Master Thesis

Robust Combiners for Cryptographic Protocols

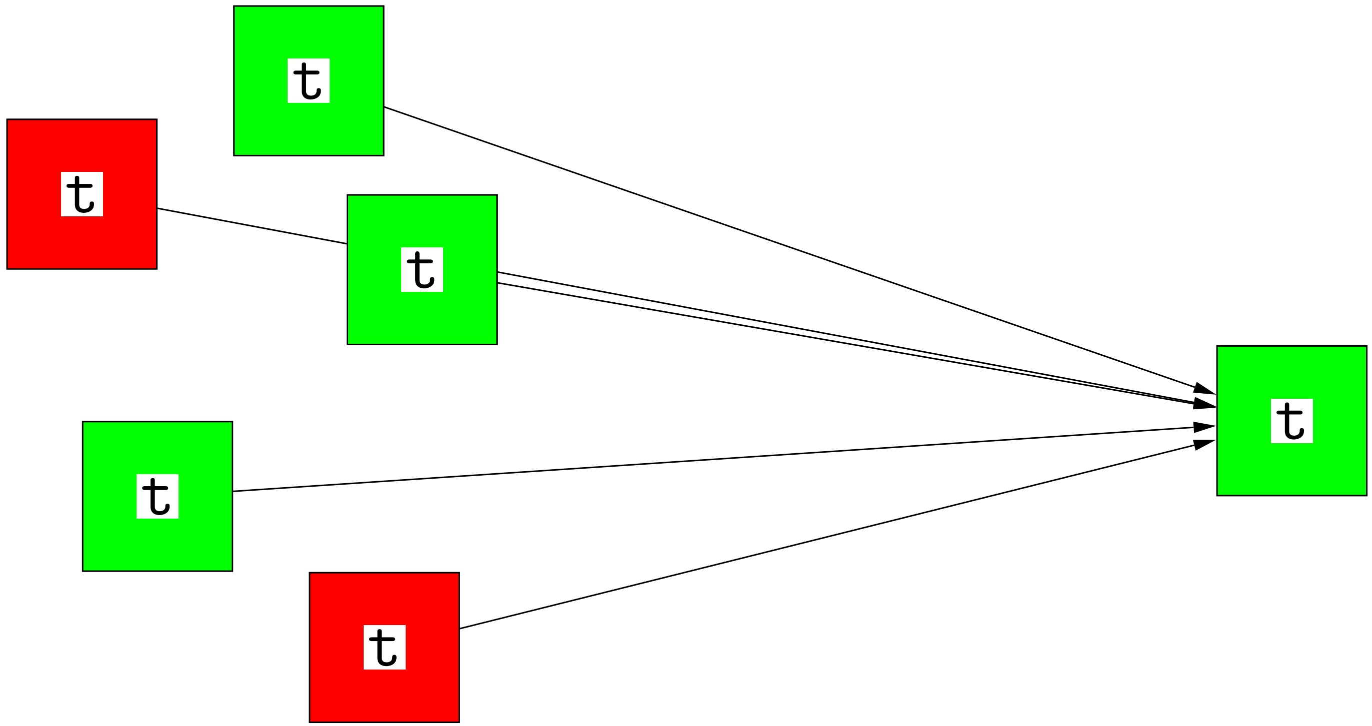
Christian Sommer, September 2006
Supervisor: Bartosz Przydatek

Motivation

- Cryptography relies on computational assumptions.
- The hardness of these is unproven (factoring, discrete $\log$).
- In the design of primitives, what assumptions shall we rely on?
- (roughly:) Combiners allow to rely on the strongest assumption without knowing which one it is.

(k, n) combiners, definition (Harnik et al., Eurocrypt 2005)

- Cryptographic primitive $\mathbf{T}$
- n implementations t
- If k implementations are secure, combiner is secure



[]

(k, n) **combiners, variations (Harnik et al., Eurocrypt 2005)**

Third-party black-box:

- inputs/outputs from third-party
- no transcript

Transparent black-box:

- for interactive primitives
- combiner can use transcript
- all messages sent to other party (on-line access only)

Black-box:

- use implementations as black-boxes
- off-line access possible

Related work

Harnik *et al.*, Eurocrypt 2005

- No transparent black-box $(1,2)$-robust combiner for $\mathbf{OT}$
- Third party black-box $(2,3)$-robust combiner for $\mathbf{OT}$

