OT Extension

CS 598 DH

Today’s objectives

Introduce notion of OT extension

Show feasibility of OT extension

Construct efficient OT extension

Setting

Semi-honest Security

Malicious Security

Zero Knowledge

General-Purpose Tools

GMW Protocol

- Multi-party
- Multi-round

Garbled Circuit

- Constant Round
- Two Party

Primitives

Oblivious Transfer

Oblivious RAM

Pseudorandom functions/encryption

Commitments

Setting

Semi-honest Security

Malicious Security

Zero Knowledge

General-Purpose Tools

GMW Protocol

- Multi-party
- Multi-round

Garbled Circuit

- Constant Round
- Two Party

Today

Primitives

Oblivious Transfer

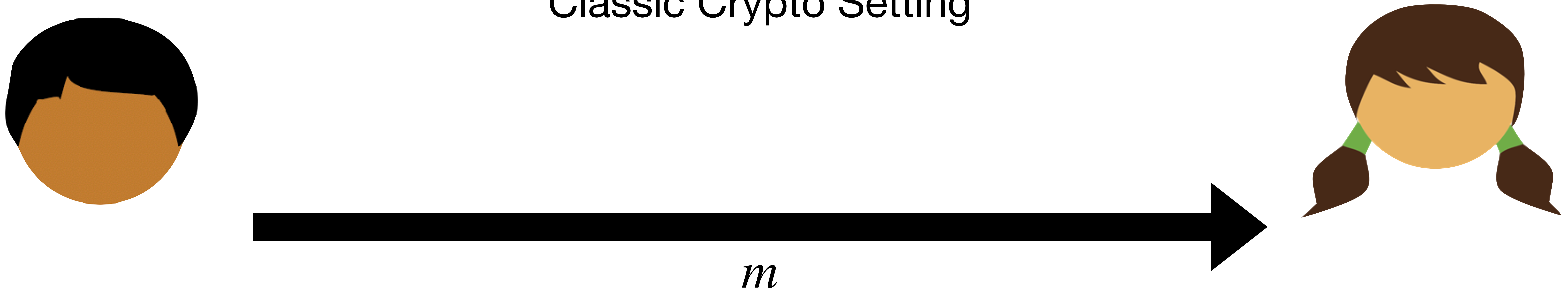
Oblivious RAM

Pseudorandom functions/encryption

Commitments

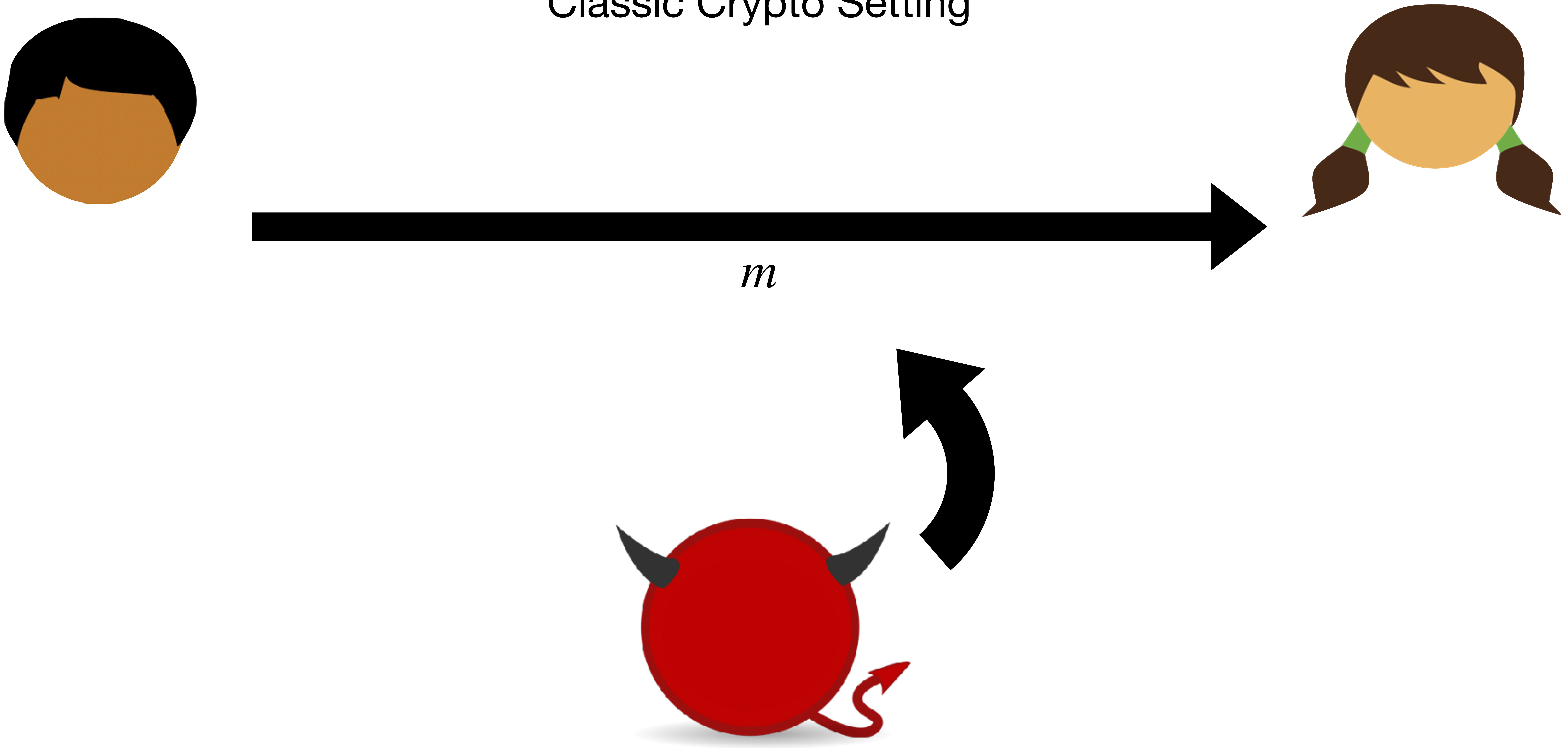
Public Key $\gg$ Symmetric Key $\gg$ Linear Operations

