# Streebog as a Random Oracle

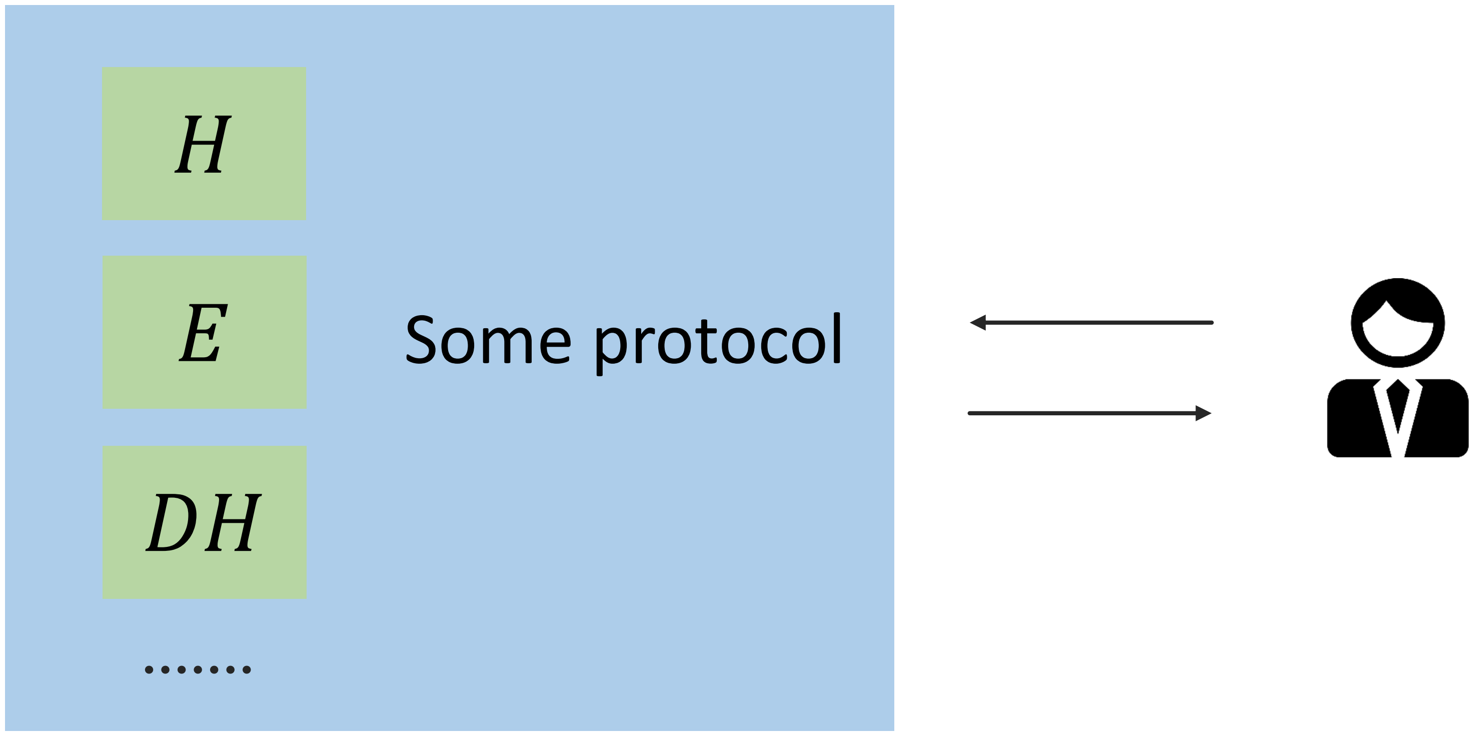
**Bozhko A.**, Akhmetzyanova L., Babueva A.