Herzberg, RSA 2005

- Analysis of folklore construction
- Sharing combiner for $\mathbf{BC}$ (majority secure)

Meier, Przydatek, Crypto 2006

- Black-box $(1,2)$-robust combiner for $\mathbf{PIR}$
- $\mathbf{PIR}$-to-$\mathbf{BC}$, $\mathbf{PIR}$-to-$\mathbf{OT}$ combiner

Work of this thesis

BC

- No transparent black-box $(1,2)$-robust combiner for $\mathbf{BC}$ (proof similar to Harnik *et al.* 2005)
- Analysis of sharing combiners for $\mathbf{BC}$ from Herzberg 2005 (results slightly improved according to different definition)
- Proof for black-box $(1,2)$-robust combiner for $\mathbf{BC}$ (known)

IP
PIR

Bit commitment

- $\textsc{Commit} : \{0,1\} \times \{0,1\}^m \rightarrow \{0,1\}^n$
  $(b, \rho) \mapsto c$, send result c
- $\textsc{Open}$: send randomness ρ and bit b
- Hiding: impossible/hard to compute b from c
- Binding: impossible/hard to open c for another $b' \neq b$

Possible $(1,2)$ combiner inputs, notation

- $(\mathtt{bc}^{H}, \mathtt{bc}^{bH})$
  First implementation: guaranteed hiding
  Second: guaranteed binding and hiding
  (capital letter: emphasis on inf.-th./stat. property)
- $(\mathtt{bc}^{bH}, \mathtt{bc}^{H})$
  positions changed
- $\{(\mathtt{bc}^{H}, \mathtt{bc}^{bH}), (\mathtt{bc}^{bH}, \mathtt{bc}^{H})\}$
  Set of all possible inputs specifies what the combiner is capable to handle.
- (Often: all permutations of one input, as above)
- $(1,2)$-combiner for $\mathbf{BC}$ handles $\{(\mathtt{bc}, \mathtt{bc}^{bh}), (\mathtt{bc}^{bh}, \mathtt{bc})\}$

Warm up, information-theoretic hiding

- Hiding is information-theoretic secure, never broken
- One binding assumption might be broken

$$\{(\texttt{bc}^{H}, \texttt{bc}^{bH}), (\texttt{bc}^{bH}, \texttt{bc}^{H})\}$$

- Commit to the same bit twice

Warm up, information-theoretic binding

- Binding is information-theoretic secure, never broken
- One hiding assumption might be broken

$\{(\mathtt{bc}^{B}, \mathtt{bc}^{Bh}), (\mathtt{bc}^{Bh}, \mathtt{bc}^{B})\}$

- Commit to b_1 and b_2, where $b = b_1 \oplus b_2$

Information-theoretic binding/hiding

- First scheme is information-theoretic binding
- Second scheme is information-theoretic hiding
- (at least) one cryptographic assumption holds

$\{(\mathtt{bc}^{Bh}, \mathtt{bc}^{H}), (\mathtt{bc}^{B}, \mathtt{bc}^{bH})\}$

- No transparent black-box combiner possible!
- $\Rightarrow$ no transparent black-box $(1, 2)$-robust combiner possible

Proof idea (adapted from Harnik et al.)