Classic Crypto Setting

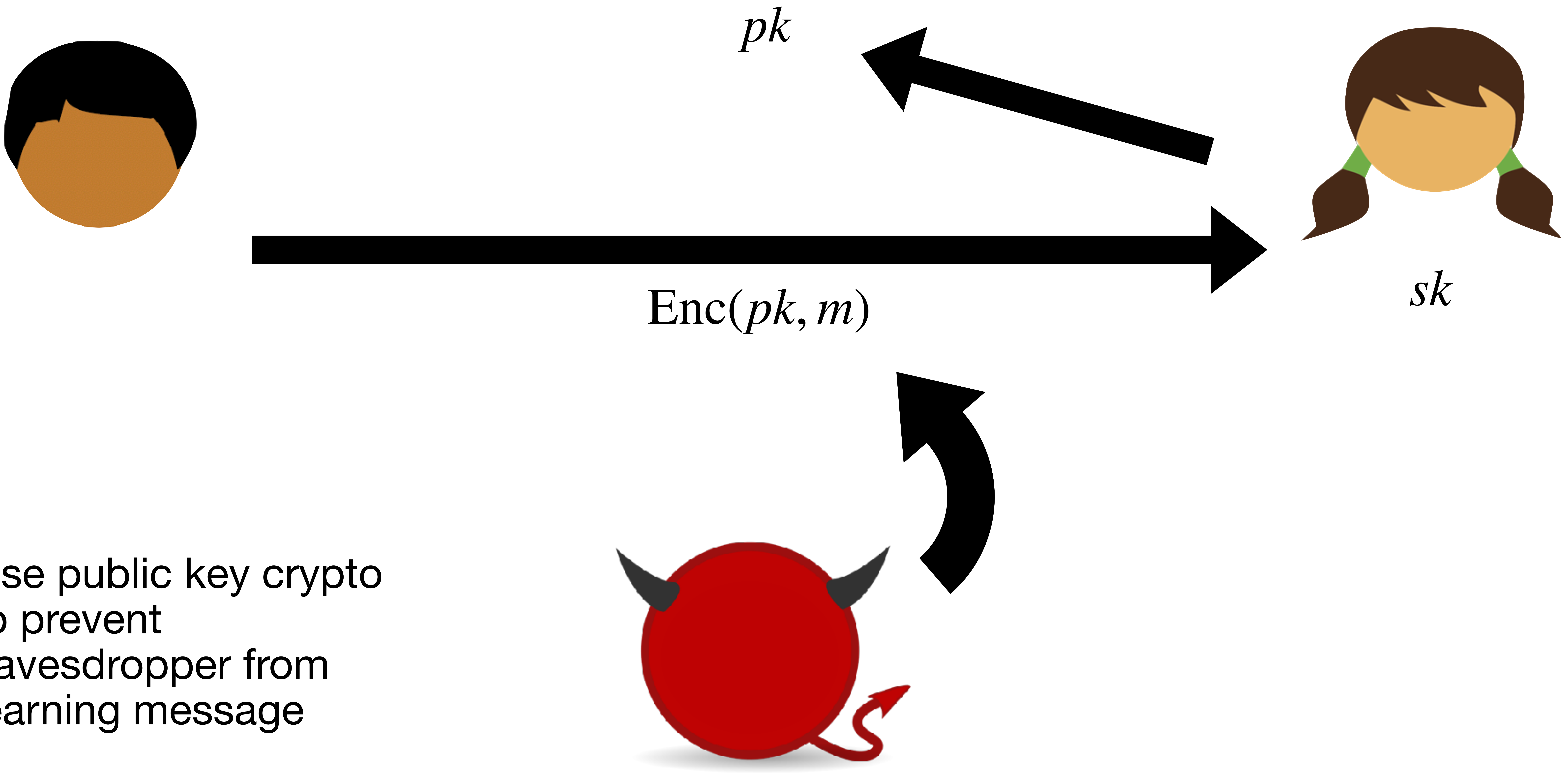


Public Key $\gg$ Symmetric Key $\gg$ Linear Operations

Classic Crypto Setting

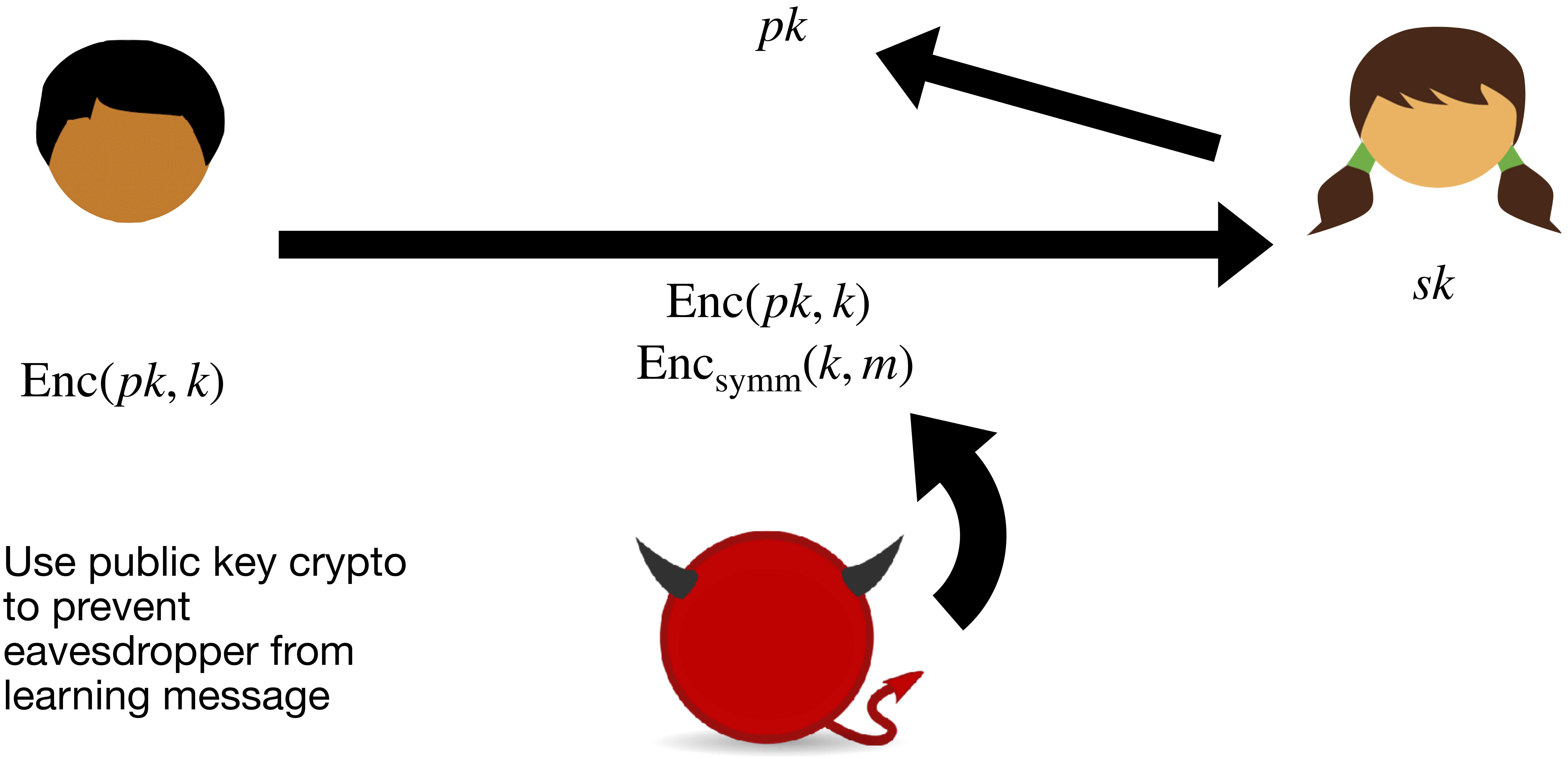


Public Key $\gg$ Symmetric Key $\gg$ Linear Operations



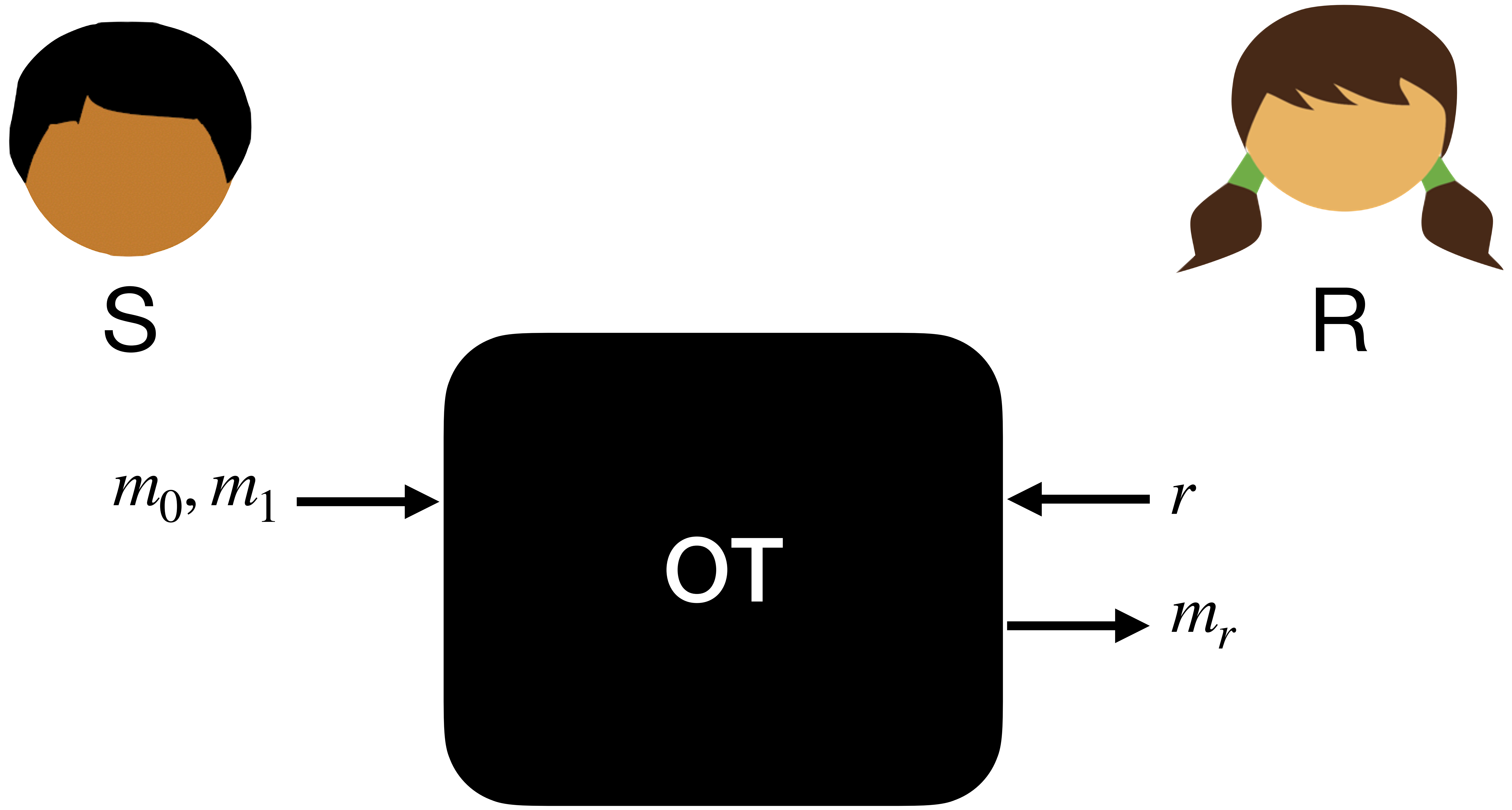
Use public key crypto to prevent eavesdropper from learning message

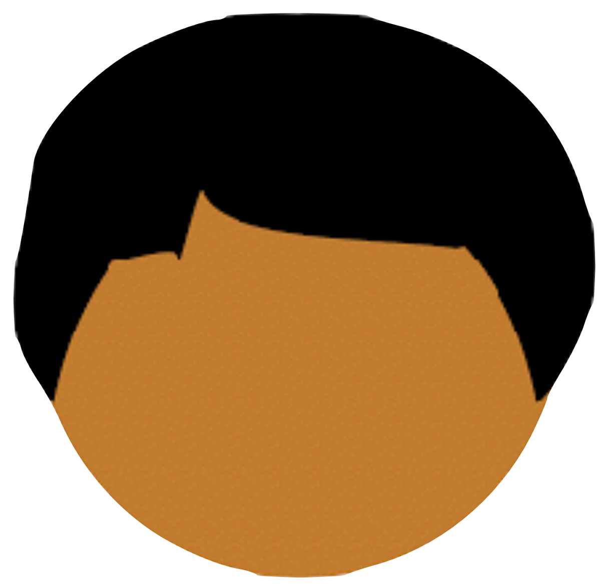
Public Key $\gg$ Symmetric Key $\gg$ Linear Operations



Use public key crypto to prevent eavesdropper from learning message

Public Key $\gg$ Symmetric Key $\gg$ Linear Operations


S
R
m_0, m_1
OT
r
m_r



Sender

m_0, m_1

b

Receiver

$a \xleftarrow{\$} \mathbb{Z}_q$

$h_b \leftarrow g^a$

$h_{1-b} \xleftarrow{\$} G$

$$\xleftarrow{\quad h_0, h_1 \quad}$$

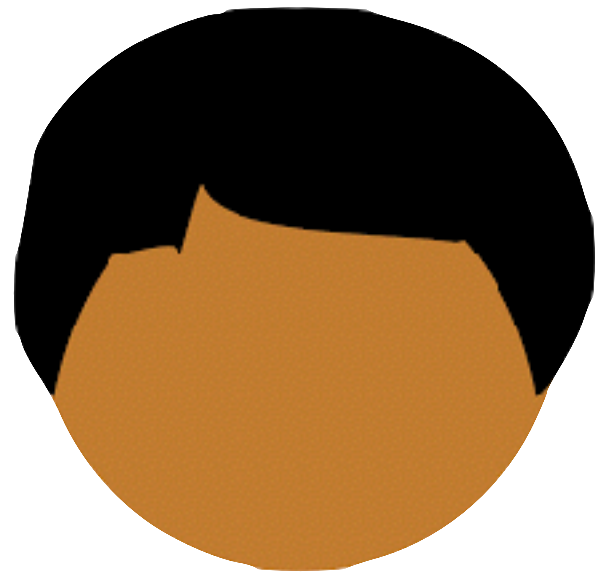
$r_0 \xleftarrow{\$} \mathbb{Z}_q$

$r_1 \xleftarrow{\$} \mathbb{Z}_q$

$$\xrightarrow{\quad g^{r_0} \quad g^{r_1} \quad h_0^{r_0} \cdot m_0 \quad h_1^{r_1} \cdot m_1 \quad}$$

$$\frac{h_b^{r_b} \cdot m_b}{\left(g^{r_b}\right)^a}$$

From long ago…



Sender

m_0, m_1

b

Receiver

$a \xleftarrow{\$} \mathbb{Z}_q$

$h_b \leftarrow g^a$

$h_{1-b} \xleftarrow{\$} G$

$$\xleftarrow{\quad h_0, h_1 \quad}$$

$r_0 \xleftarrow{\$} \mathbb{Z}_q$

$r_1 \xleftarrow{\$} \mathbb{Z}_q$

$$\xrightarrow[\quad]{\begin{matrix} g^{r_0} & g^{r_1} \\ h_0^{r_0} \cdot m_0 & h_1^{r_1} \cdot m_1 \end{matrix}}$$

$$\frac{h_b^{r_b} \cdot m_b}{\left(g^{r_b}\right)^a}$$

From long ago…

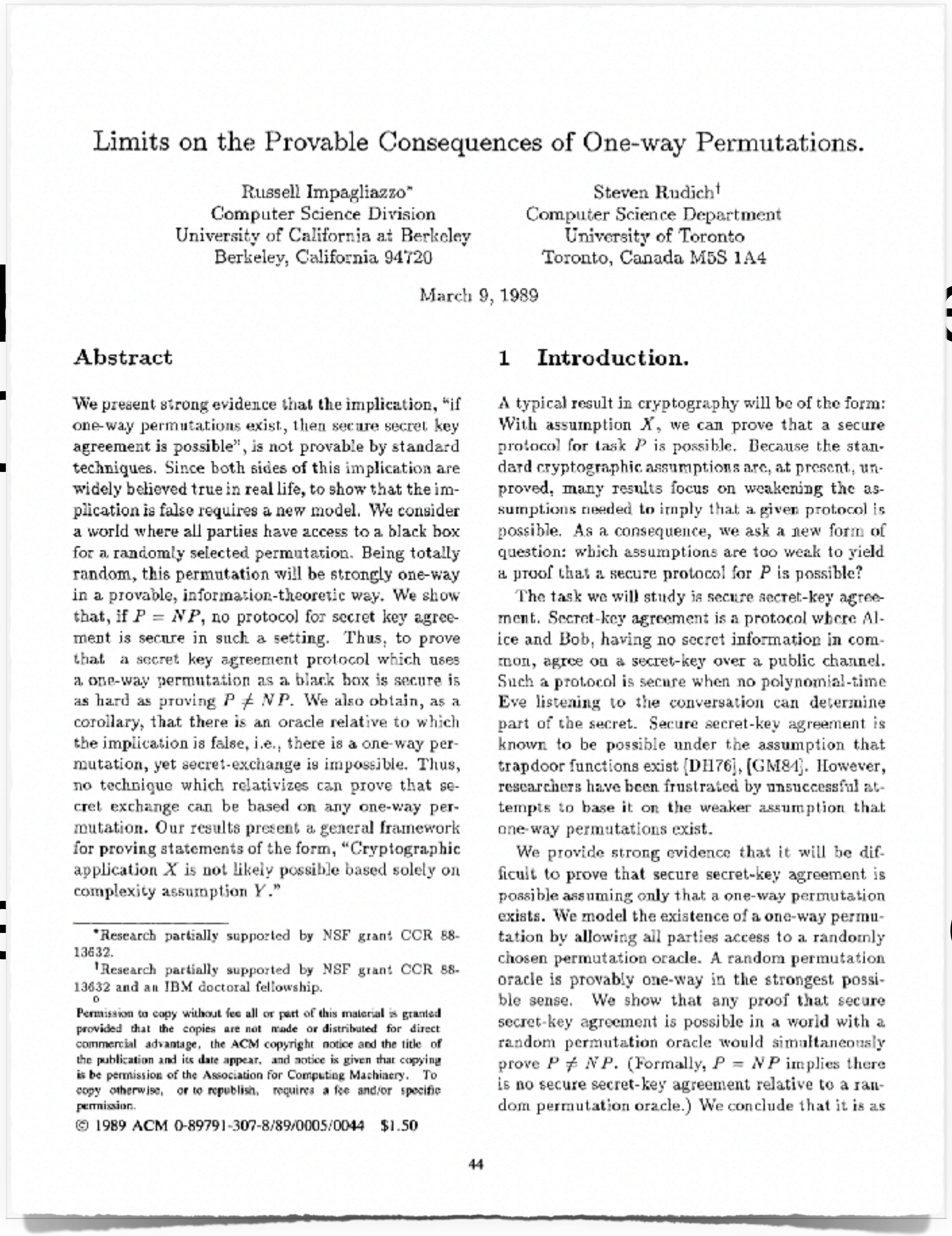
Operations on DDH groups are expensive!

OT is the backbone of MPC techniques (GMW, GC, PSI)

OT requires public key cryptography

OT is the   echniques
(GMW, GC

OT require   graphy



Limits on the Provable Consequences of One-way Permutations.

Russell Impagliazzo[*]
Computer Science Division
University of California at Berkeley
Berkeley, California 94720

Steven Rudich[†]
Computer Science Department
University of Toronto
Toronto, Canada M5S 1A4

March 9, 1989

Abstract

We present strong evidence that the implication, "if one-way permutations exist, then secure secret key agreement is possible", is not provable by standard techniques. Since both sides of this implication are widely believed true in real life, to show that the implication is false requires a new model. We consider a world where all parties have access to a black box for a randomly selected permutation. Being totally random, this permutation will be strongly one-way in a provable, information-theoretic way. We show that, if $P = NP$, no protocol for secret key agreement is secure in such a setting. Thus, to prove that a secret key agreement protocol which uses a one-way permutation as a black box is secure is as hard as proving $P \neq NP$. We also obtain, as a corollary, that there is an oracle relative to which the implication is false, i.e., there is a one-way permutation, yet secret-exchange is impossible. Thus, no technique which relativizes can prove that secret exchange can be based on any one-way permutation. Our results present a general framework for proving statements of the form, "Cryptographic application X is not likely possible based solely on complexity assumption Y."

1 Introduction.

A typical result in cryptography will be of the form: With assumption X, we can prove that a secure protocol for task P is possible. Because the standard cryptographic assumptions are, at present, unproved, many results focus on weakening the assumptions needed to imply that a given protocol is possible. As a consequence, we ask a new form of question: which assumptions are too weak to yield a proof that a secure protocol for P is possible?

The task we will study is secure secret-key agreement. Secret-key agreement is a protocol where Alice and Bob, having no secret information in common, agree on a secret-key over a public channel. Such a protocol is secure when no polynomial-time Eve listening to the conversation can determine part of the secret. Secure secret-key agreement is known to be possible under the assumption that trapdoor functions exist [DH76], [GM84]. However, researchers have been frustrated by unsuccessful attempts to base it on the weaker assumption that one-way permutations exist.

We provide strong evidence that it will be difficult to prove that secure secret-key agreement is possible assuming only that a one-way permutation exists. We model the existence of a one-way permutation by allowing all parties access to a randomly chosen permutation oracle. A random permutation oracle is provably one-way in the strongest possible sense. We show that any proof that secure secret-key agreement is possible in a world with a random permutation oracle would simultaneously prove $P \neq NP$. (Formally, $P = NP$ implies there is no secure secret-key agreement relative to a random permutation oracle.) We conclude that it is as

[*]Research partially supported by NSF grant CCR 88-13632.

[†]Research partially supported by NSF grant CCR 88-13632 and an IBM doctoral fellowship.