CryptoPro LLC

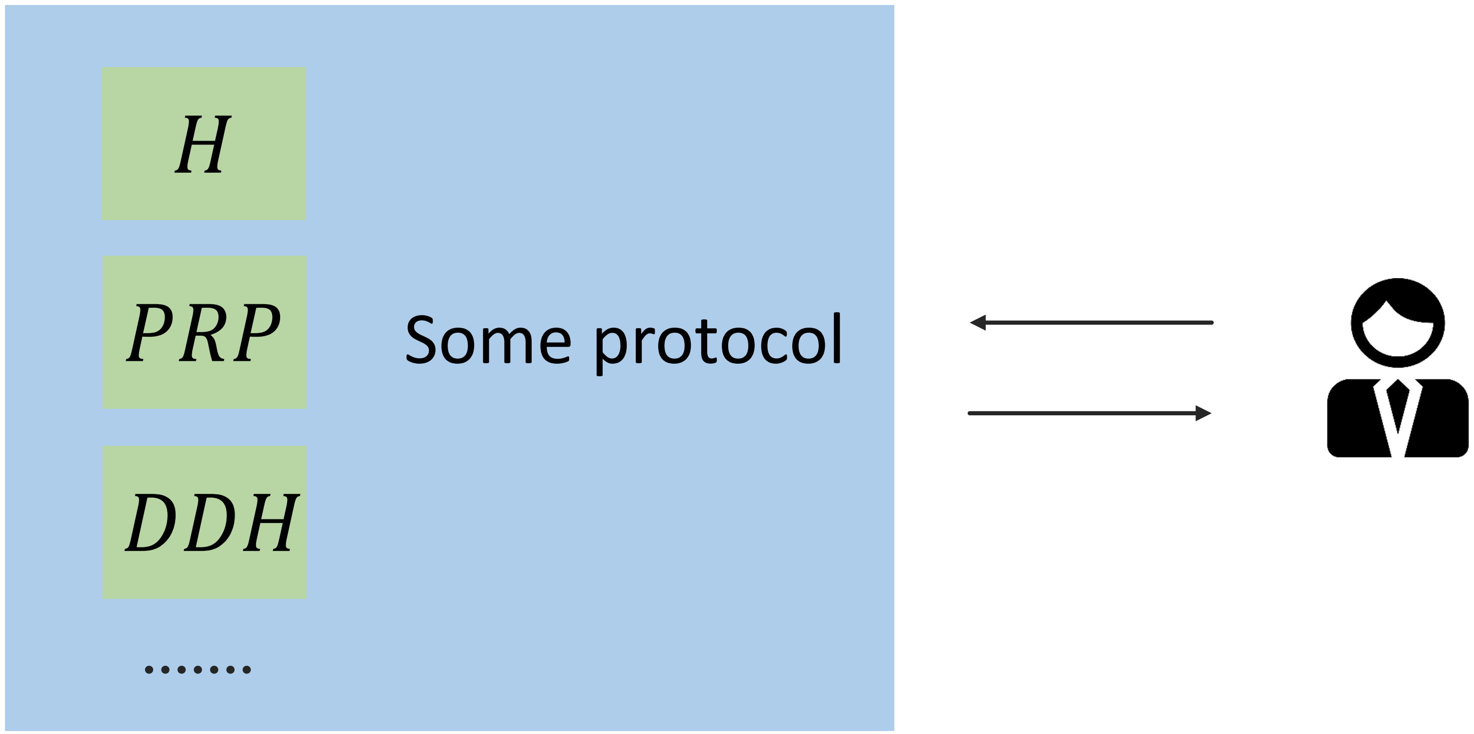
CRYPTOPRO

CTCrypt'2023

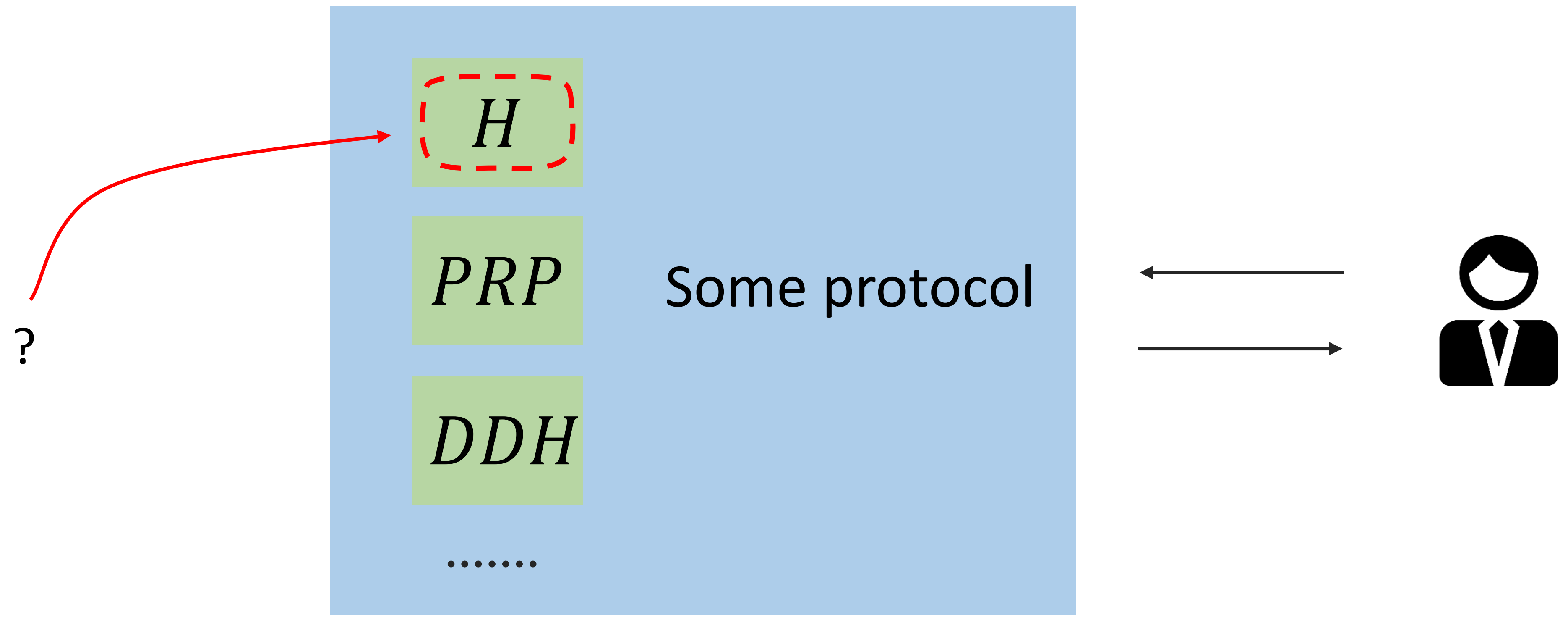
# Random Oracle model



# Random Oracle model

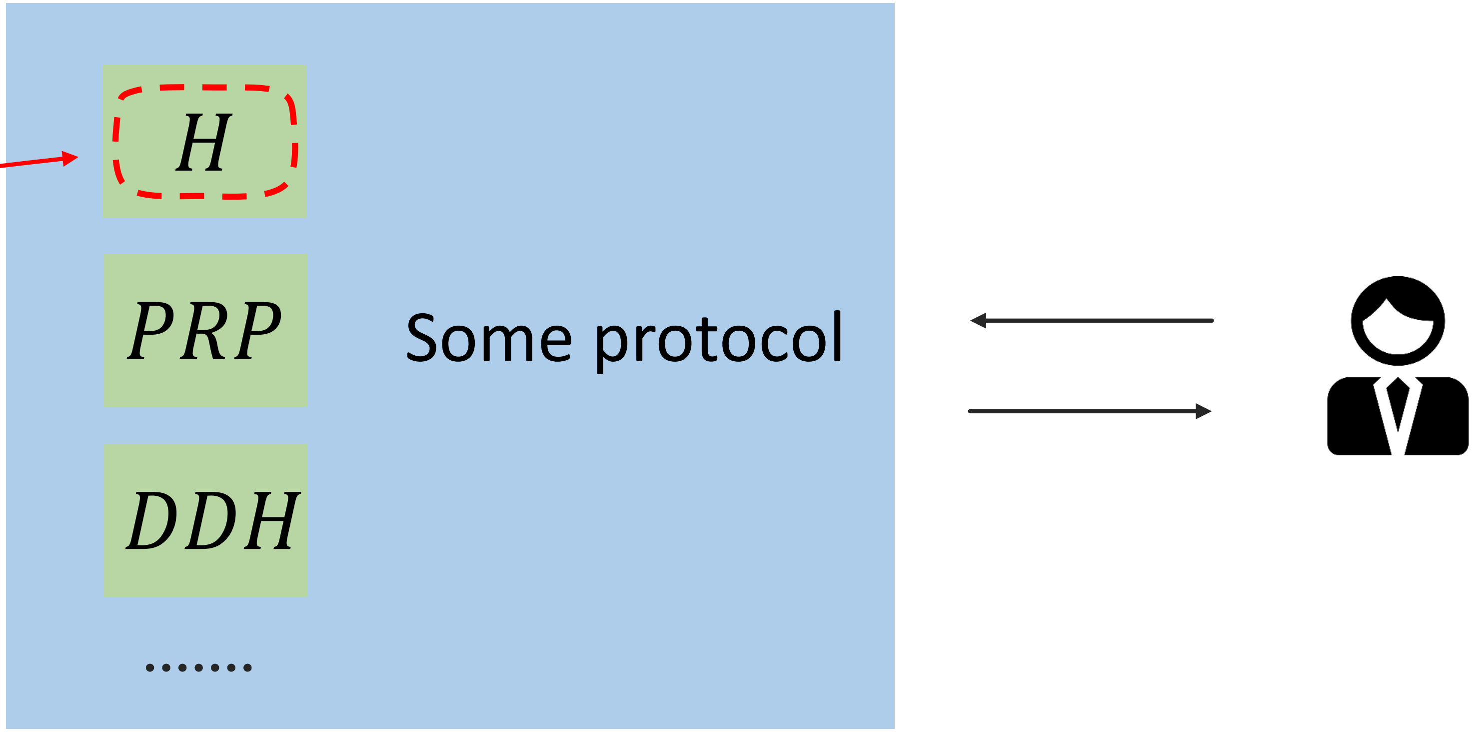


# Random Oracle model



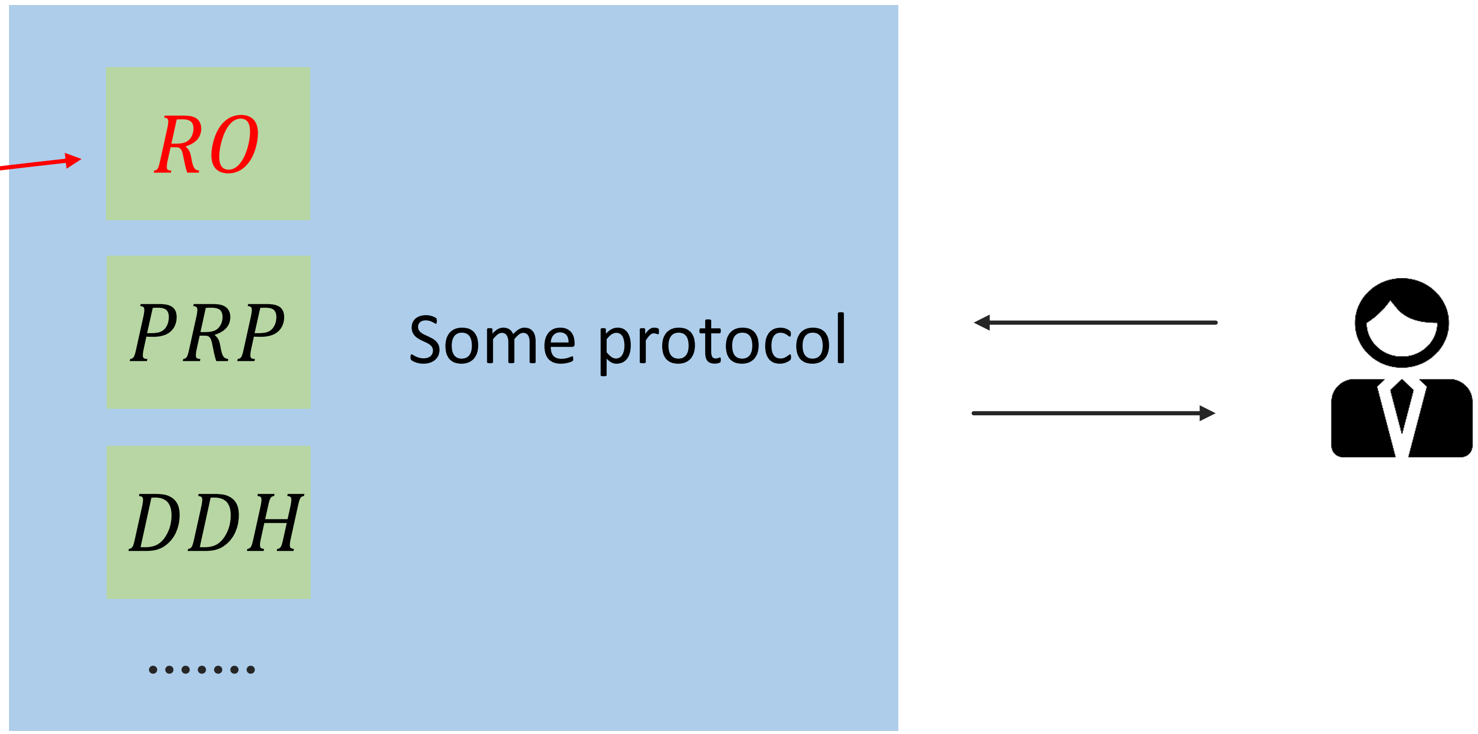
# Random Oracle model

Usual hash function properties might not be sufficient to achieve (or prove) security