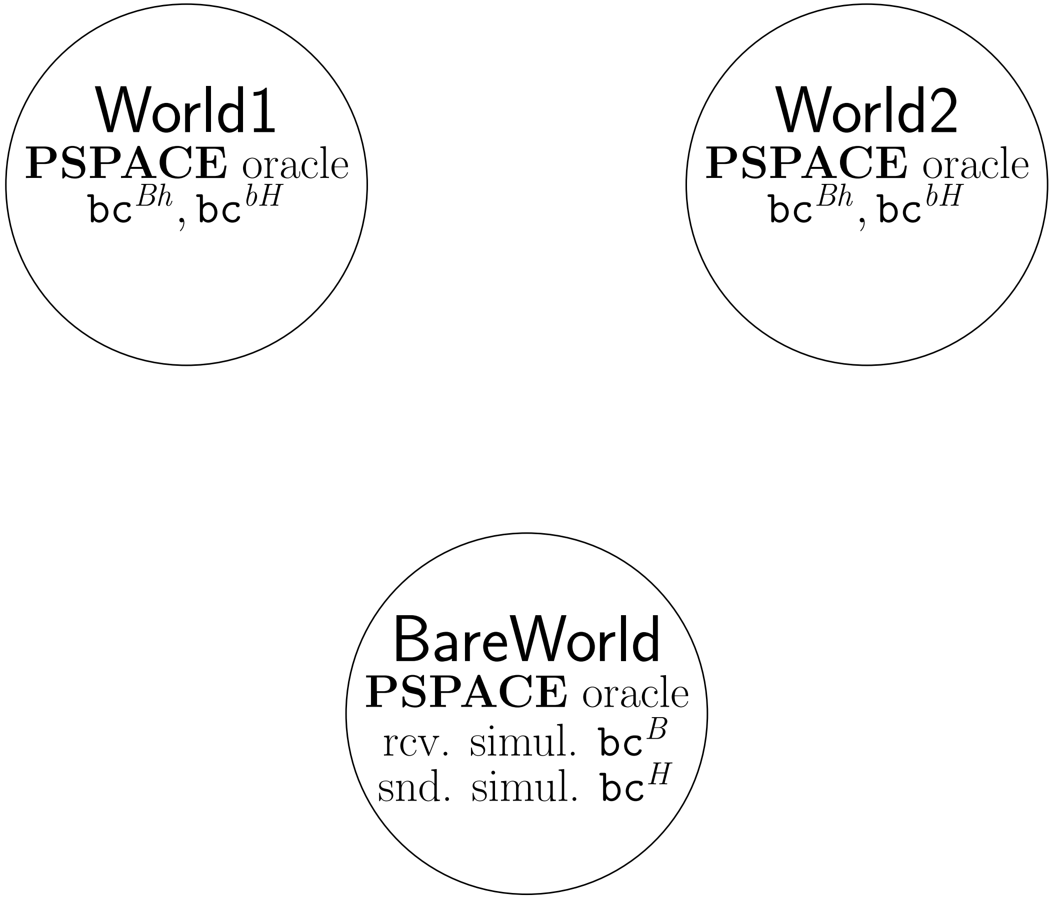
- Standard proof: construct a world where $\mathbf{BC}$ exists (with oracles) but combiners do not.
- Combiner could use only secure implementation, i.e., it exists.
- Therefore, we show two worlds and in at least one the situation is as described.

Random oracles for BC

- $\mathsf{bc}^{Bh} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$, no collision

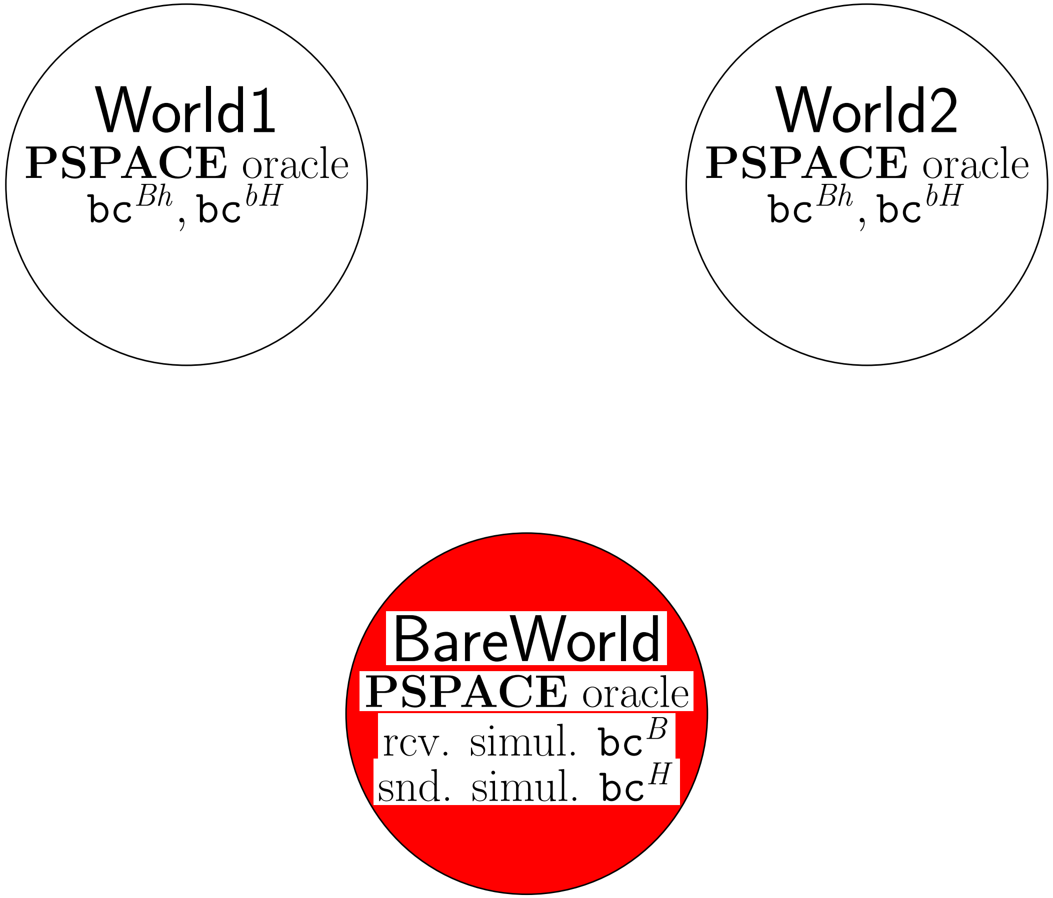
- $\mathsf{bc}^{bH} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^n$, two random strings per c
  $\forall c \in \{0,1\}^n \exists r_0, r_1 \in \{0,1\}^n : r_0 \neq r_1 \wedge \mathsf{bc}^{bH}(0, r_0) = \mathsf{bc}^{bH}(1, r_1) = c$