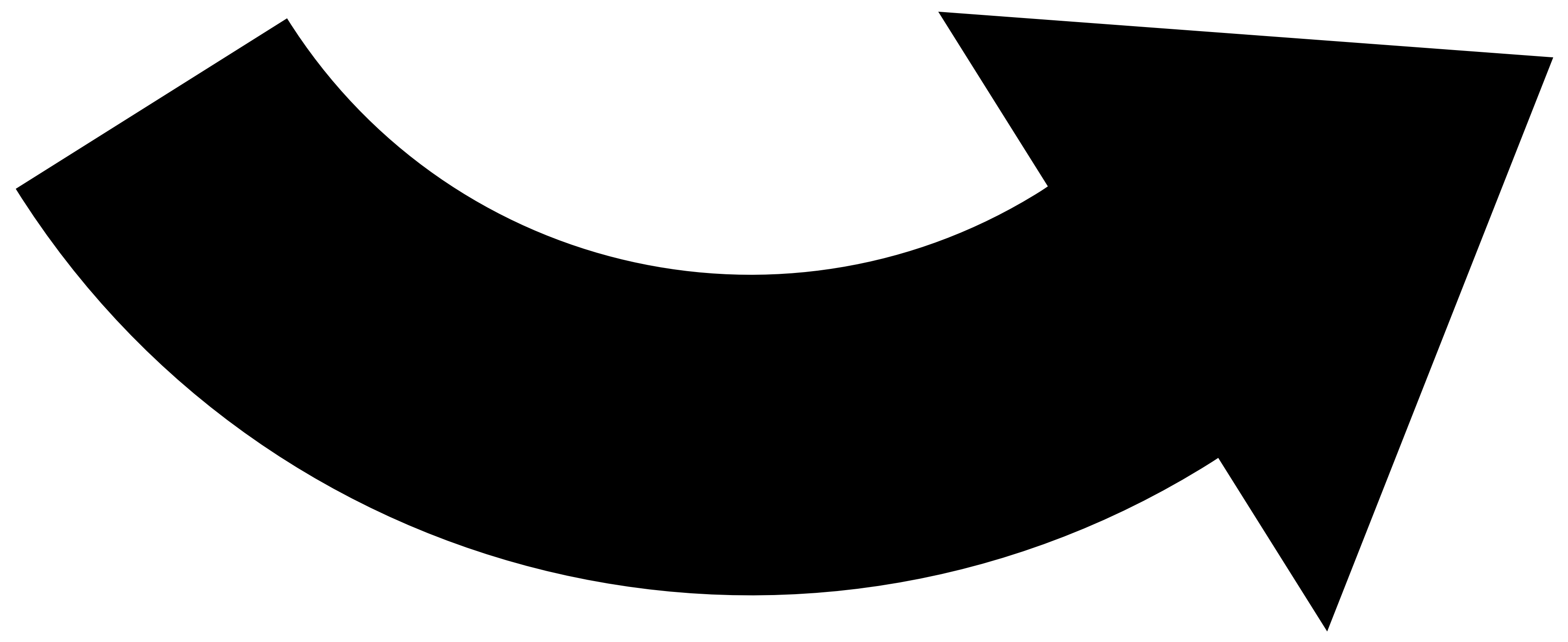
OT requires public key cryptography…

What is OT extension?

Public Key $\gg$ Symmetric Key $\gg$ Linear Operations

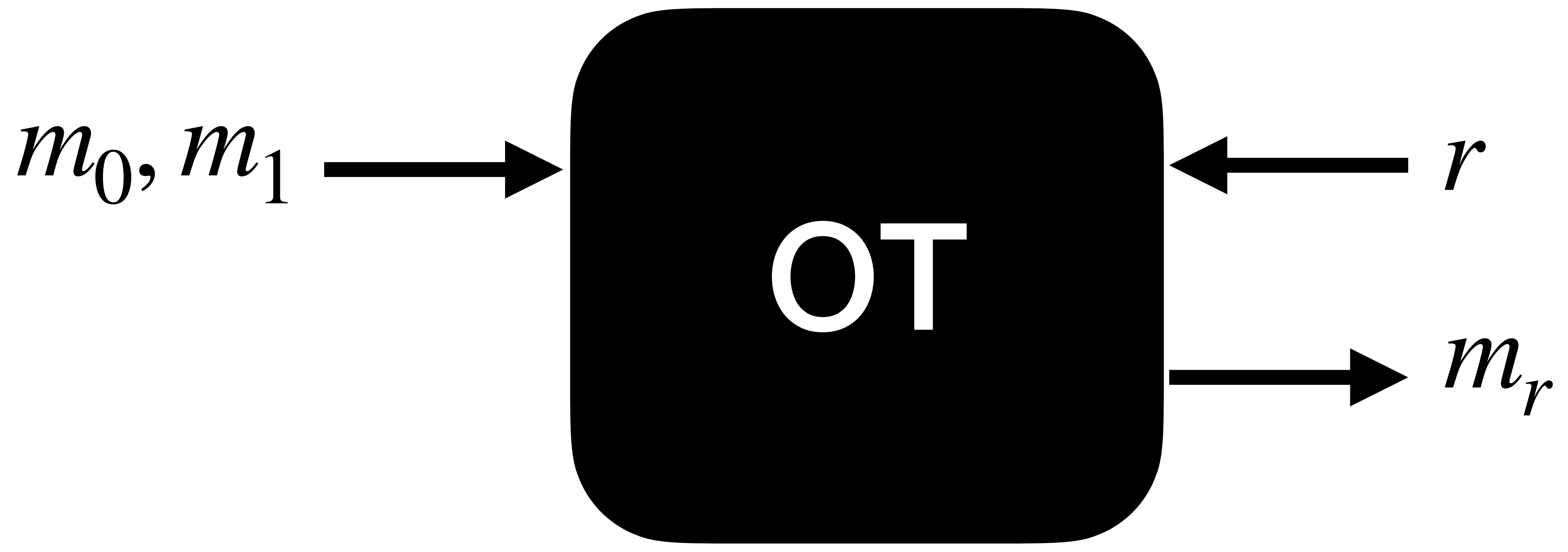
orders of magnitude

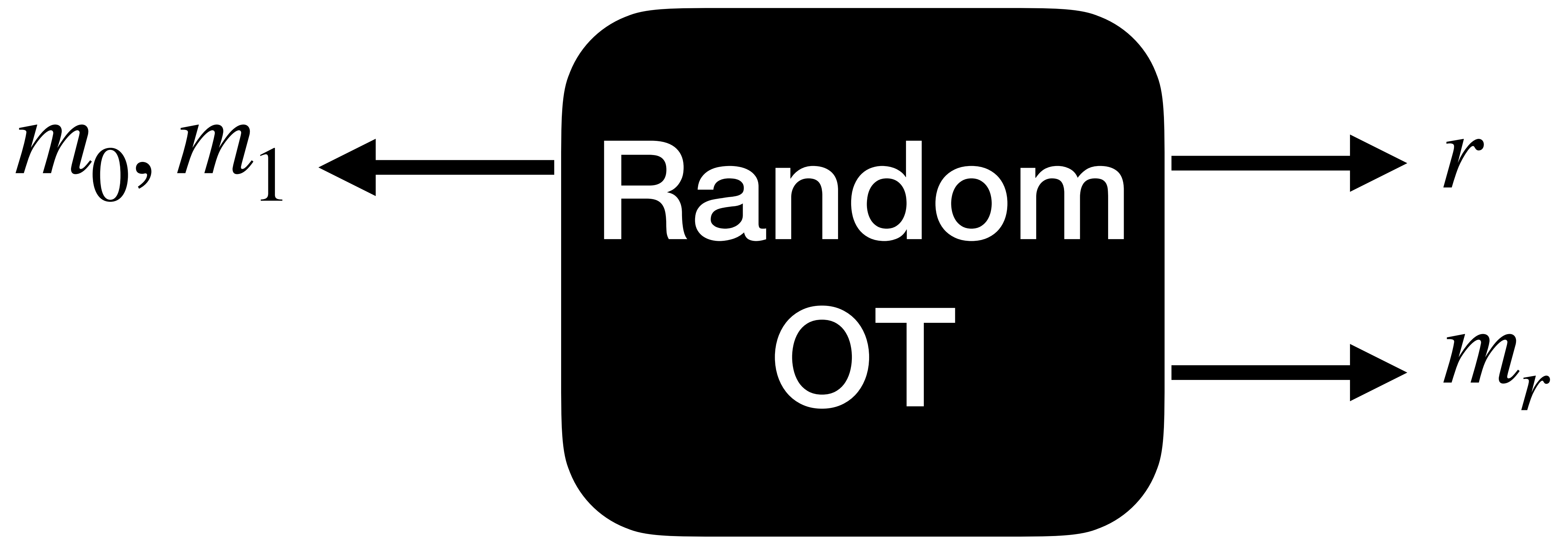
Public Key $\gg$ Symmetric Key $\gg$ Linear Operations



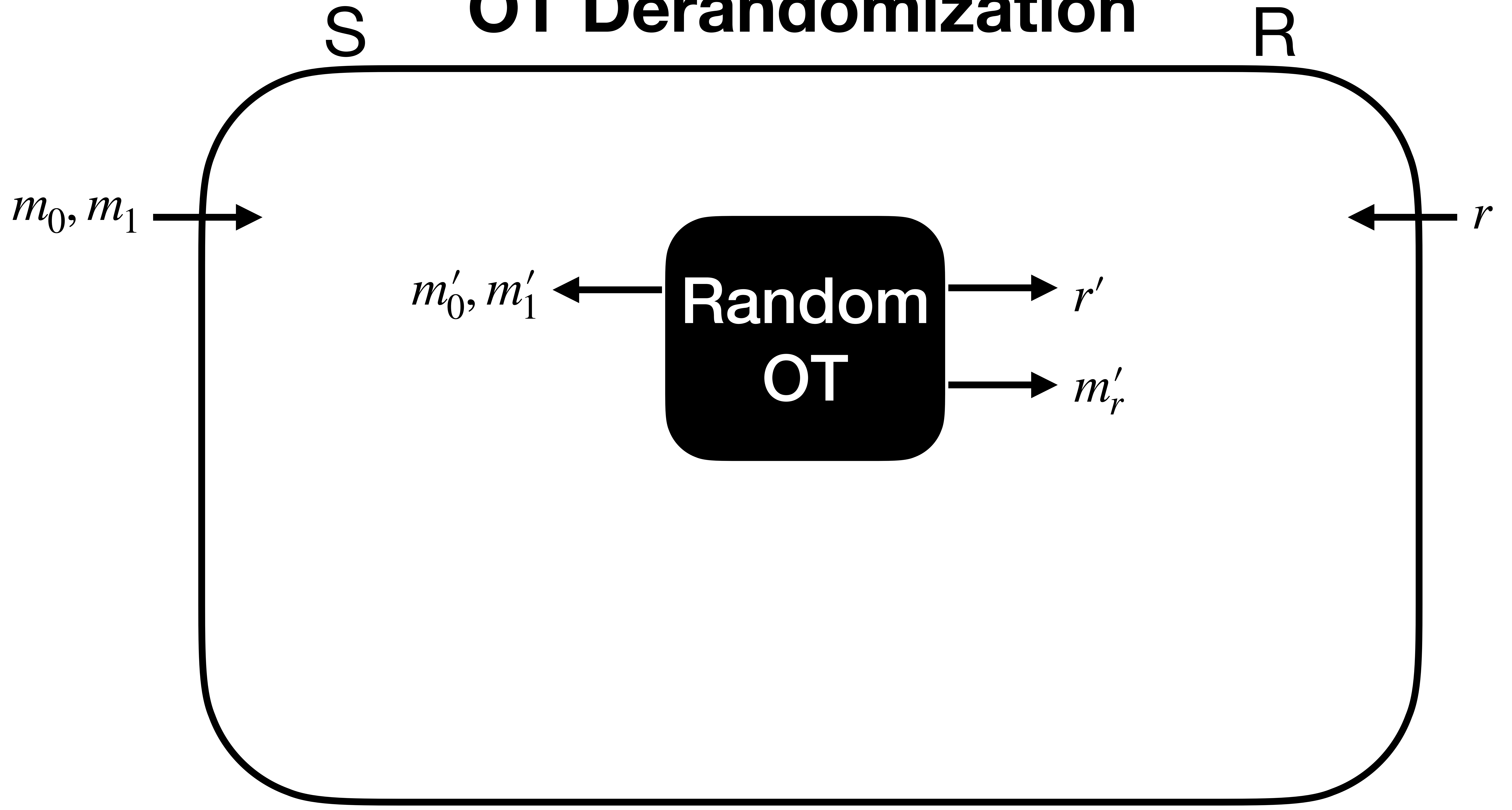
OT requires public key cryptography…

But **how much** public key crypto you need to use?