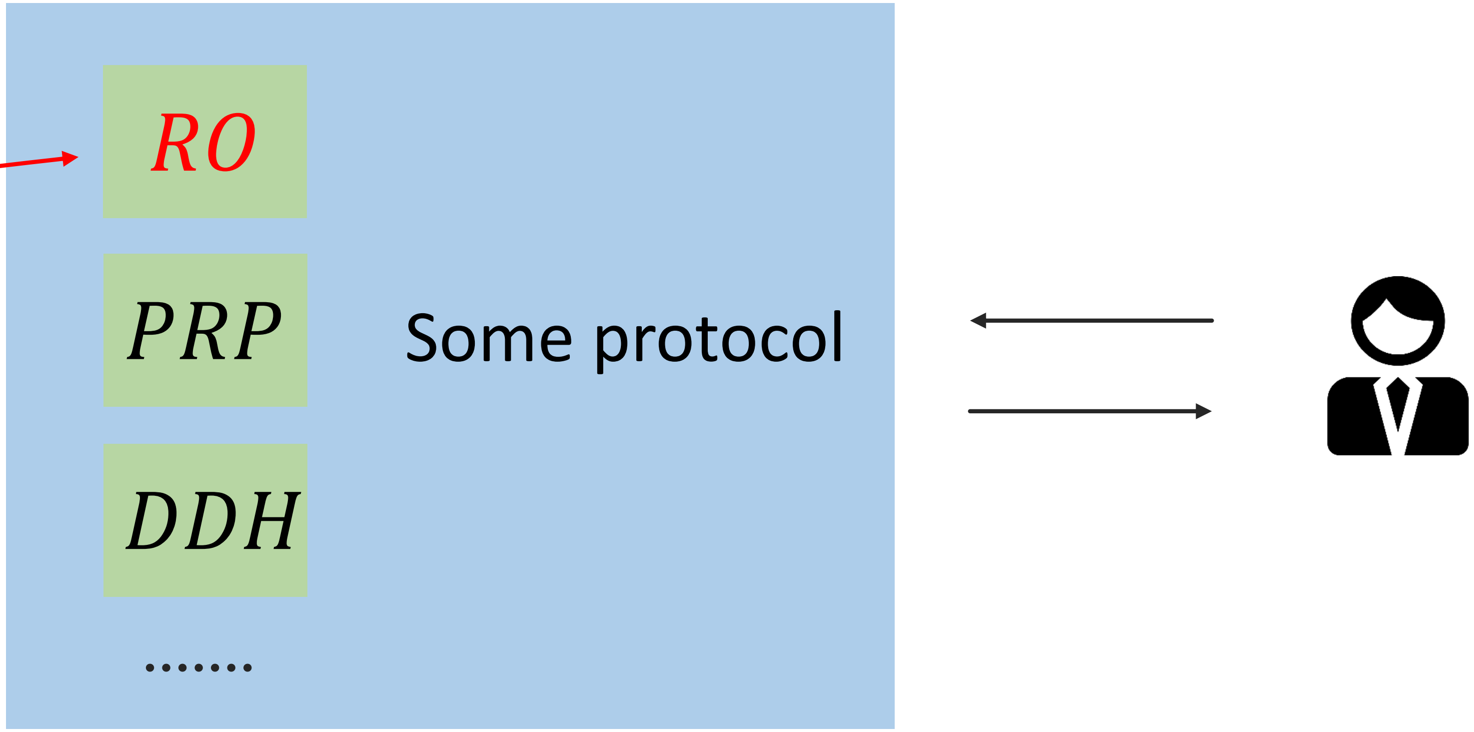


# Random Oracle model

Answers with a random string on every unique query



# Random Oracle model




Answers with a random string on every unique query

Analysis in a random oracle model shows that there are no structural flaws in the protocol

# Random Oracle model

$RO$

$PRP$

$DDH$

.......

