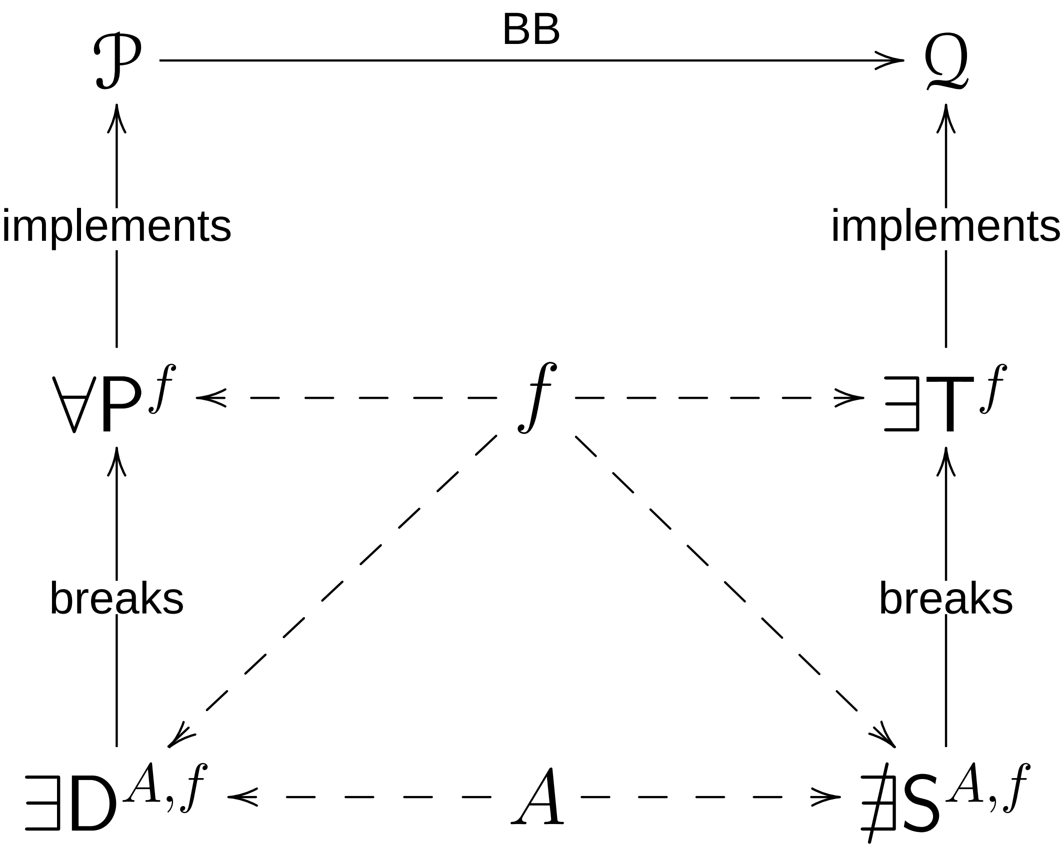
Oracle separation



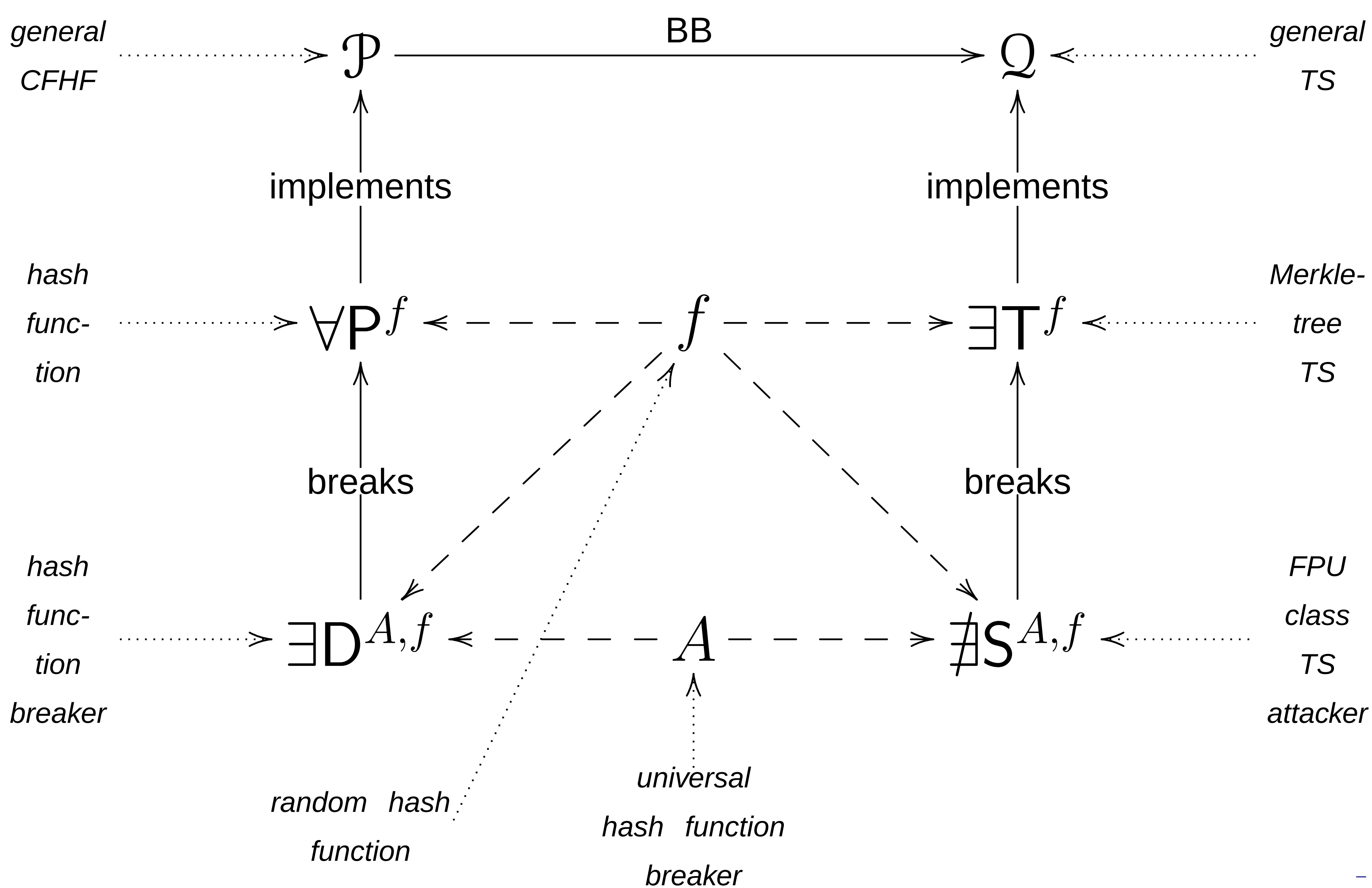
Oracle separation



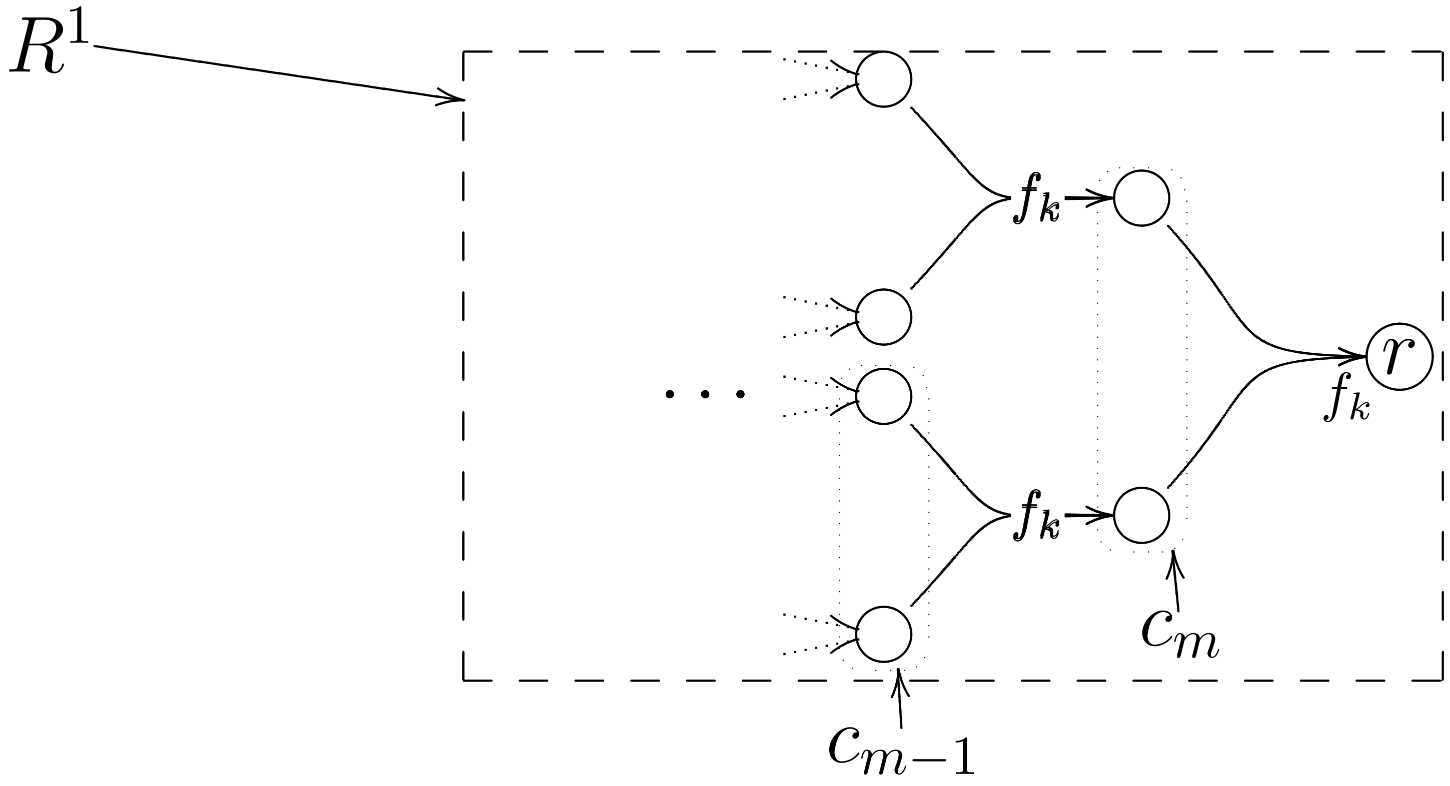