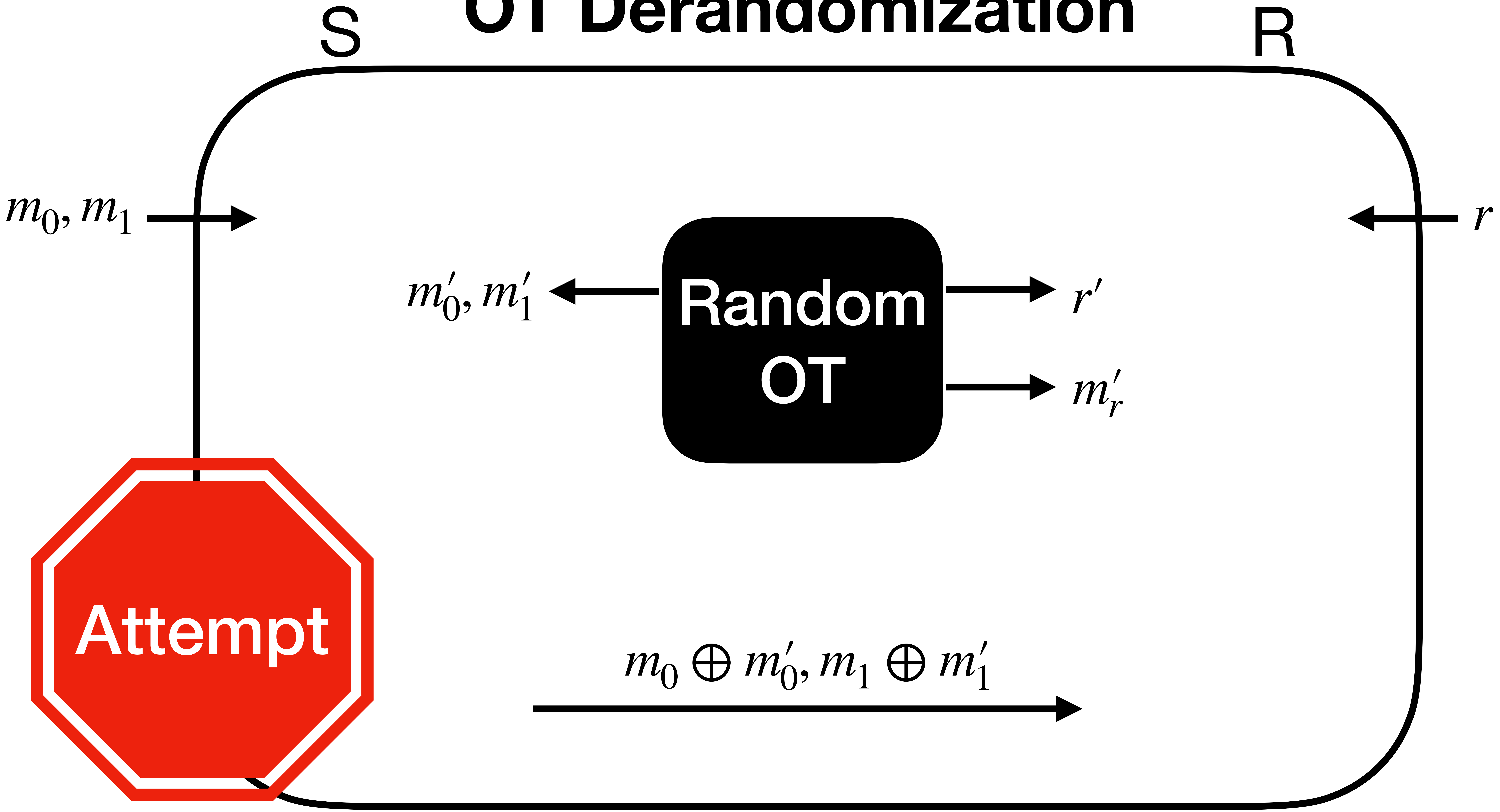

m_0, m_1
OT
r
m_r


m_0, m_1
Random OT
r
m_r

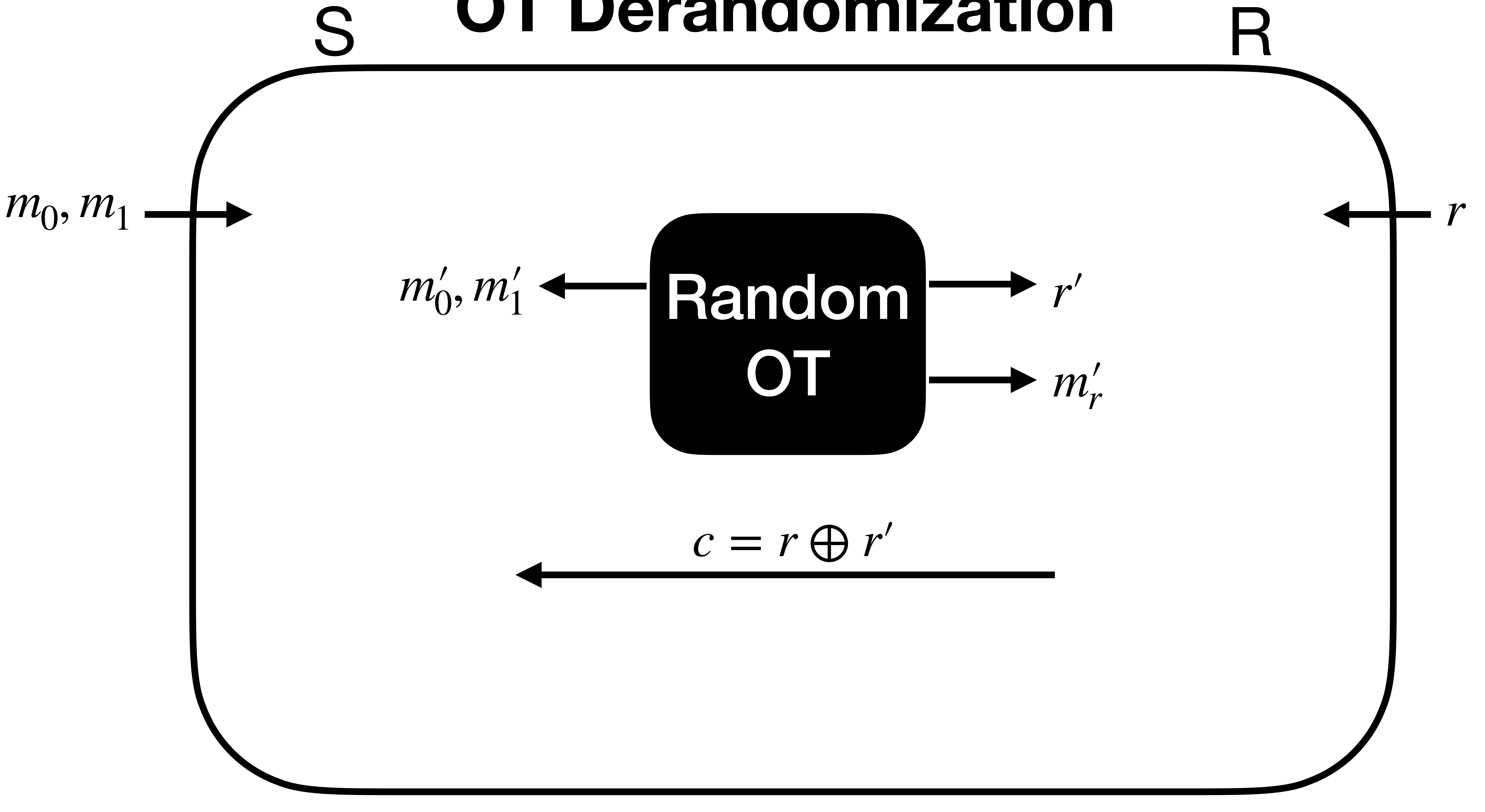
OT Derandomization



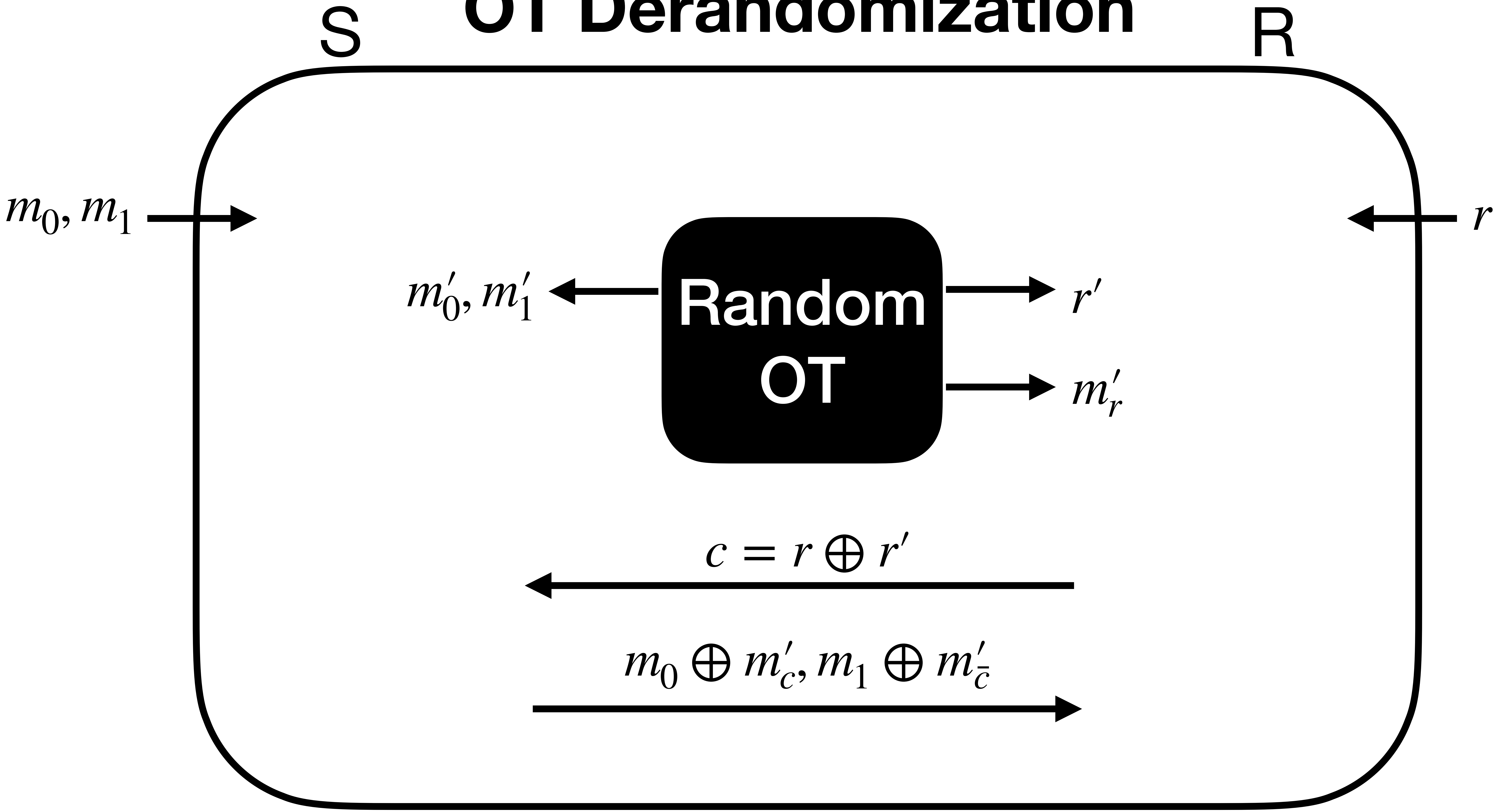
OT Derandomization



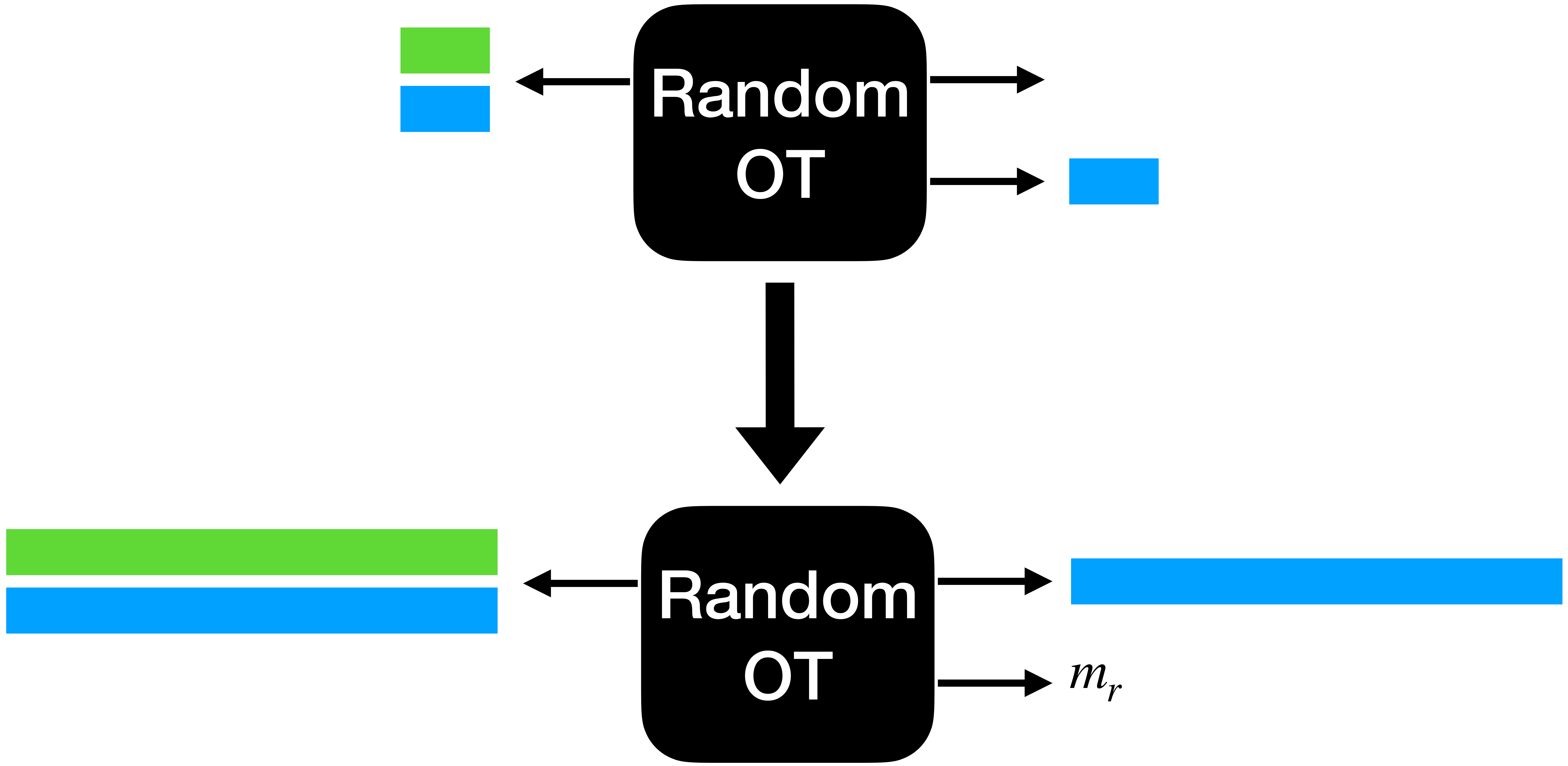
OT Derandomization

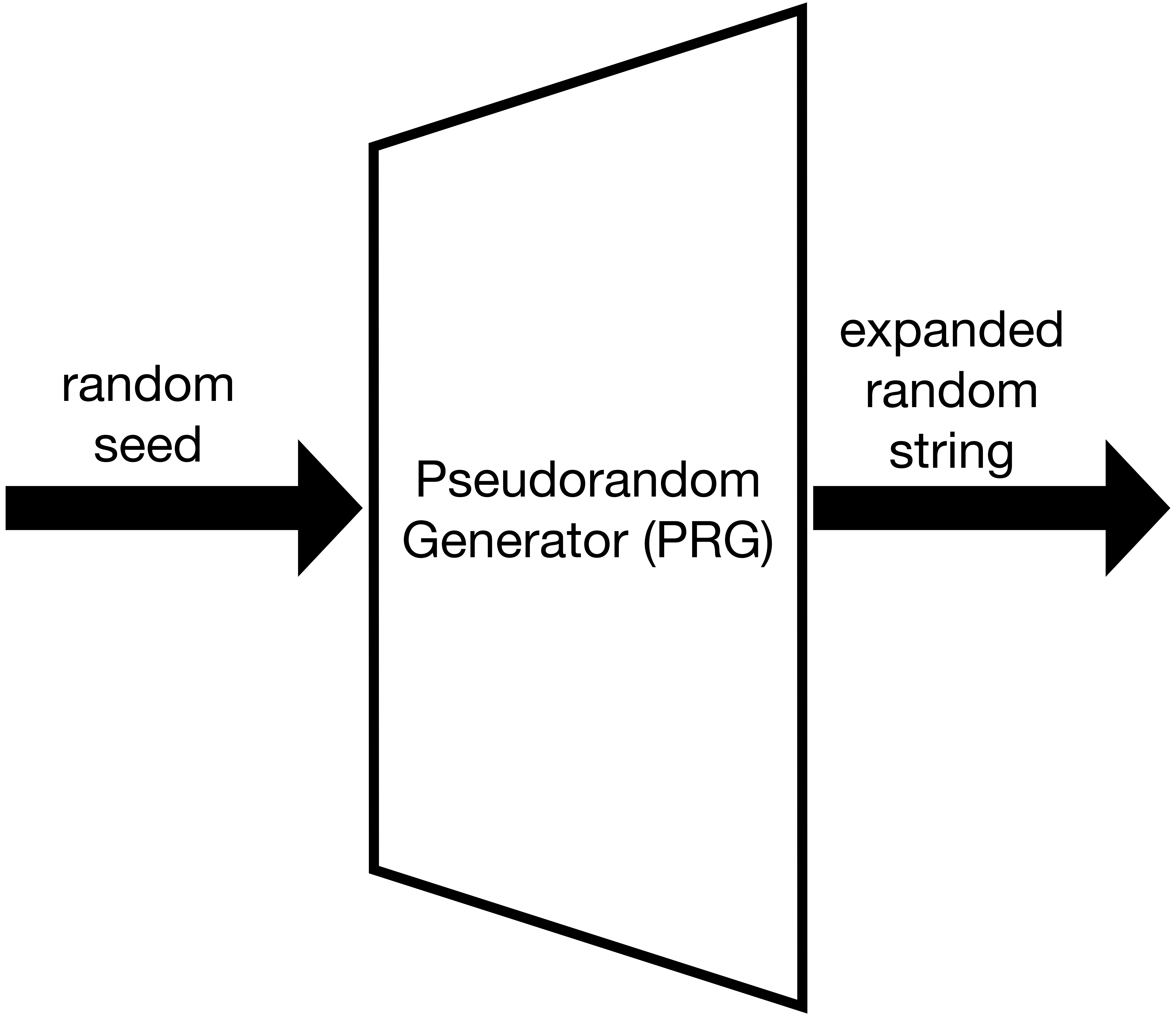


OT Derandomization



OT Length Extension





Pseudorandom generator:

A deterministic algorithm
that expands random strings

If the PRG input is random
and hidden, then the PRG
output is indistinguishable
from random

The Joy of Cryptography

by Mike Rosulek • joyofcryptography.com •

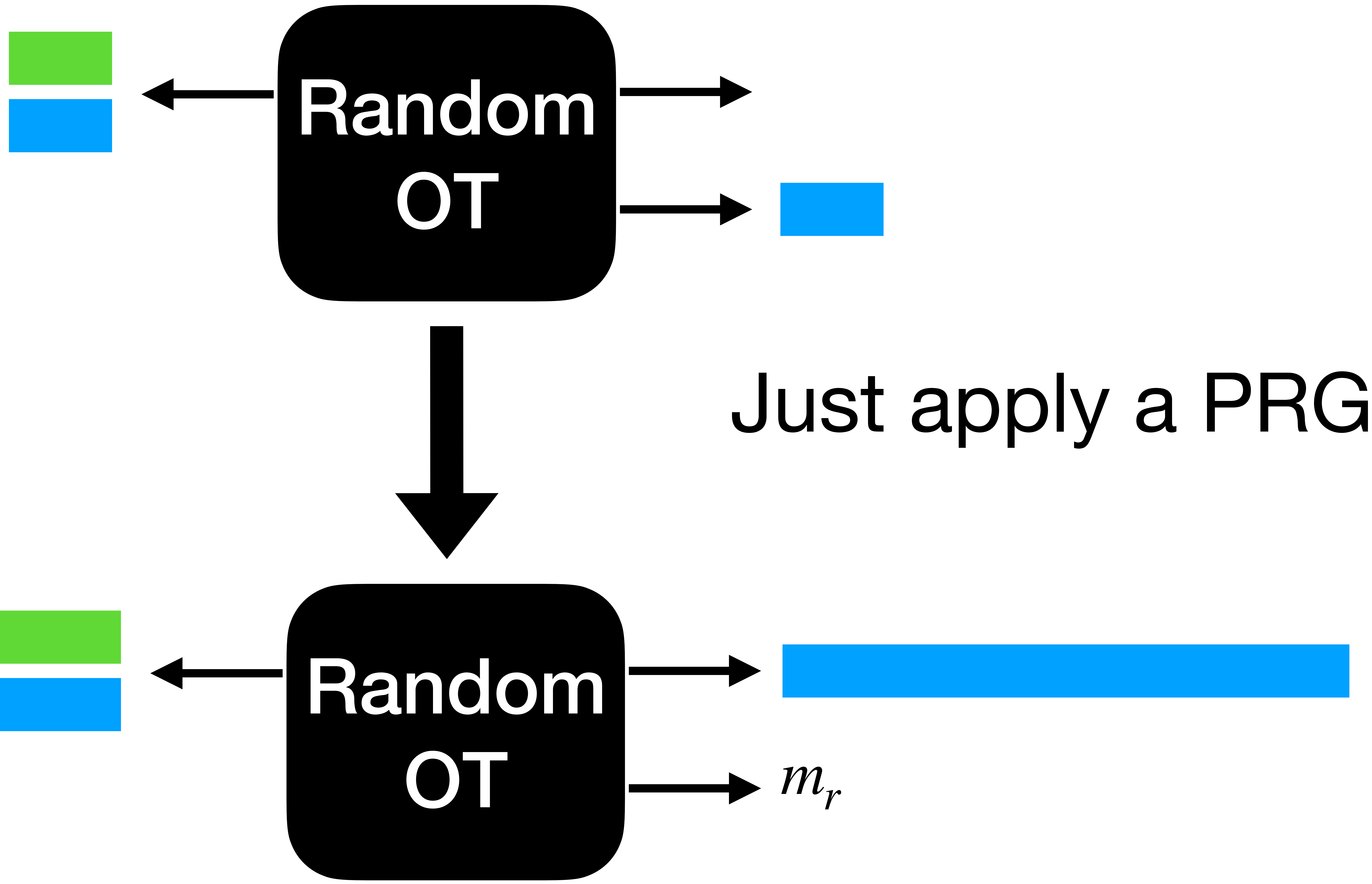
Definition 5.1 (PRG security) *Let $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ be a deterministic function with $\ell > 0$. We say that G is a* ***secure pseudorandom generator (PRG)*** *if $\mathcal{L}^G_{\text{prg-real}} \approx \mathcal{L}^G_{\text{prg-rand}}$, where:*

```
$\mathcal{L}^G_{\text{prg-real}}$
QUERY():
    $s \leftarrow \{0,1\}^\lambda$
    return $G(s)$
```

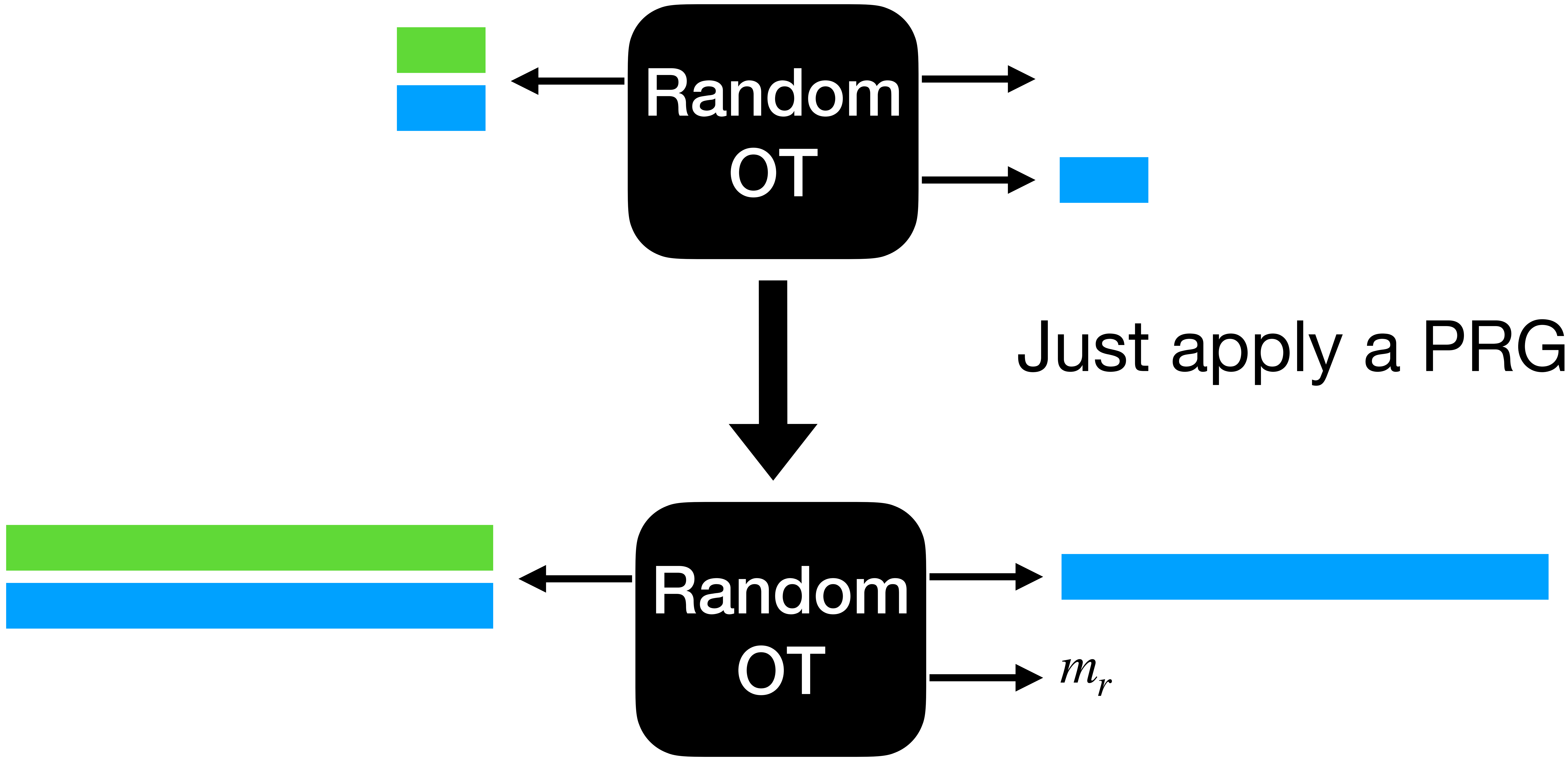
```
$\mathcal{L}^G_{\text{prg-rand}}$
QUERY():
    $r \leftarrow \{0,1\}^{\lambda+\ell}$
    return $r$
```

The value ℓ is called the ***stretch*** *of the PRG. The input to the PRG is typically called a* ***seed****.*

OT Length Extension



Just apply a PRG



OT Extension

In MPC (e.g., GMW), we need lots of short OTs

Can we turn a few OTs into a lot of OTs?

OT Extension

In MPC (e.g., GMW), we need lots of short OTs

Can we turn a few OTs into a lot of OTs?

λ *base* OTs

OT Extension

In MPC (e.g., GMW), we need lots of short OTs

Can we turn a few OTs into a lot of OTs?

λ *base* OTs

Public key

OT Extension

In MPC (e.g., GMW), we need lots of short OTs

Can we turn a few OTs into a lot of OTs?

λ *base* OTs $\longrightarrow$ n *extended* OTs

Public key

OT Extension

In MPC (e.g., GMW), we need lots of short OTs

Can we turn a few OTs into a lot of OTs?

λ *base* OTs $\longrightarrow$ n *extended* OTs

Public key

Symmetric key

Correlated Pseudorandomness and the Complexity of Private Computations

Donald Beaver [*]