No transparent black-box $(1, 2)$-combiner for BC, illustration



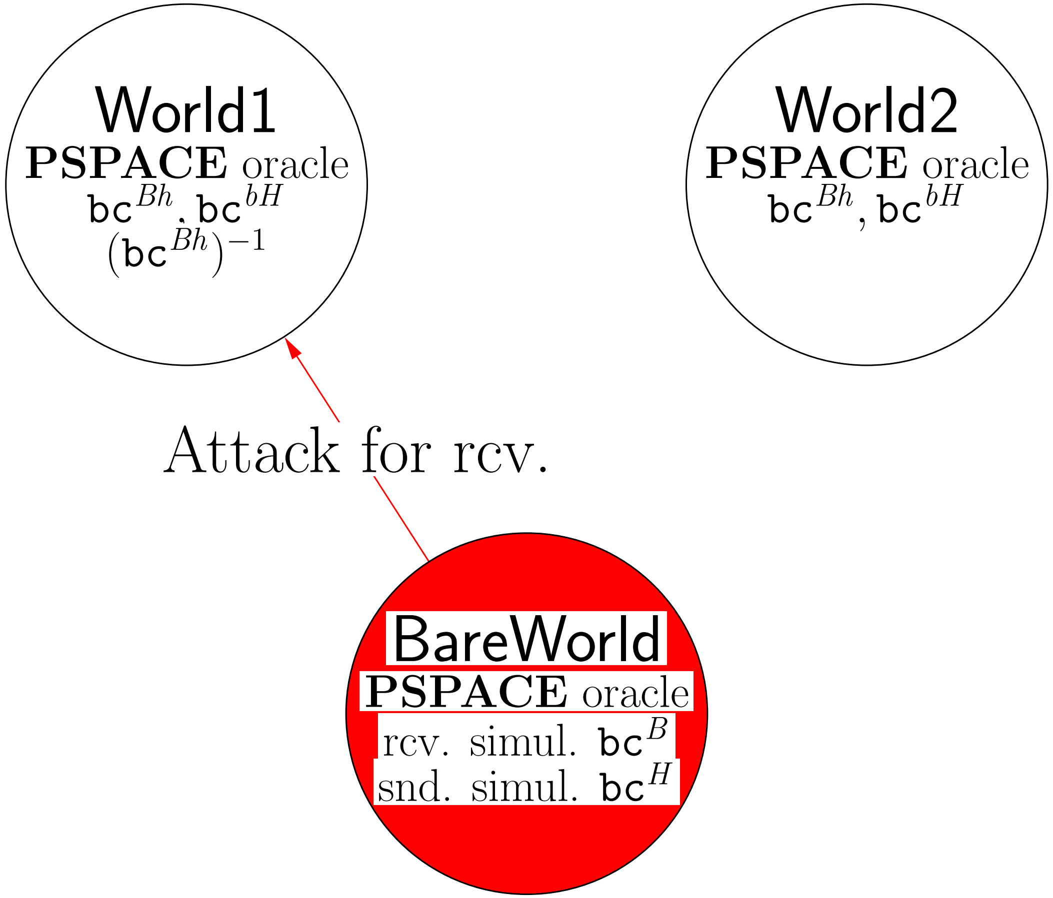
- $\mathsf{bc}^{Bh} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$ (no collision)
- $\mathsf{bc}^{bH} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^n$ (two random strings per c)
  $\forall c \in \{0,1\}^n \exists r_0, r_1 \in \{0,1\}^n : r_0 \neq r_1 \wedge \mathsf{bc}^{bH}(0, r_0) = \mathsf{bc}^{bH}(1, r_1) = c$