TLS
IPSec
Schnorr sign.
SESPAKE
RSBS blind sign.
Shipovnik

Answers with a random string on every unique query

Analysis in a random oracle model shows that there are no structural flaws in the protocol

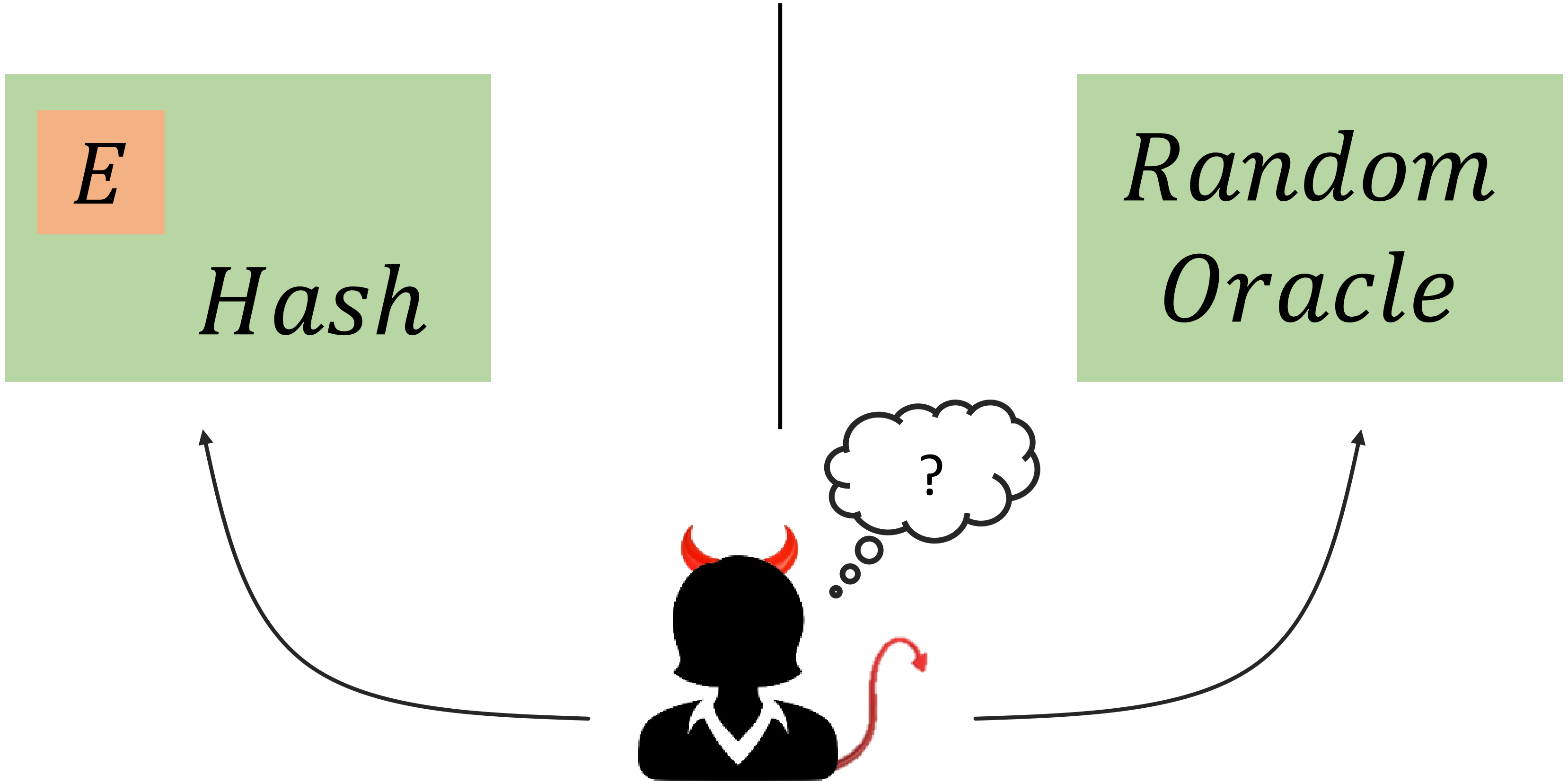
Is there at least one function which is a Random Oracle?

Is there at least one function which is a Random Oracle?

No.

But some hash functions “behave” like them

Behaves like a Random Oracle




Can distinguish easily, since the
distinguisher can compute $Hash$ itself

# Idealized primitives

$IC$

Family of random permutations indexed by a key

$IC(K,P) \rightarrow C$ $\quad IC^{-1}(K,C) \rightarrow P$

| | |
|---|---|
| $K_1$ | $\pi_1 \sim \mathcal{U}(Perm(n))$ |
| $K_2$ | $\pi_2 \sim \mathcal{U}(Perm(n))$ |
| $K_3$ | $\pi_3 \sim \mathcal{U}(Perm(n))$ |
| … | … |