Abstract.

The race to find the weakest possible assumptions on which to base cryptographic primitives such as oblivious transfer was abruptly halted by Impagliazzo's and Rudich's surprising result: basing oblivious transfer or other related problems on a black-box one-way permutation (as opposed to a one-way trapdoor permutation) is tantamount to showing P$\neq$NP. In contrast, we show how to generate OT – in the sense of random number generation – using any one-way function in a black-box manner. That is, an initial "seed" of k OT's suffices to generate $O(k^c)$ OT's.

In turn, we show that such generation is impossible in an information-theoretic setting, thus placing OT on an equal footing with random number generation, and resolving an artificial asymmetry in the analysis of randomness and partially-correlated randomness.

We also initiate a complexity theory of privately-computable probabilistic functions[1] and show that there is a provably rich hierarchy among them. Previous work has considered deterministic functions of possibly-random inputs, and focused on whether reductions exist, the class of primitives that are complete, and the amount of information leaked *vs.* message complexity. We show that any complete boolean function gives rise to a nontrivial complexity hierarchy of privately-computable functions, measured according to invocations of a complete primitive – and that this hierarchy collapses when restricted to "computational" security.

[*]Transarc Corp., Gulf Tower, 707 Grant St., Pittsburgh, PA 15219. Email: beaver@transarc.com.

[1]Functions mapping inputs to joint distributions.

1 Introduction

Oblivious Transfer, a broadly used primitive introduced by Rabin [Rab81], is a protocol for sending a bit that arrives with precisely 50-50 probability – without the sender knowing the result. This asymmetry in knowledge makes OT a natural basis for achieving security in a wide variety of interactive protocols, ranging from bit commitment to zero-knowledge proofs to multiparty computations to coin tossing, and most of cryptography (*cf.* [GMW87, Kil88]).