Oracle separation



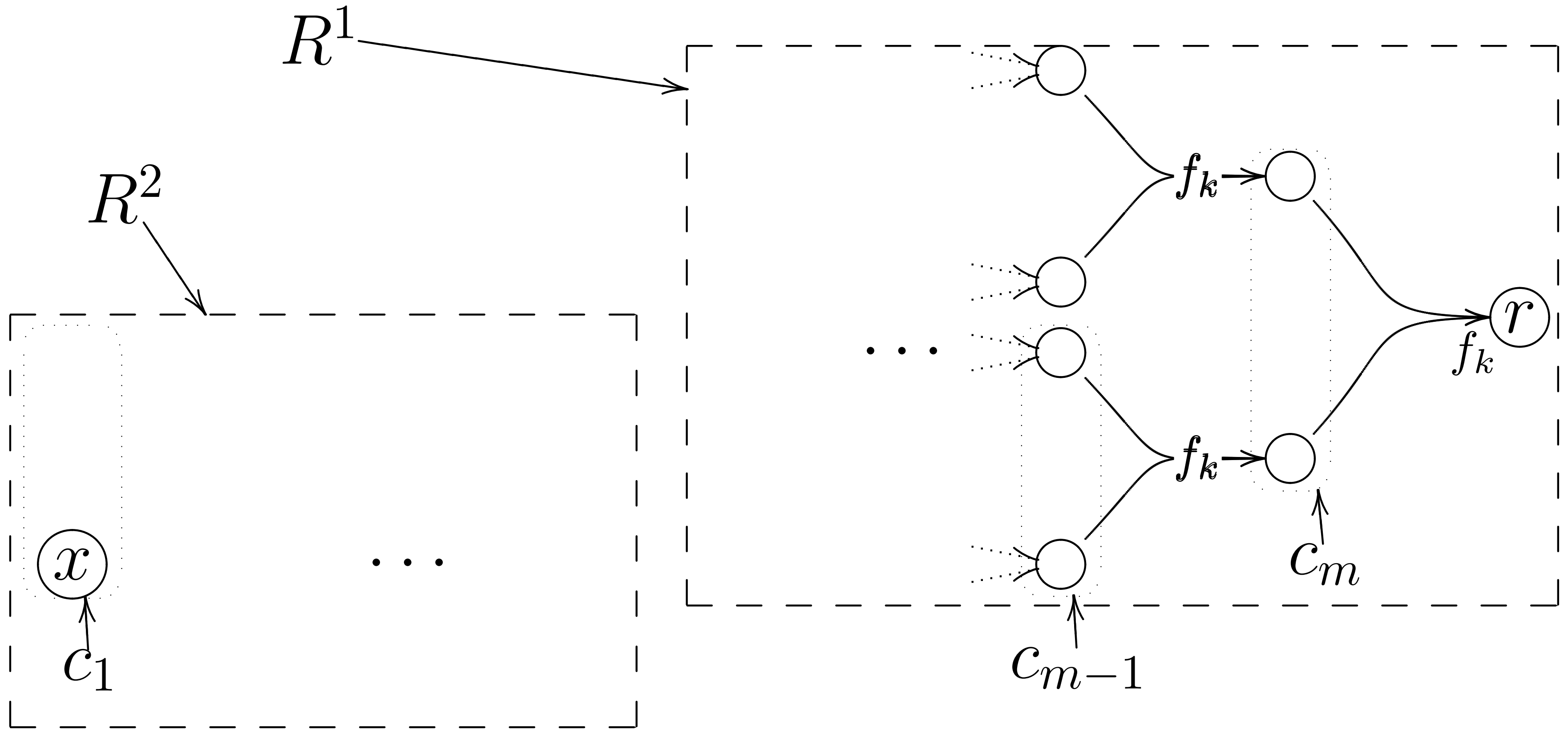
Oracle separation



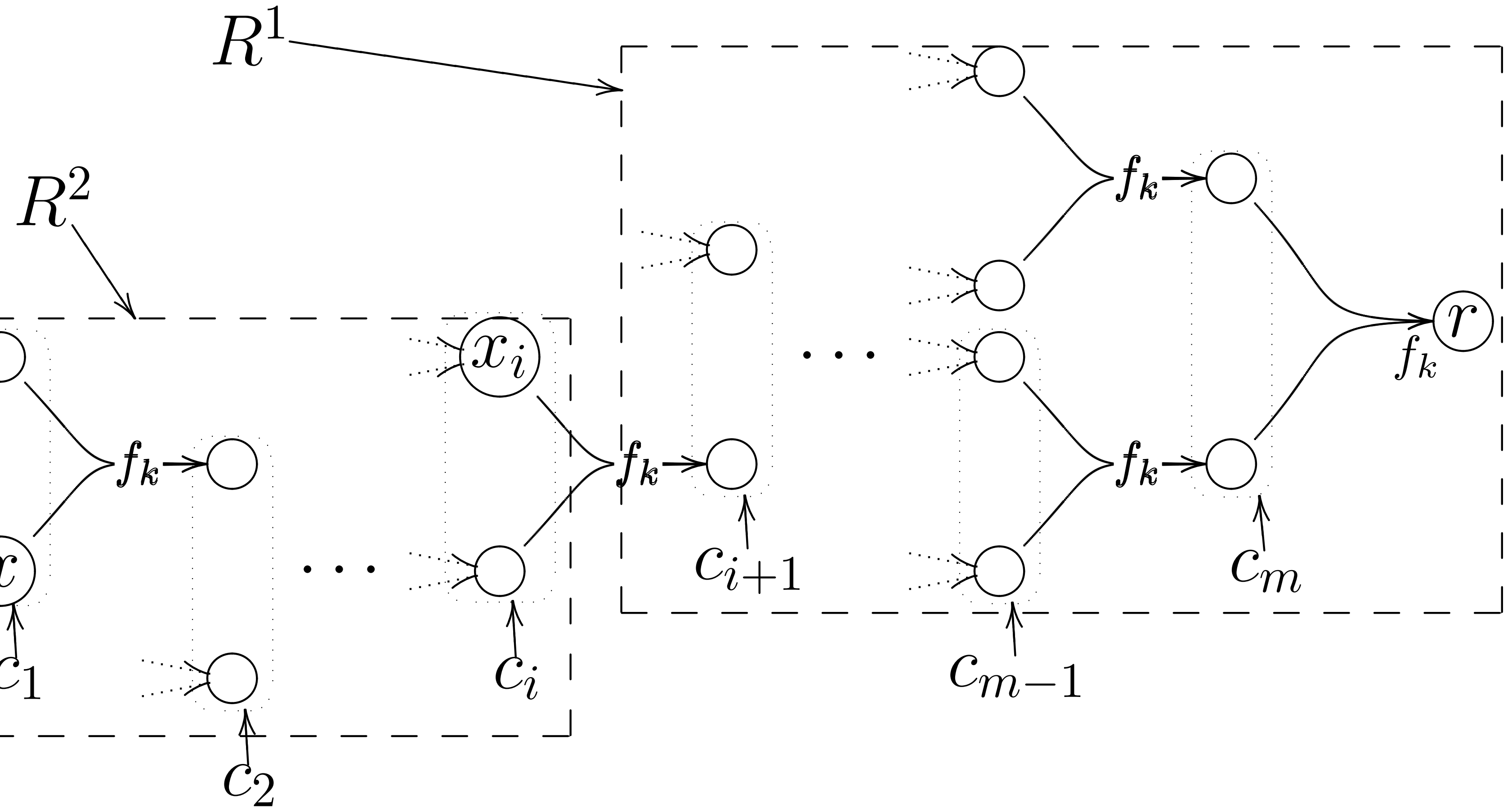
$\mathsf{S}^{A,f} = (\mathsf{S}_1, \mathsf{S}_2)$ in work



$\mathsf{S}^{A,f} = (\mathsf{S}_1, \mathsf{S}_2)$ in work



$\mathsf{S}^{A,f} = (\mathsf{S}_1, \mathsf{S}_2)$ in work



Conclusion

- $$\Pr\Big[(r,a) \leftarrow \mathsf{S}_1^{A,f}(1^k), (x,c) \leftarrow \mathsf{S}_2^{A,f}(r,a) : \mathsf{Ver}(x,c,r) = \mathsf{yes}\Big] = k^{-\omega(1)}$$
- blackbox reduction of CFHF to TS is not possible