$Random\ Oracle$

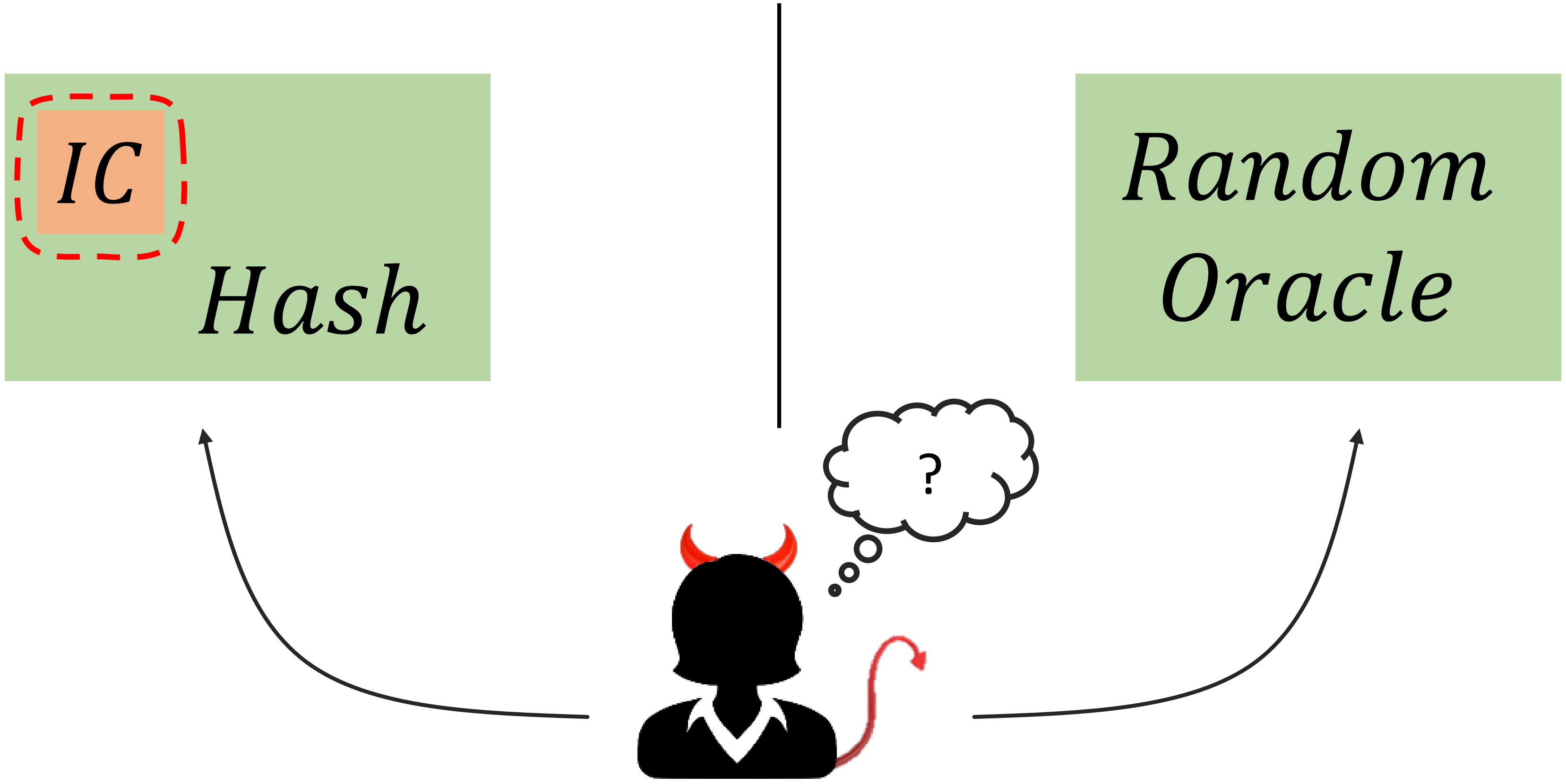
Returns a random string on every unique input

$IC(x) \rightarrow y$

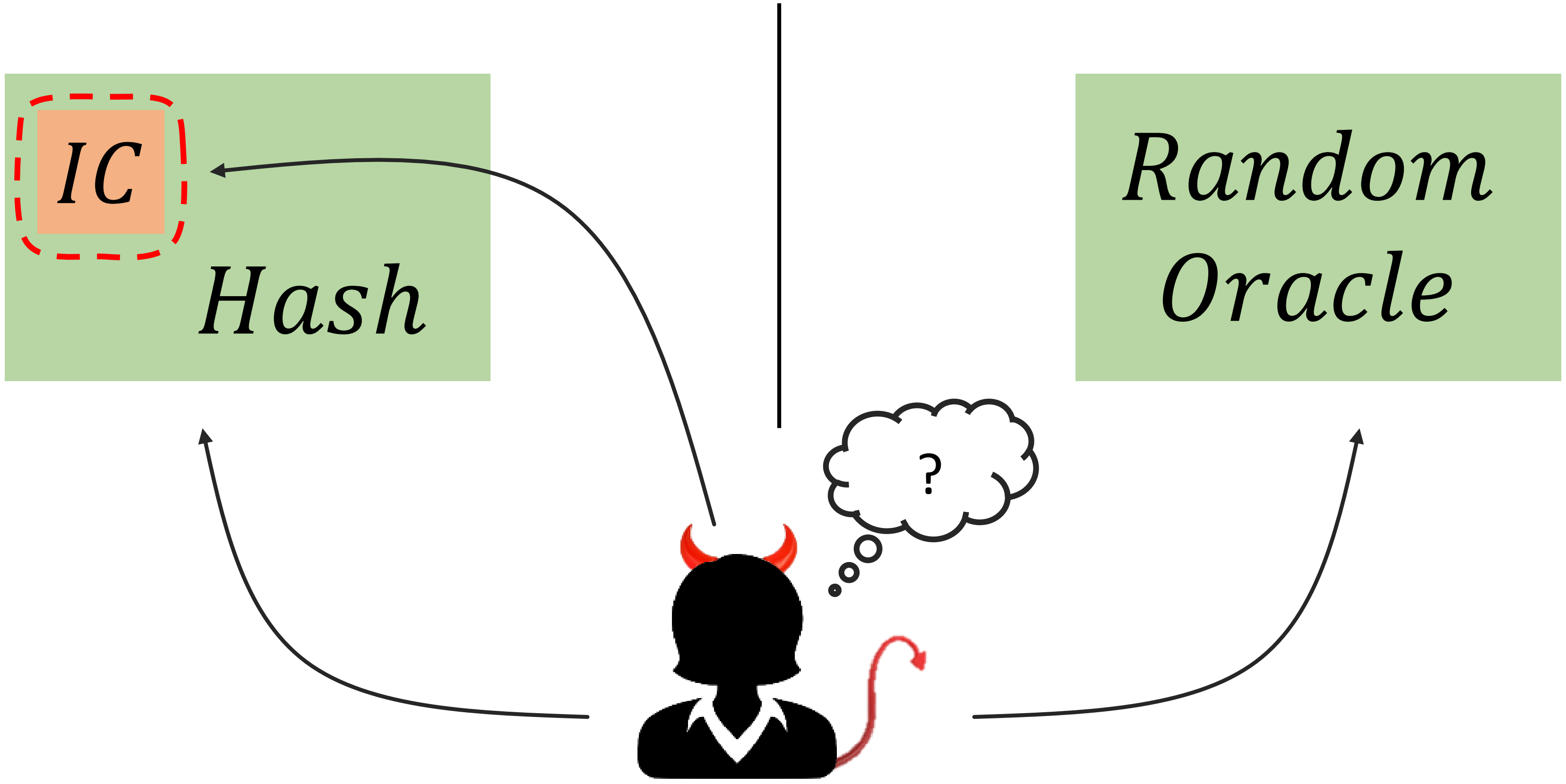
| | |
|---|---|
| $x_1$ | $y_1 \sim \mathcal{U}(\{0,1\}^n)$ |
| $x_2$ | $y_2 \sim \mathcal{U}(\{0,1\}^n)$ |
| $x_3$ | $y_3 \sim \mathcal{U}(\{0,1\}^n)$ |
| … | … |

$\mathcal{U}(A)$ – uniform distribution on a set $A$
$Perm(n)$ – set of all permutations on $\{0,1\}^n$