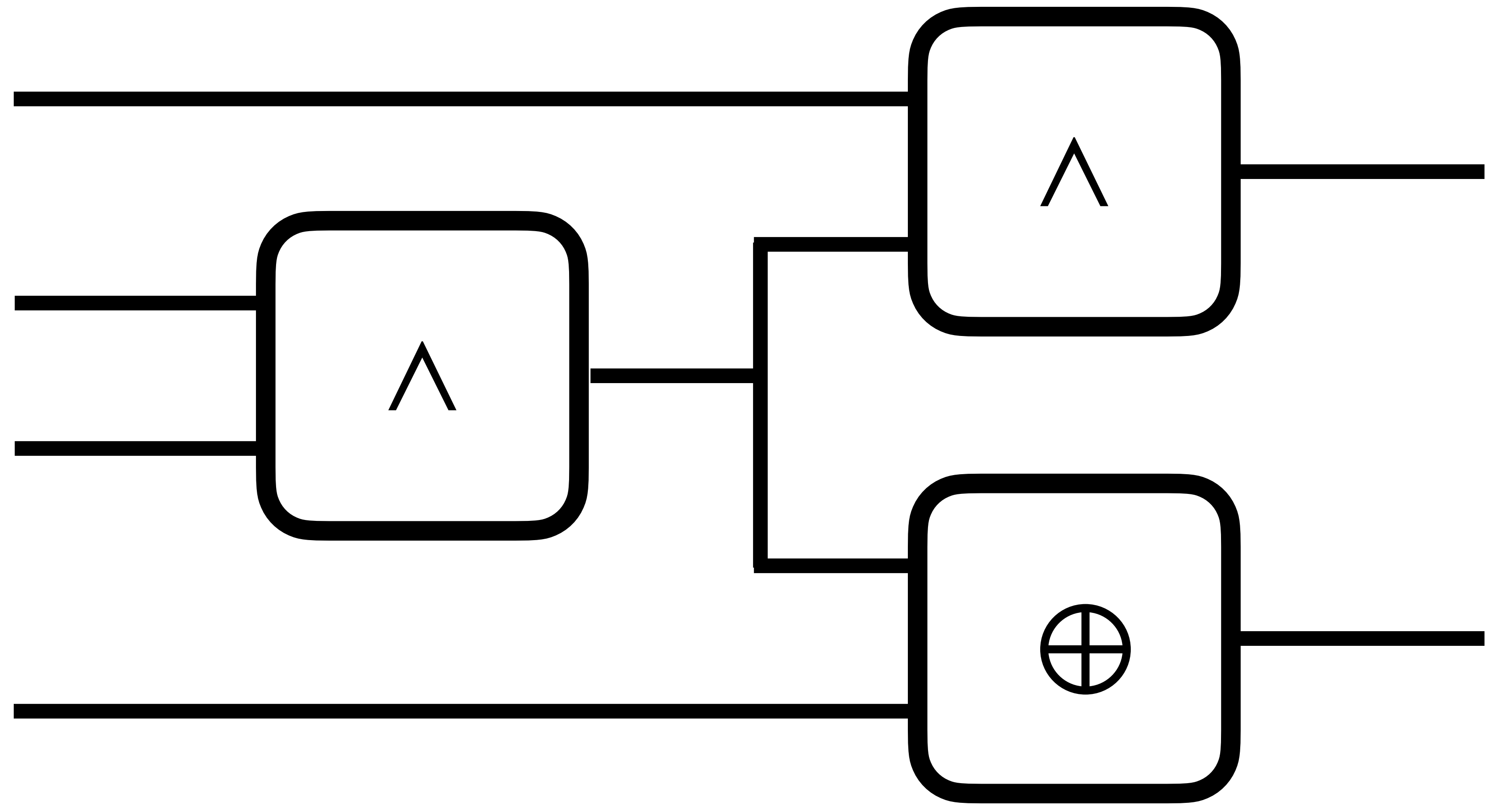
Despite the widespread use of OT as a primitive, implementations of OT rely on relatively strong assumptions, such as the existence of trapdoor one-way permutations and the difficulty of factoring or taking discrete logarithms [Rab81, EGL82, BM89, Boe91].

In 1989, Impagliazzo and Rudich showed a remarkable but negative result: basing OT on weaker assumptions would be a difficult task [IR89]. In particular, if there exists an OT protocol that uses a one-way function as a black-box, then P $\neq$ NP. This result bears strong contrast to pseudorandom number generation, which similarly started with number-theoretic assumptions [BM84] yet was indeed reduced to any one-way function [ILL89].

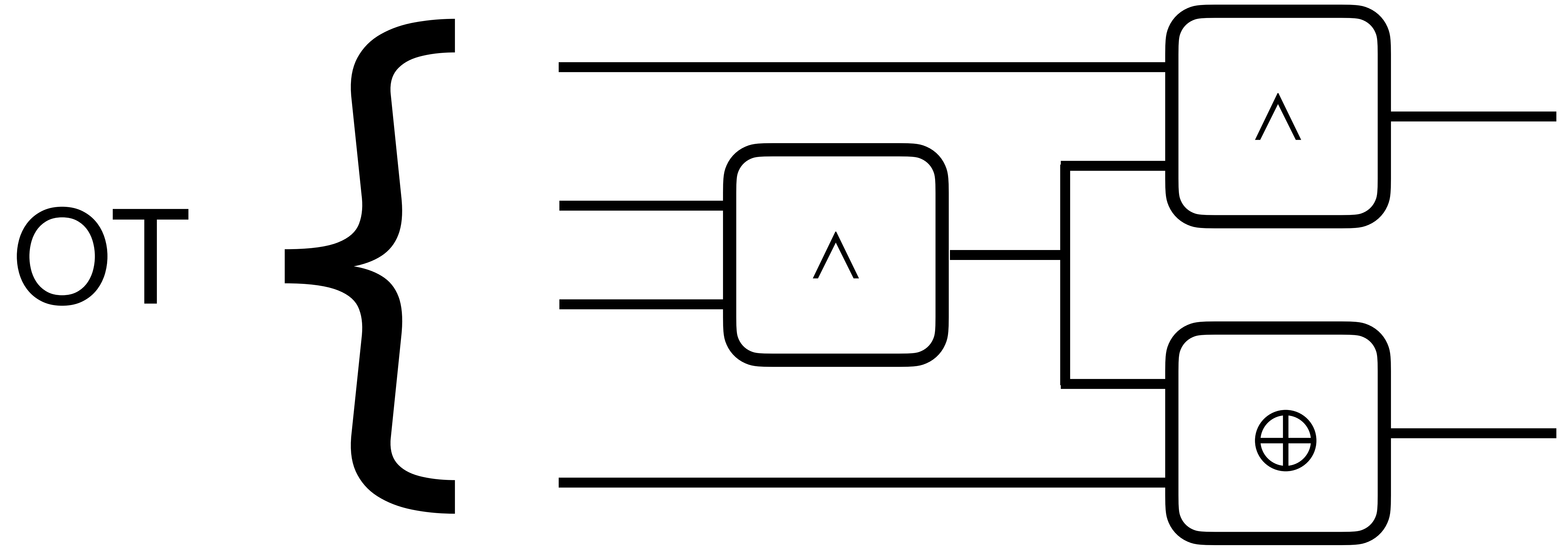
Expenses and Strong Complexity Assumptions. Imagine that quantum OT devices are finally in mass production, but each bit costs a penny to send. Cryptographers rejoice that complexity assumptions are no longer needed for security, but the price is heavy. Or imagine that the security of known trapdoor one-way permutations has been cast in doubt, or that computing them is as expensive as quantum OT. Meanwhile, tantalizingly cheap one-way functions beckon! But images of Impagliazzo and Rudich stand in their way.

Our work shows how to move past [IR89] by placing OT on an equal footing with pseudorandom number generation. In particular, a short "seed" of initial OT's can be expanded into a polynomially-long sequence of OT's, based only on the existence of a one-way function (used as a black box). In light of the intricacies and high OT cost of general two-party protocols and methodologies, it is somewhat surprising that this can

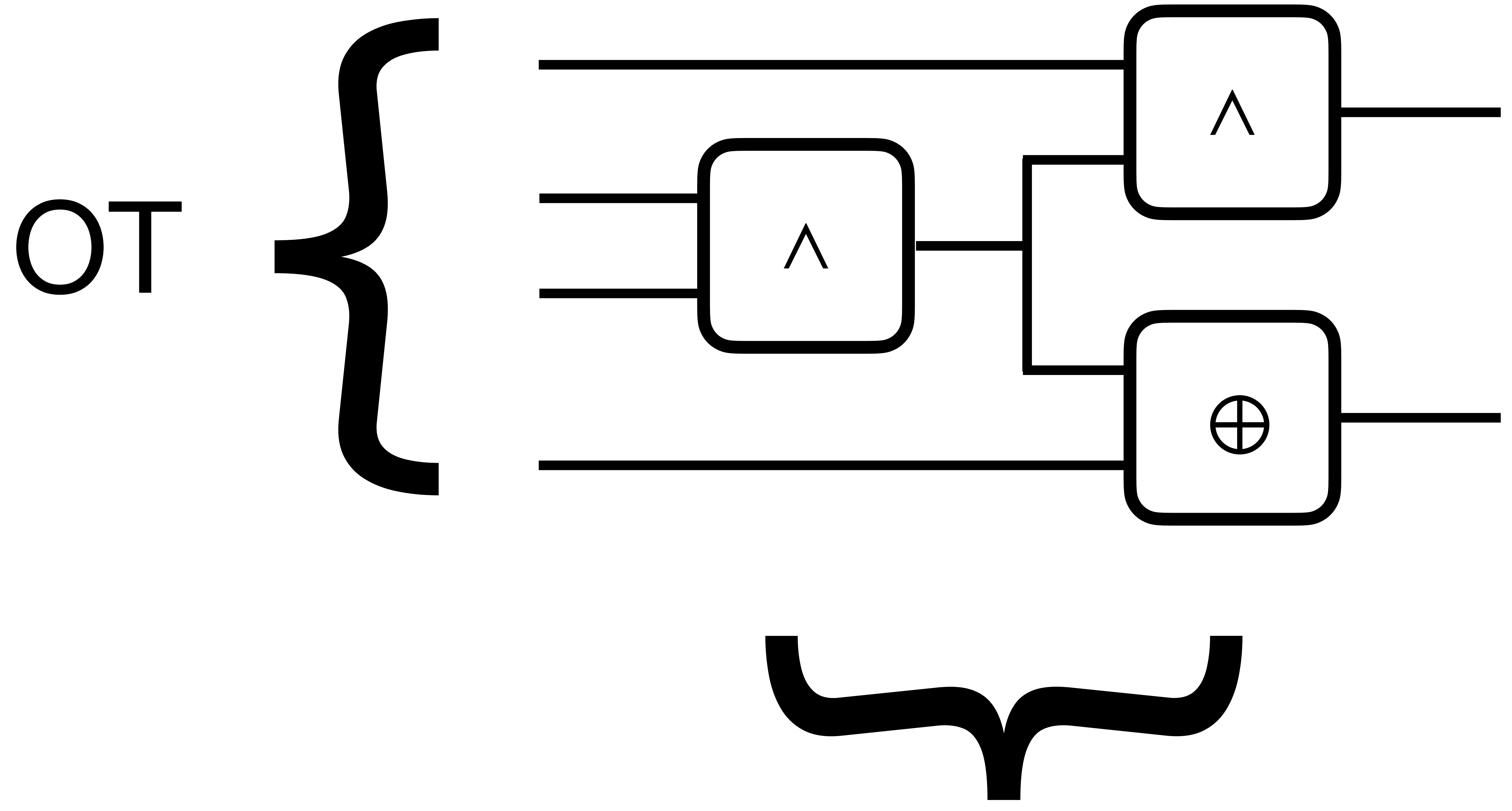
Garbled Circuit



Garbled Circuit



Garbled Circuit



Symmetric key

Garbled Circuit

```
C($k_0, k_1$):
    $k \leftarrow k_0 \oplus k_1$
    for $i$ in [$n$]:
        $r \leftarrow_\$$ G($k$)     // 1 bit
        $m_0 \leftarrow_\$$ G($k$)   // $\lambda$ bits
        $m_1 \leftarrow_\$$ G($k$)   // $\lambda$ bits
        output $(m_0, m_1), (r, m_r)$
```

Garbled Circuit

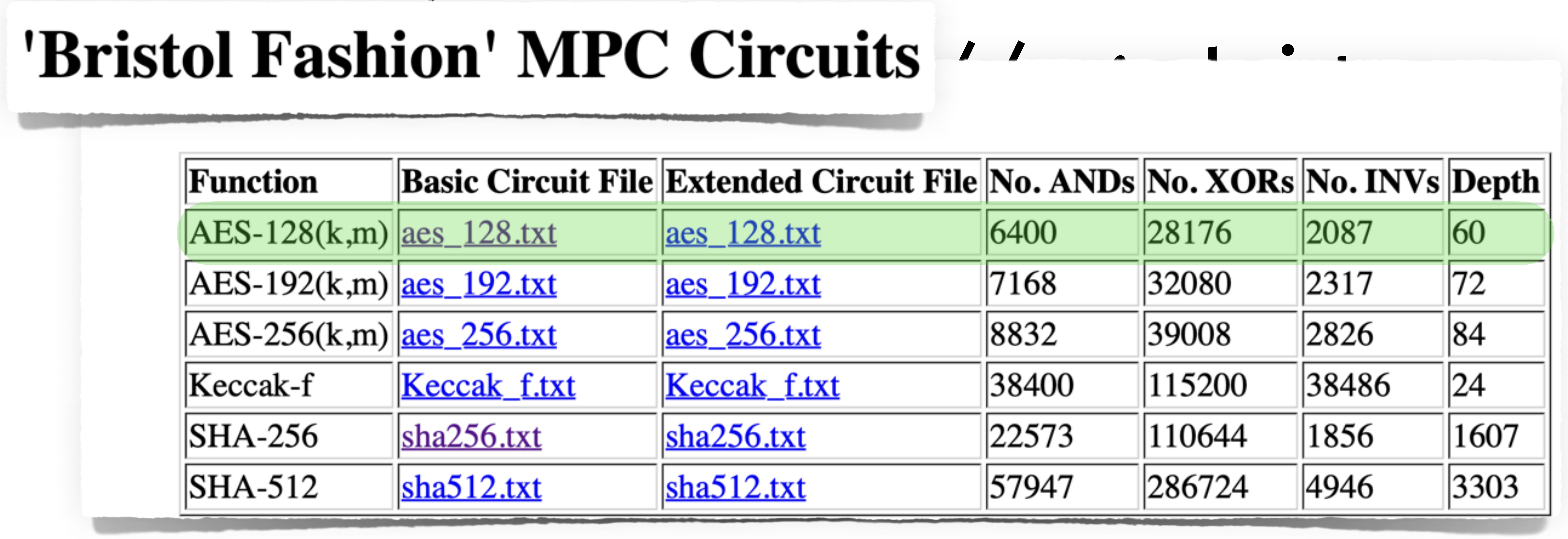
```
C($k_0, k_1$):
    $k \leftarrow k_0 \oplus k_1$
    for $i$ in [n]:
        $r \leftarrow_\$$ G($k$)     // 1 bit
        $m_0 \leftarrow_\$$ G($k$)   // $\lambda$ bits
        $m_1 \leftarrow_\$$ G($k$)   // $\lambda$ bits
        output $(m_0, m_1), (r, m_r)$
```