No transparent black-box $(1,2)$-combiner for BC, illustration



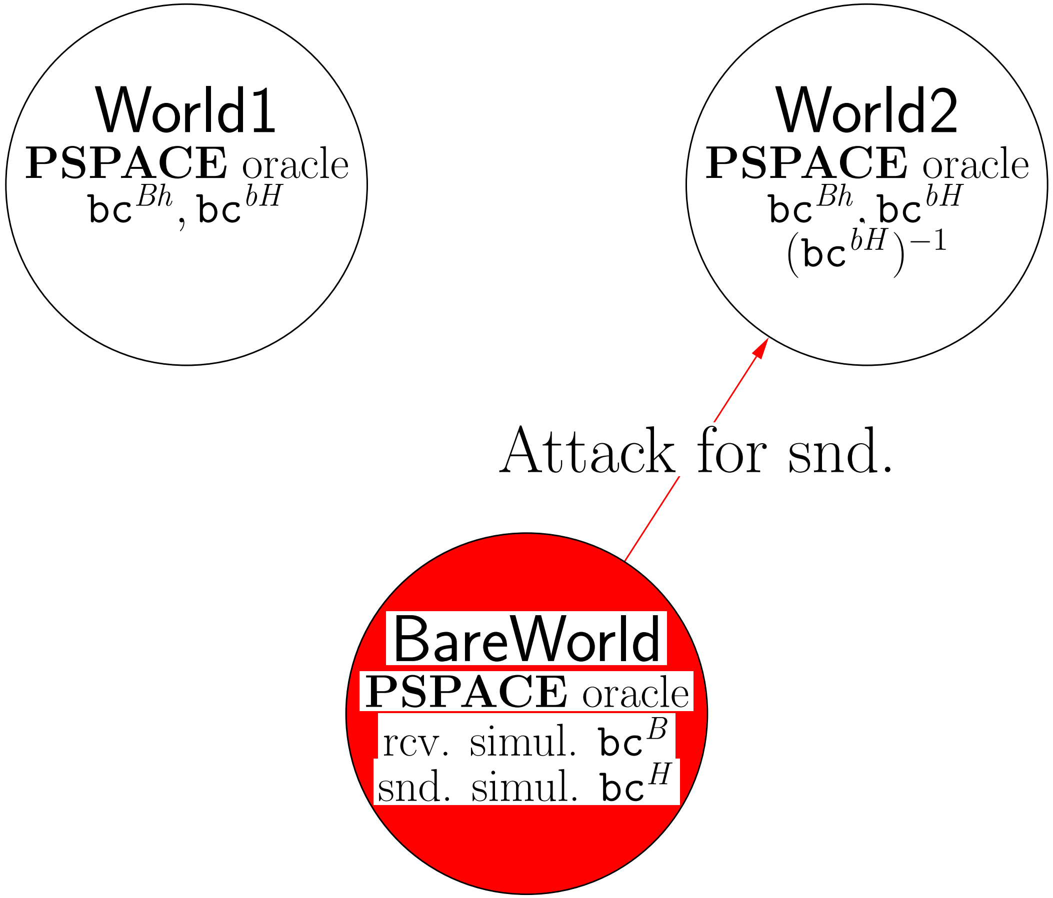
- $\mathtt{bc}^{Bh} : \{0,1\} \times \{0,1\}^n \to \{0,1\}^{2n}$ (no collision)
- $\mathtt{bc}^{bH} : \{0,1\} \times \{0,1\}^n \to \{0,1\}^n$ (two random strings per c)
  $\forall c \in \{0,1\}^n \exists r_0, r_1 \in \{0,1\}^n : r_0 \neq r_1 \wedge \mathtt{bc}^{bH}(0, r_0) = \mathtt{bc}^{bH}(1, r_1) = c$