Behaves like a Random Oracle

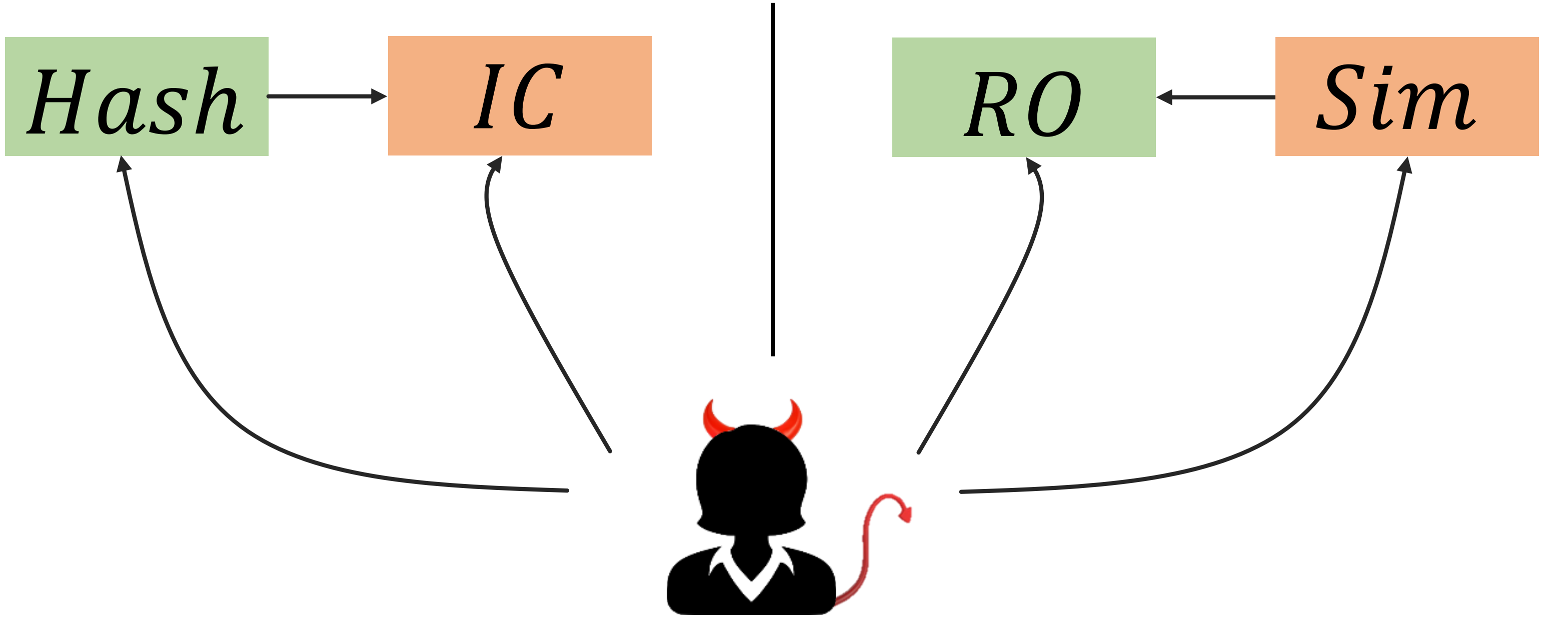


Behaves like a Random Oracle



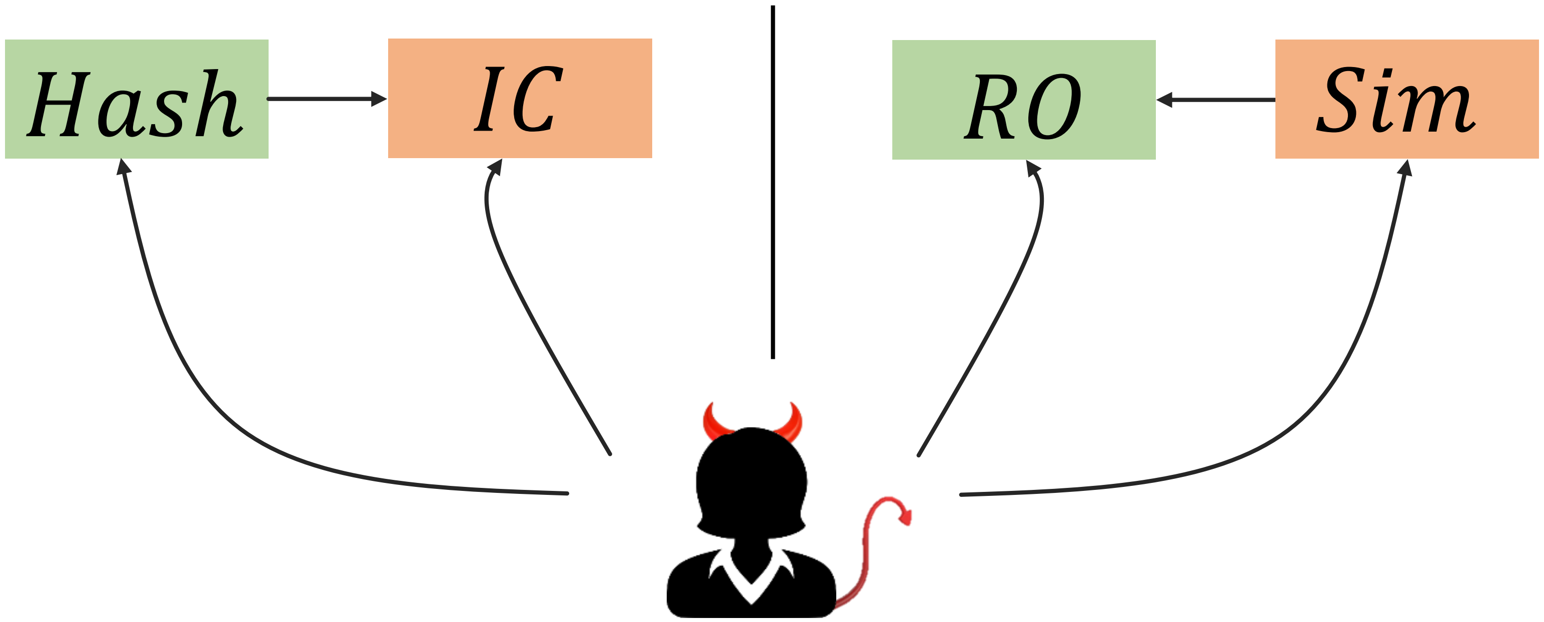

It's fair to assume, that the distinguisher has an access to IC, but we need something on the right side too

Indifferentiability – the way to formalize “behaves like”




There exists an algorithm $Sim$, called simulator, which imitates ideal cipher in such a way, that distinguisher can’t tell apart the world where it interacts with real Hash and IC and  the world where it interact with RO and simulator.