Often in practice, implement PRG G with AES

Garbled Circuit

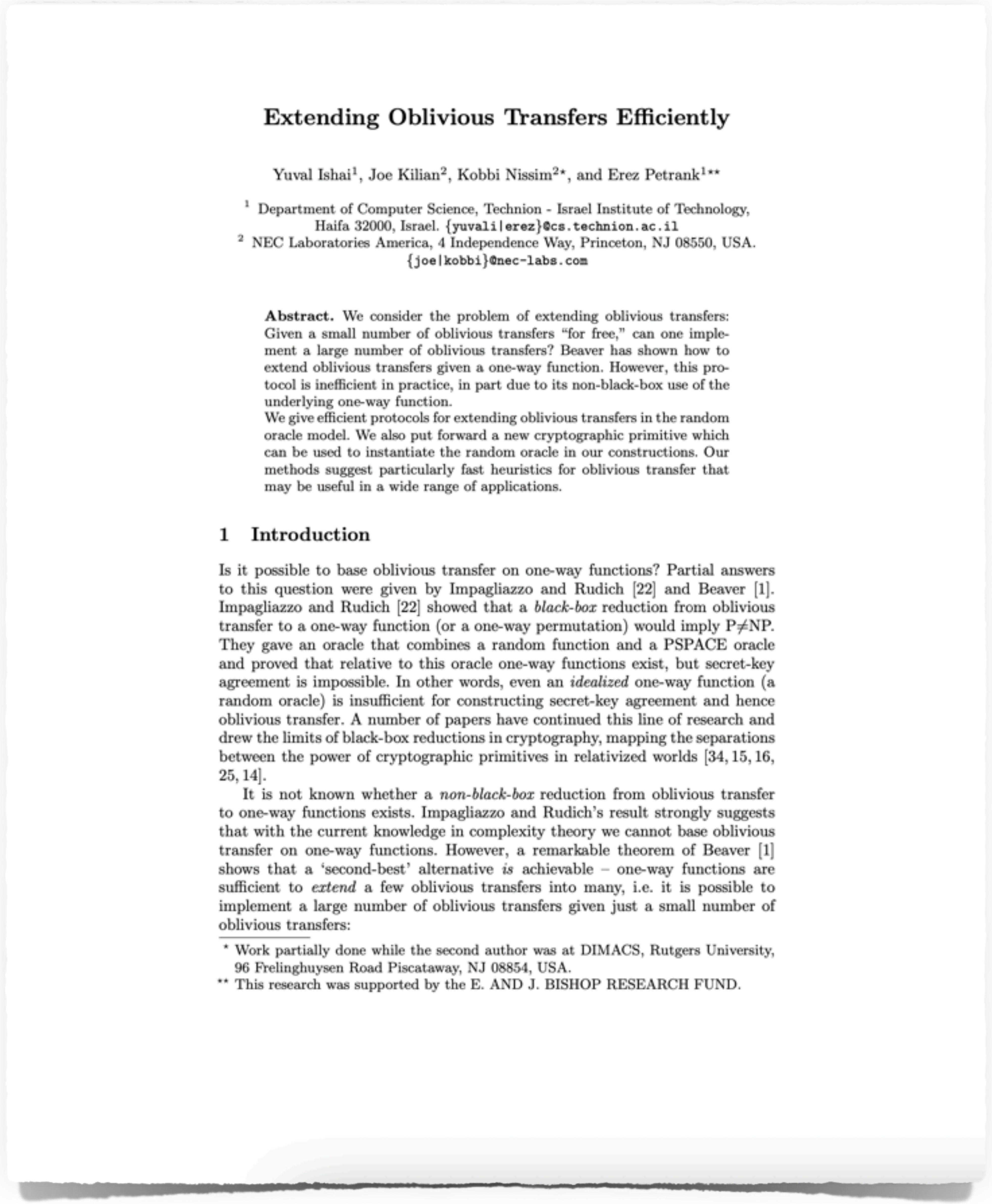
```
C($k_0, k_1$):
    $k \leftarrow k_0 \oplus k_1$
    for $i$ in [n]:
        $r \leftarrow_{\$}$ G($k$)      // 1 bit
```

Often in practice, implement PRG G with AES



'Bristol Fashion' MPC Circuits

Function	Basic Circuit File	Extended Circuit File	No. ANDs	No. XORs	No. INVs	Depth
AES-128(k,m)	aes_128.txt	aes_128.txt	6400	28176	2087	60
AES-192(k,m)	aes_192.txt	aes_192.txt	7168	32080	2317	72
AES-256(k,m)	aes_256.txt	aes_256.txt	8832	39008	2826	84
Keccak-f	Keccak_f.txt	Keccak_f.txt	38400	115200	38486	24
SHA-256	sha256.txt	sha256.txt	22573	110644	1856	1607
SHA-512	sha512.txt	sha512.txt	57947	286724	4946	3303



Extending Oblivious Transfers Efficiently

Yuval Ishai[1], Joe Kilian[2], Kobbi Nissim[2*], and Erez Petrank[1**]

[1] Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel. {yuvali|erez}@cs.technion.ac.il
[2] NEC Laboratories America, 4 Independence Way, Princeton, NJ 08550, USA. {joe|kobbi}@nec-labs.com

Abstract. We consider the problem of extending oblivious transfers: Given a small number of oblivious transfers "for free," can one implement a large number of oblivious transfers? Beaver has shown how to extend oblivious transfers given a one-way function. However, this protocol is inefficient in practice, in part due to its non-black-box use of the underlying one-way function.
We give efficient protocols for extending oblivious transfers in the random oracle model. We also put forward a new cryptographic primitive which can be used to instantiate the random oracle in our constructions. Our methods suggest particularly fast heuristics for oblivious transfer that may be useful in a wide range of applications.

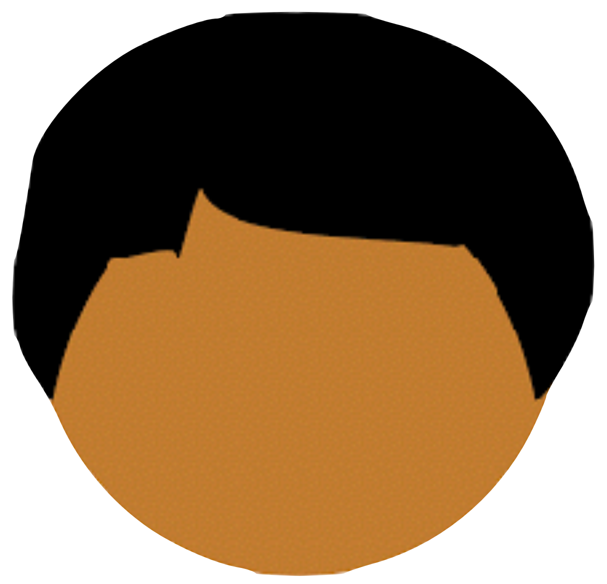
1 Introduction

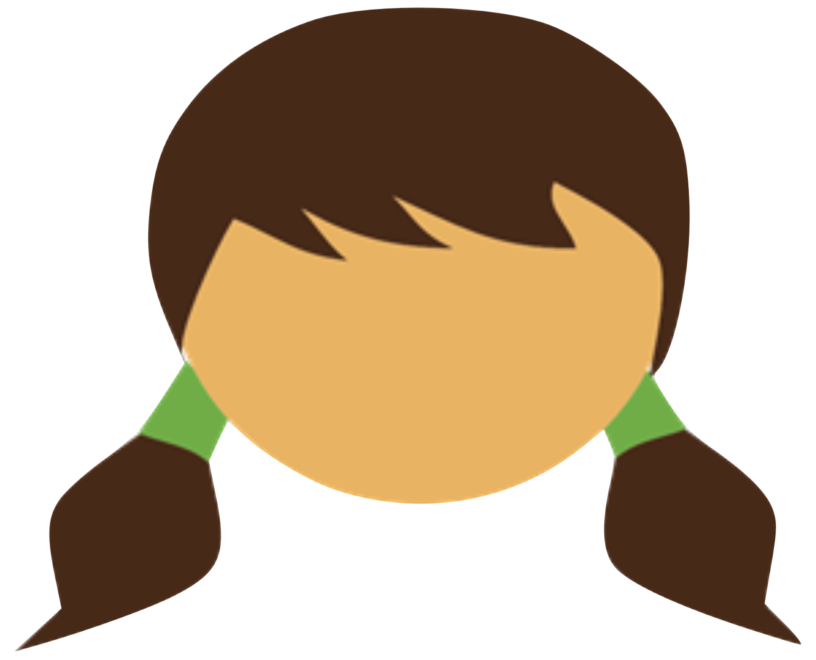
Is it possible to base oblivious transfer on one-way functions? Partial answers to this question were given by Impagliazzo and Rudich [22] and Beaver [1]. Impagliazzo and Rudich [22] showed that a *black-box* reduction from oblivious transfer to a one-way function (or a one-way permutation) would imply P≠NP. They gave an oracle that combines a random function and a PSPACE oracle and proved that relative to this oracle one-way functions exist, but secret-key agreement is impossible. In other words, even an *idealized* one-way function (a random oracle) is insufficient for constructing secret-key agreement and hence oblivious transfer. A number of papers have continued this line of research and drew the limits of black-box reductions in cryptography, mapping the separations between the power of cryptographic primitives in relativized worlds [34, 15, 16, 25, 14].

It is not known whether a *non-black-box* reduction from oblivious transfer to one-way functions exists. Impagliazzo and Rudich's result strongly suggests that with the current knowledge in complexity theory we cannot base oblivious transfer on one-way functions. However, a remarkable theorem of Beaver [1] shows that a 'second-best' alternative *is* achievable – one-way functions are sufficient to *extend* a few oblivious transfers into many, i.e. it is possible to implement a large number of oblivious transfers given just a small number of oblivious transfers:

* Work partially done while the second author was at DIMACS, Rutgers University, 96 Frelinghuysen Road Piscataway, NJ 08854, USA.
** This research was supported by the E. AND J. BISHOP RESEARCH FUND.

One of the most important results in secure computation, because it demonstrates how to **efficiently** construct **large numbers** of OTs


S

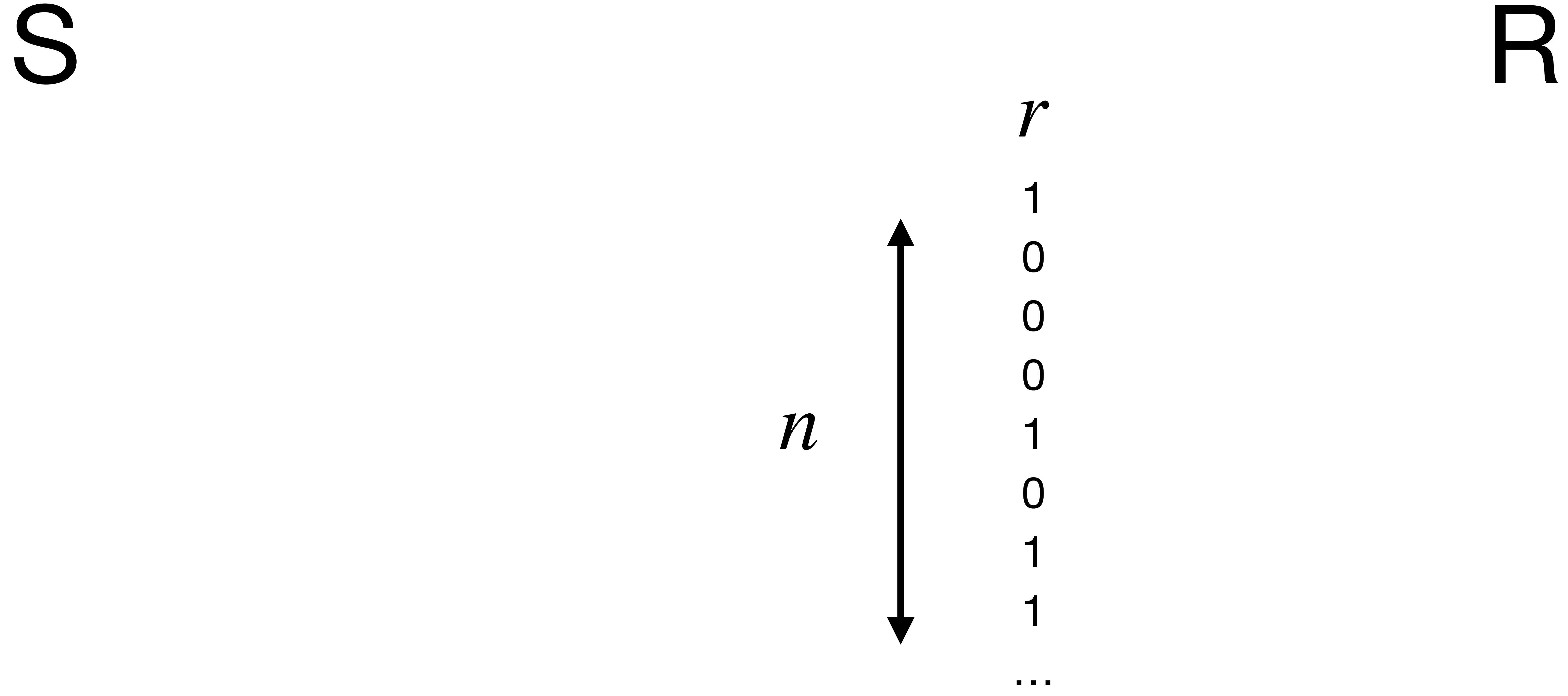

R

First, use public key cryptography to implement “backwards” OT

λ times, R sends to S one of two long random strings

Locally transform these into large number of random OTs from S to R

- Matrix algebra
- Simple symmetric key operations


S
R
r
n
1
0
0
0
1
0
1
1
...