No transparent black-box $(1,2)$-combiner for BC, illustration



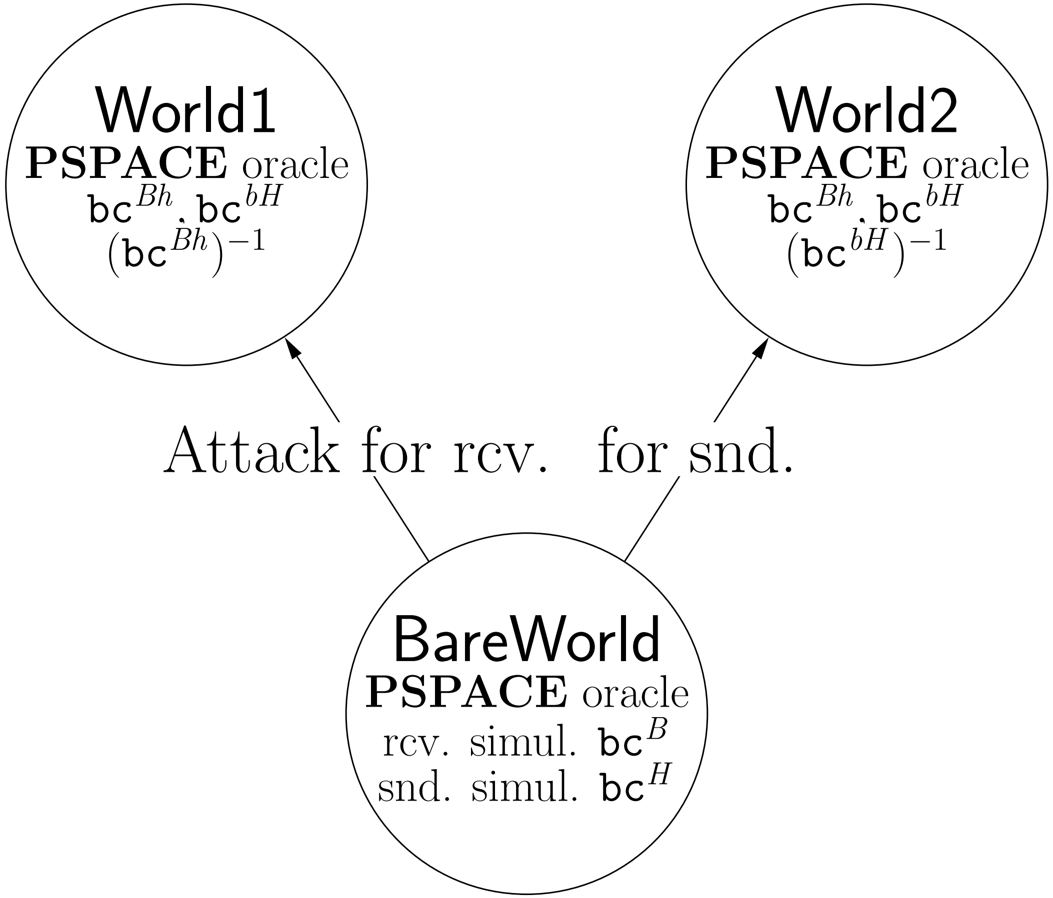
- $\mathtt{bc}^{Bh} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$ (no collision)
- $\mathtt{bc}^{bH} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{n}$ (two random strings per c)
  $\forall c \in \{0,1\}^n \exists r_0, r_1 \in \{0,1\}^n : r_0 \neq r_1 \wedge \mathtt{bc}^{bH}(0, r_0) = \mathtt{bc}^{bH}(1, r_1) = c$