# Indifferentiability – the way to formalize “behaves like”



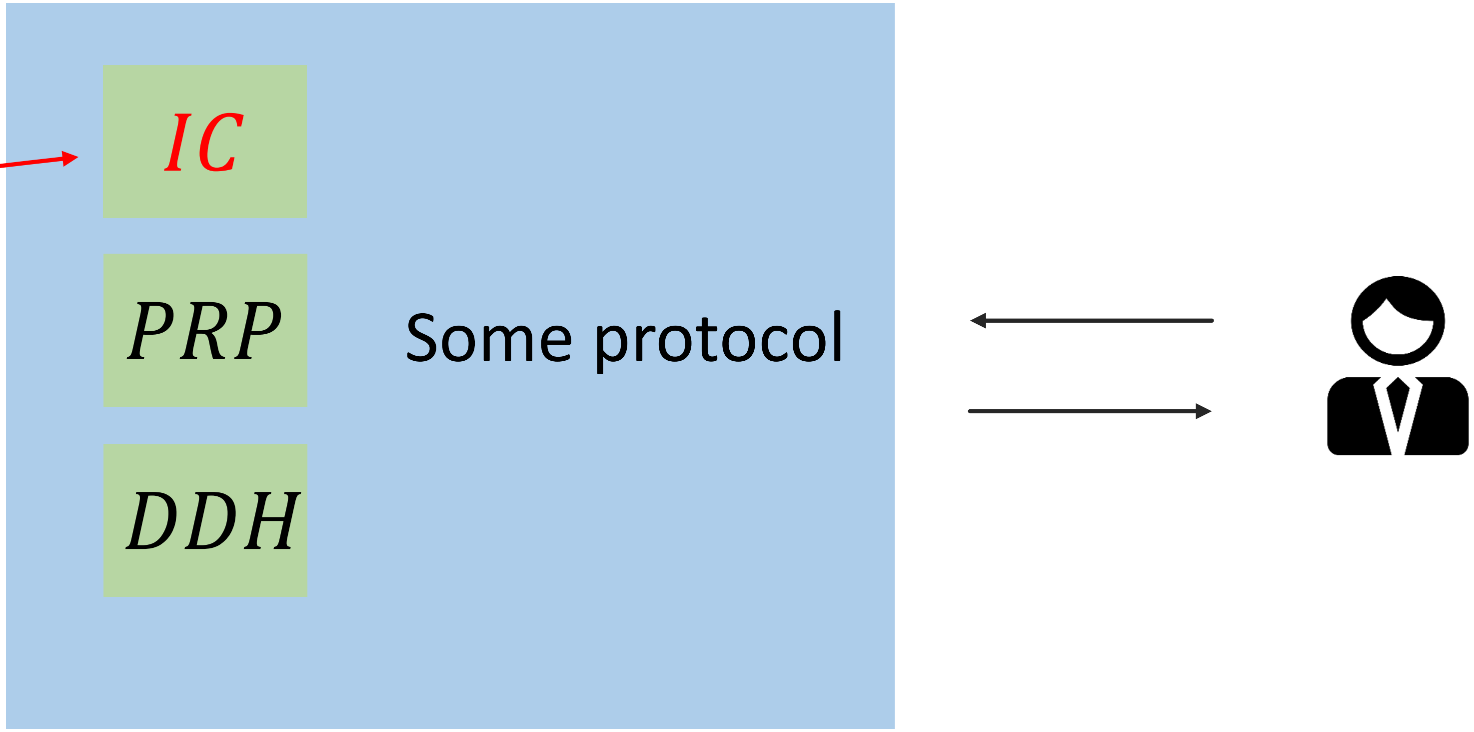

**Definition.** A hash function $Hash$ with oracle access to an ideal cipher $IC$ is said to be $(\varepsilon, q_H, q_E)$-indifferentiable from a random oracle $RO$ if there exists a simulator $Sim$, such that for any distinguisher $\mathcal{D}$ with binary output it holds that:

$$\left|\Pr[\mathcal{D}^{Hash,IC} \to 1] - \Pr[\mathcal{D}^{RO,Sim} \to 1]\right| < \varepsilon$$

The distinguisher and makes at most $q_H$ and $q_{IC}$ queries to its oracles.

## Behaves like a Random Oracle

The protocol is secure in an Ideal Cipher model which is a lower level assumption



# Indifferentiability

- SHA3 – built with Random Oracle model in mind
- SHA256 – not a Random Oracle in general, but is a RO for prefix-free messages
- Davis-Meyer Merkle-Damgard – is a Random Oracle if uses prefix-free encoding

# Indifferentiability

- SHA3 – built with Random Oracle model in mind
- SHA256 – not a Random Oracle in general, but is a RO for prefix-free messages
- Davis-Meyer Merkle-Damgard – is a Random Oracle if uses prefix-free encoding

For every message $m$ an input of MD construction is some string $g(m)$, called encoding of $m$:

$$H(m) = MD\big(g(m)\big),$$

$$MD(x_1||x_2|| \ldots ||x_l) = h(\ldots(h(h(IV, x_1), x_2), \ldots), x_l),$$

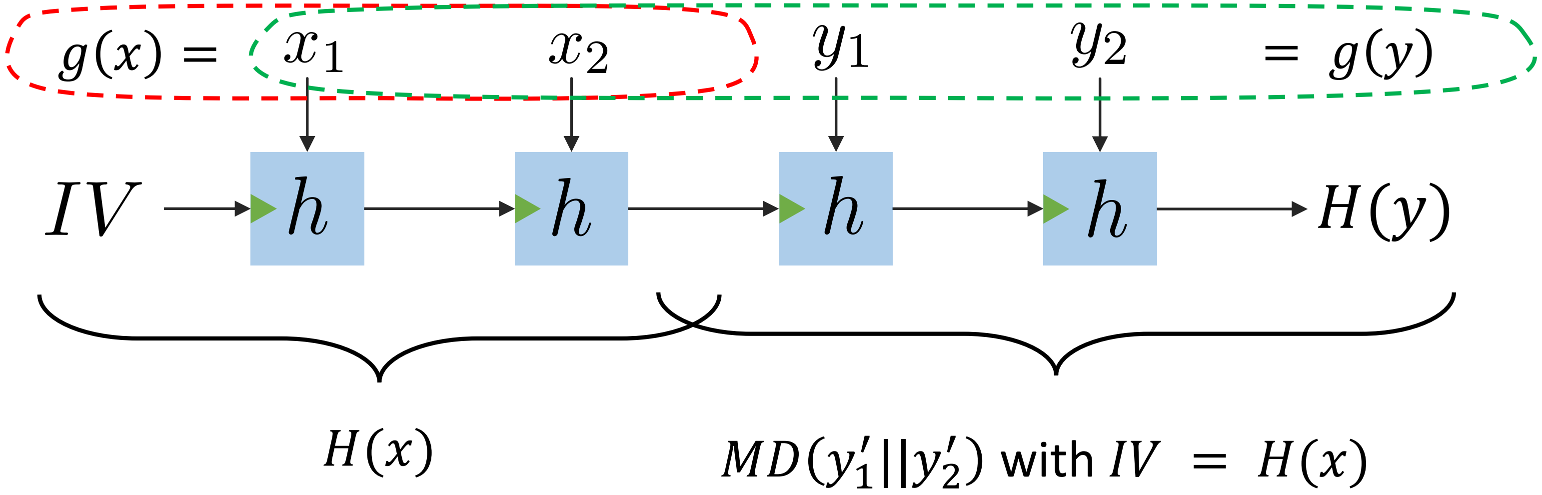
where $h$ is a compression function and encoding is called prefix-free iff for every two strings $a, b$ it is guaranteed that $g(a)$ is not a prefix of $g(b)$