No transparent black-box $(1,2)$-combiner for BC, illustration



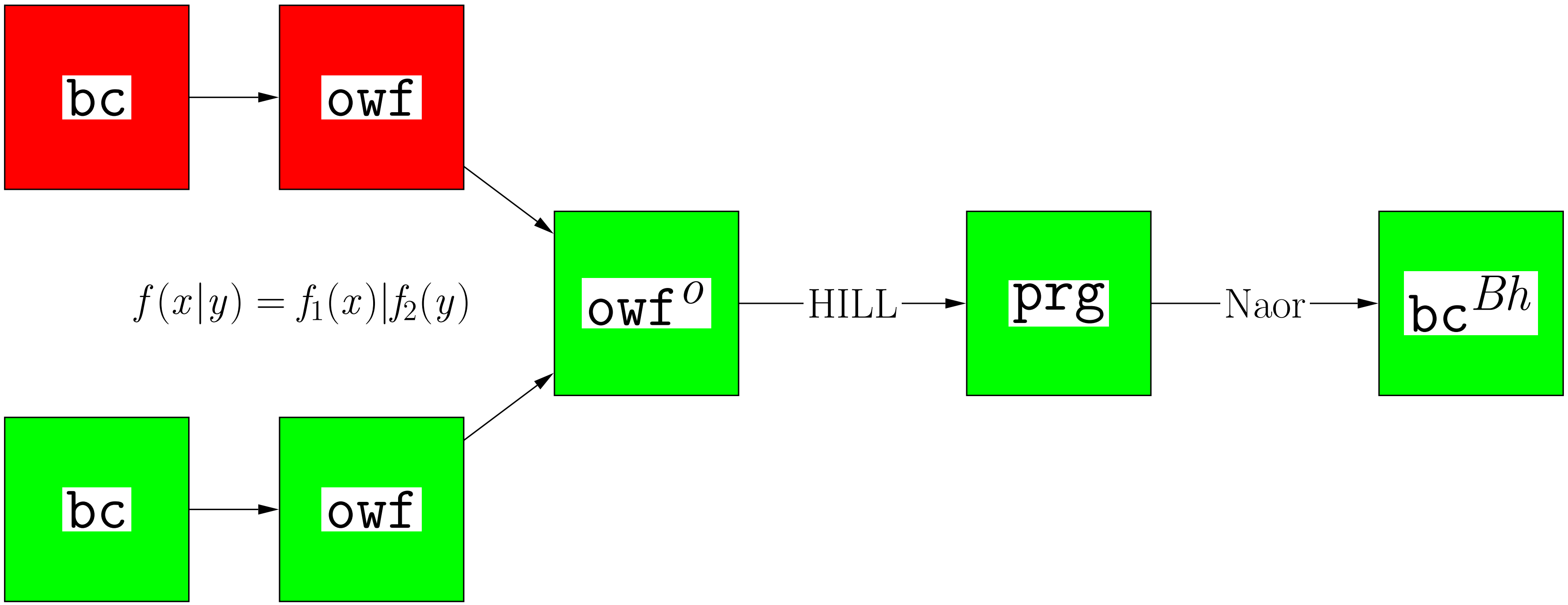
- $\mathtt{bc}^{Bh} : \{0,1\} \times \{0,1\}^n \to \{0,1\}^{2n}$ (no collision)
- $\mathtt{bc}^{bH} : \{0,1\} \times \{0,1\}^n \to \{0,1\}^n$ (two random strings per c)
  $\forall c \in \{0,1\}^n \exists r_0, r_1 \in \{0,1\}^n : r_0 \neq r_1 \wedge \mathtt{bc}^{bH}(0, r_0) = \mathtt{bc}^{bH}(1, r_1) = c$

No transparent black-box $(1,2)$-combiner for BC, illustration



- $\mathtt{bc}^{Bh} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$ (no collision)
- $\mathtt{bc}^{bH} : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{n}$ (two random strings per c)
  $\forall c \in \{0,1\}^n \exists r_0, r_1 \in \{0,1\}^n : r_0 \neq r_1 \wedge \mathtt{bc}^{bH}(0, r_0) = \mathtt{bc}^{bH}(1, r_1) = c$

("efficient") black-box combiner for BC



Summary, BC-combiners

- Third-party black-box
  - Easy if majority of input implementations is secure
  - Easy if we know which player to protect

- Transparent black-box
  - $(1,2)$ combiner impossible

- Black-box
  - $(1,2)$ combiner through $\mathbf{OWF}$

Thank you!

Questions?