CRYPTOPRO

# What if encoding is not prefix-free?

**Length extension attacks:**

- Encoding is not prefix-free $\Rightarrow$ there exist two inputs $x, y$ such that $g(x) = x_1 || \ldots || x_l$ and $g(y) = g(x) || y_1' || \ldots || y_m'$
- Query $m_0 = H(x)$
- Compute $m_1' = MD(y_1' || \ldots || y_m')$ by making IC queries to compute compression function $h$ and using $m_0$ as an IV
- Query $m_1 = H(y)$
- If $m_1 = m_1'$, we are in the real world, else in the Random Oracle world




$\Rightarrow MD(y_1' || \ldots || y_m') = H(y)$

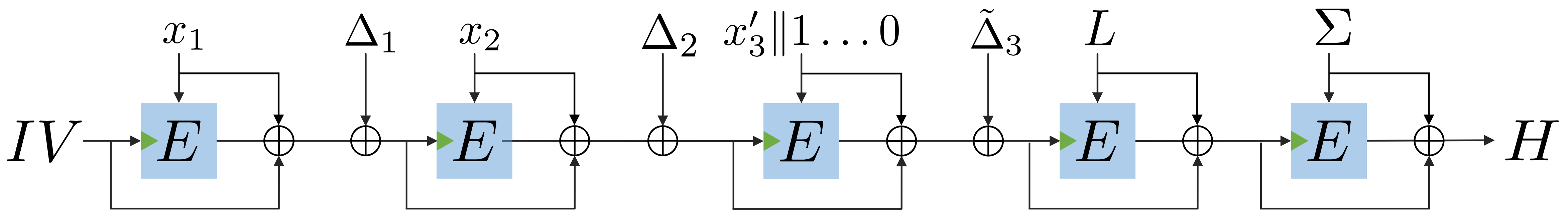
in real world!

# Indifferentiability

- SHA3 – built with Random Oracle model in mind
- SHA256 – not a Random Oracle in general, but is a RO for prefix-free messages
- Davis-Mayer Merkle-Damgard – is a Random Oracle if uses prefix-free encoding
- **Streebog – ???**

# Streebog*

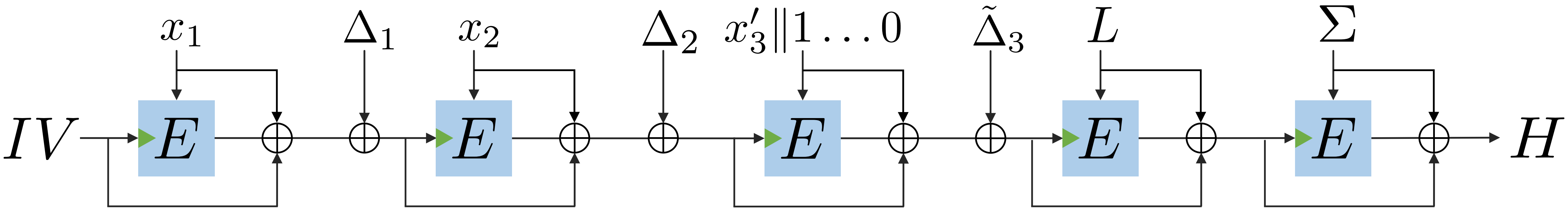
$$Streebog(x = x_1||x_2||x_3') \to H$$




$$\Delta_i = i \cdot n \oplus (i-1) \cdot n$$
$$\widetilde{\Delta}_i = i \cdot n \oplus (i-1) \cdot n$$
$$L = len(x)$$
$$\Sigma = x_1 + x_2 + x_3'||1 \ldots 0$$

*equivalent representation, Guo J., Jean J., Leurent G., Peyrin T., Wang L., “The Usage of Counter Revisited: Second-Preimage Attack on New Russian Standardized Hash Function”

# Streebog as a Random Oracle



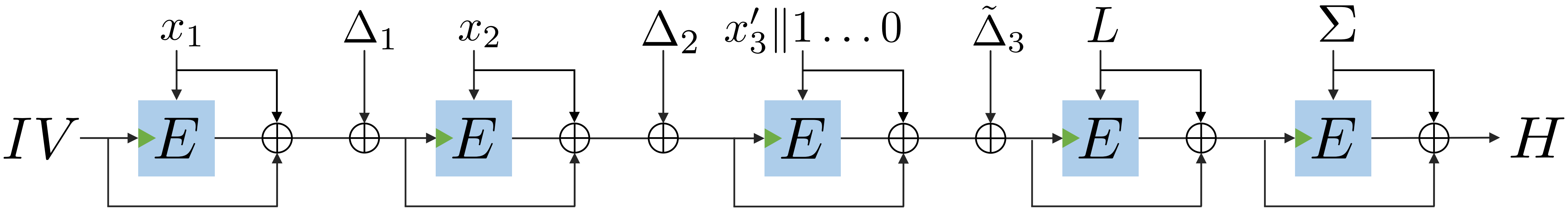

We can't use the classic DM-MD with prefix-free encoding result because it is not a classical MD construction due to deltas, also a Miyaguchi-Preneel compression function is used.

But we notice that the following function:

$$g(x) = (x_1, \Delta_1)||(x_2, \Delta_2)|| \ldots ||(x_l'||1 \ldots 0, \widetilde{\Delta_l})||(L, 0)||(\Sigma, 0)$$

is prefix-free.

# Streebog as a Random Oracle




We can't use the classic DM-MD with prefix-free encoding result because it is not a classical MD construction due to deltas, also a Miyaguchi-Preneel compression function is used.

But we notice that the following function:

$$g(x) = (x_1, \Delta_1)||(x_2, \Delta_2)|| \ldots ||(x_l'||1 \ldots 0, \widetilde{\Delta_l})||(L, 0)||(\Sigma, 0) \quad \Bigg\} \textbf{ Hope}$$

is prefix-free.

# Streebog as a Random Oracle

**Theorem.** The hash function Streebog with a cipher $E: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is $(\varepsilon, q_H, q_{IC})$-indifferentiable from a random oracle in the ideal cipher model for $E$ with

$$\varepsilon = \frac{q}{2^{n-3}} + \frac{(6 + 4n)q^2}{2^{n-4}} + \frac{q^3}{2^{n-7}}$$

where $q = q_{IC} + q_H(l_m + 2)$ and $l_m$ is the maximum message length (in blocks, including padding) queried by the distinguisher to its left oracle.

The proof follows the structure of the prefix-free Davis-Meyer Merkle-Damgard case proof and is mainly combinatorial.

# Streebog as a Random Oracle

**Theorem.** The hash function Streebog with a cipher $E: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is $(\varepsilon, q_H, q_{IC})$-indifferentiable from a random oracle in the ideal cipher model for $E$ with

$$\varepsilon = \frac{q}{2^{n-3}} + \frac{(6 + 4n)q^2}{2^{n-4}} + \boxed{\frac{q^3}{2^{n-7}}}$$

where $q = q_{IC} + q_H(l_m + 2)$ and $l_m$ is the maximum message length (in blocks, including padding) queried by the distinguisher to its left oracle.

The proof follows the structure of the prefix-free Davis-Meyer Merkle-Damgard case proof and is mainly combinatorial.

**Open question:** Is it possible to get rid of the cube?

# Questions?

Contacts:
bozhko@cryptopro.ru