S

R

R samples a long random string r and makes a matrix that repeats λ times

$$\begin{array}{cc|cccccccc}
 & r & & & & & & & & \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & \multicolumn{8}{c}{\longleftrightarrow} \\
 & & \multicolumn{8}{c}{\lambda}
\end{array}$$

S

R

R samples a long random string r and makes a matrix that repeats λ times

$$\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots &
\end{array}$$

R makes two secret shares of her matrix, t and s

$$\begin{array}{cccccccc|c}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots &
\end{array}$$

S

$$\Delta \quad \underline{1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1}$$

S samples a length λ random string Δ

The correlation

Δ is S's secret

R

$$\begin{array}{c|cccccccc c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$\begin{array}{cccccccc c}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & & & & & & & \\
 & 1 & & & & & & & \\
 & 0 & & & & & & & \\
 & 1 & & & & & & & \\
 & 0 & & & & & & & \\
 & 1 & & & & & & & \\
 & 1 & & & & & & & \\
 & 1 & & & & & & & \\
 & \cdots & & & & & & &
\end{array}$$

R

$$\begin{array}{c|cccccccc c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$\begin{array}{cccccccc c}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & & & & & & \\
 & 1 & 0 & & & & & & \\
 & 0 & 1 & & & & & & \\
 & 1 & 1 & & & & & & \\
 & 0 & \mathbf{1} & & & & & & \\
 & 1 & 0 & & & & & & \\
 & 1 & \mathbf{0} & & & & & & \\
 & 1 & \mathbf{0} & & & & & & \\
 & \cdots & \cdots & & & & & &
\end{array}$$

R

$$\begin{array}{c|cccccccc}
r & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array} \quad t$$

$$\begin{array}{cccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array} \quad s$$

S

Δ	0	1	0	0	0	1	1	0

1	**0**	0
1	0	1
0	1	1
1	1	0
0	**1**	1
1	0	1
1	**0**	1
1	**0**	0
…	…	…

R

r									
1	1	1	0	1	1	0	0	1	
0	1	0	1	1	0	1	0	0	
0	0	1	1	0	0	0	1	1	
0	1	1	0	1	1	0	1	1	
1	0	0	1	0	1	0	0	1	*t*
0	1	0	1	0	1	0	0	0	
1	1	1	1	1	1	0	1	0	
1	1	1	0	0	0	1	0	1	
…	…	…	…	…	…	…	…	…	

0	**0**	**1**	**0**	**0**	**1**	**1**	**0**	
1	0	1	1	0	1	0	0	
0	1	1	0	0	0	1	1	
1	1	0	1	1	0	1	1	
1	**1**	**0**	**1**	**0**	**1**	**1**	**0**	*s*
1	0	1	0	1	0	0	0	
0	**0**	**0**	**0**	**0**	**1**	**0**	**1**	
0	**0**	**1**	**1**	**1**	**0**	**1**	**0**	
…	…	…	…	…	…	…	…	

S

$$
\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}
$$

R

$$
\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\
 & & & & & & & & & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
 & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}
$$

S

$$\begin{array}{ccccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

R

$$\begin{array}{c|ccccccccc}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$\begin{array}{ccccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

S

Δ	0	1	0	0	0	1	1	0

	1	**0**	0	1	1	**1**	**1**	1
	1	0	1	1	0	1	0	0
	0	1	1	0	0	0	1	1
	1	1	0	1	1	0	1	1
q	0	**1**	1	0	1	**1**	**1**	1
	1	0	1	0	1	0	0	0
	1	**0**	1	1	1	**1**	**0**	0
	1	**0**	0	0	0	**0**	**1**	1
	…	…	…	…	…	…	…	…

$$q_i = \begin{cases} t_i & \text{if } r_i = 0 \\ t_i \oplus \Delta & \text{if } r_i = 1 \end{cases}$$

R

r									
1	1	1	0	1	1	0	0	1	
0	1	0	1	1	0	1	0	0	
0	0	1	1	0	0	0	1	1	
0	1	1	0	1	1	0	1	1	
1	0	0	1	0	1	0	0	1	t
0	1	0	1	0	1	0	0	0	
1	1	1	1	1	1	0	1	0	
1	1	1	0	0	0	1	0	1	
…	…	…	…	…	…	…	…	…	

0	**0**	**1**	**0**	**0**	**1**	**1**	**0**	
1	0	1	1	0	1	0	0	
0	1	1	0	0	0	1	1	
1	1	0	1	1	0	1	1	
1	**1**	**0**	**1**	**0**	**1**	**1**	**0**	s
1	0	1	0	1	0	0	0	
0	**0**	**0**	**0**	**0**	**1**	**0**	**1**	
0	**0**	**1**	**1**	**1**	**0**	**1**	**0**	
…	…	…	…	…	…	…	…	

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

$$q_i = t_i \oplus r_i \Delta$$

R

$$\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$\begin{array}{cccccccc|c}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

S

Δ	0	1	0	0	0	1	1	0
	1	**0**	0	1	1	**1**	**1**	1
	1	0	1	1	0	1	0	0
	0	1	1	0	0	0	1	1
	1	1	0	1	1	0	1	1
q	0	**1**	1	0	1	**1**	**1**	1
	1	0	1	0	1	0	0	0
	1	**0**	1	1	1	**1**	**0**	0
	1	**0**	0	0	0	**0**	**1**	1
	…	…	…	…	…	…	…	…

$$q_i = t_i \oplus r_i\Delta$$

R

r									
1	1	1	0	1	1	0	0	1	
0	1	0	1	1	0	1	0	0	
0	0	1	1	0	0	0	1	1	
0	1	1	0	1	1	0	1	1	
1	0	0	1	0	1	0	0	1	t
0	1	0	1	0	1	0	0	0	
1	1	1	1	1	1	0	1	0	
1	1	1	0	0	0	1	0	1	
…	…	…	…	…	…	…	…	…	

$$t_i$$

S

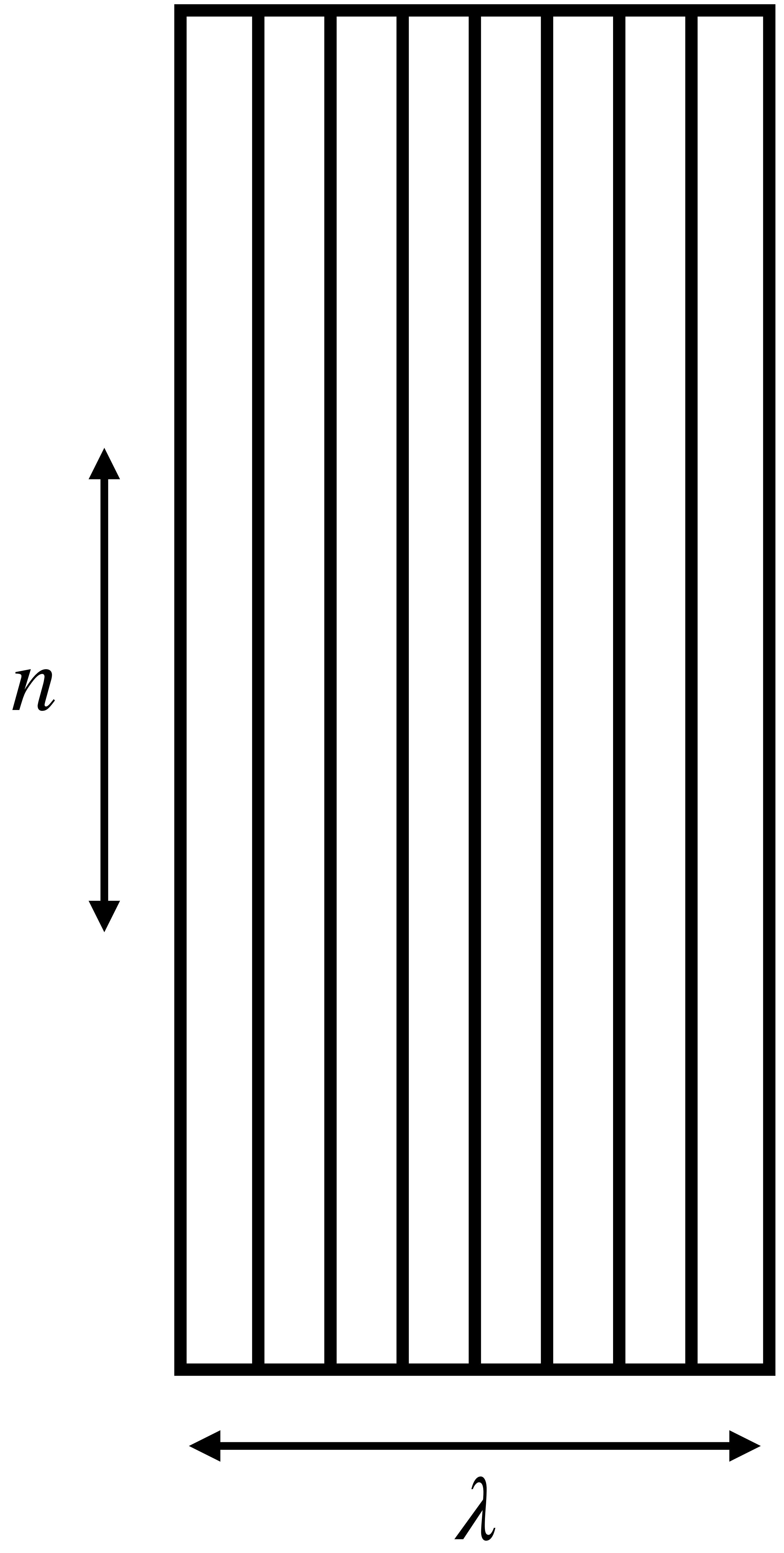
$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

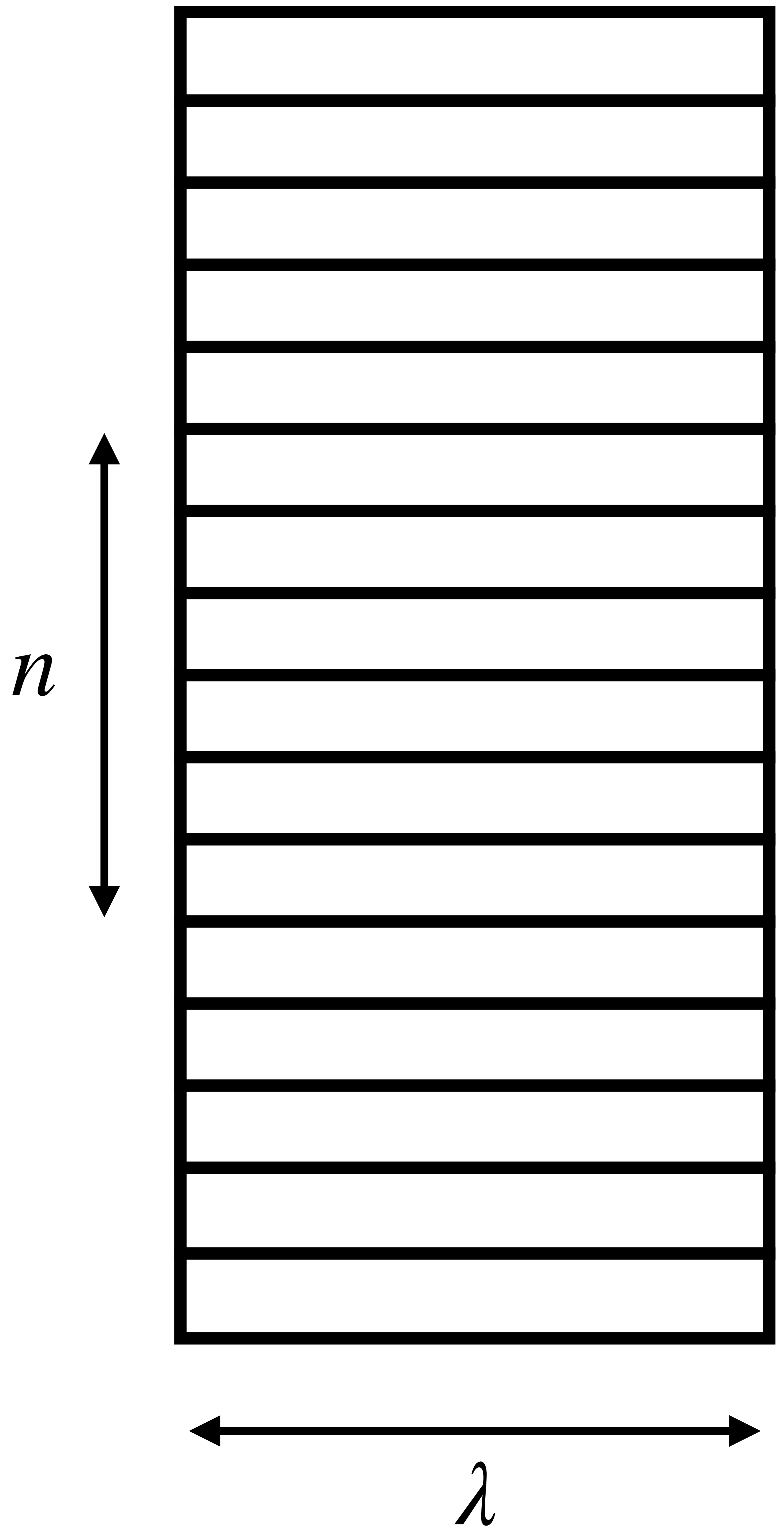
q_i

R

$$\begin{array}{c|cccccccc}
r & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \quad t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

$q_i \oplus r_i \Delta$


n
λ


n
λ

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

R

$$\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$\begin{array}{c|cccccccc|c}
 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
s^i = t^i \oplus r & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

$$q^i = \oplus \begin{cases} t^i & \text{if } \Delta_i = 0 \\ s^i & \text{if } \Delta_i = 1 \end{cases}$$

R

$$\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$\begin{array}{c|cccccccc|c}
 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
s^i = t^i \oplus r & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & s \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \\
 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

$$q^i = \oplus \begin{cases} t^i & \text{if } \Delta_i = 0 \\ s^i & \text{if } \Delta_i = 1 \end{cases}$$

$$q^i = t^i \oplus \begin{cases} 0^n & \text{if } \Delta_i = 0 \\ r & \text{if } \Delta_i = 1 \end{cases}$$

R

$$\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$s^i = t^i \oplus r \quad \begin{array}{cccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array} \quad s$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

$$q^i = \oplus \begin{cases} t^i & \text{if } \Delta_i = 0 \\ s^i & \text{if } \Delta_i = 1 \end{cases}$$

$$q^i = t^i \oplus \Delta_i \cdot r$$

R

$$\begin{array}{c|cccccccc}
r & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array} \quad t$$

$$s^i = t^i \oplus r \quad \begin{array}{cccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\
\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array} \quad s$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

R

$$\begin{array}{c|ccccccccc}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

$$q^i = t^i \oplus \Delta_i \cdot r$$

S

$$
\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}
$$

R

$$
\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}
$$

$$q^i = t^i \oplus \Delta_i \cdot r$$

$$q = t \oplus \Delta \cdot r^{\intercal}$$

S

$$
\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}
$$

R

$$
\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}
$$

$$q^i = t^i \oplus \Delta_i \cdot r$$

$$q_i = t_i \oplus r_i \cdot \Delta$$

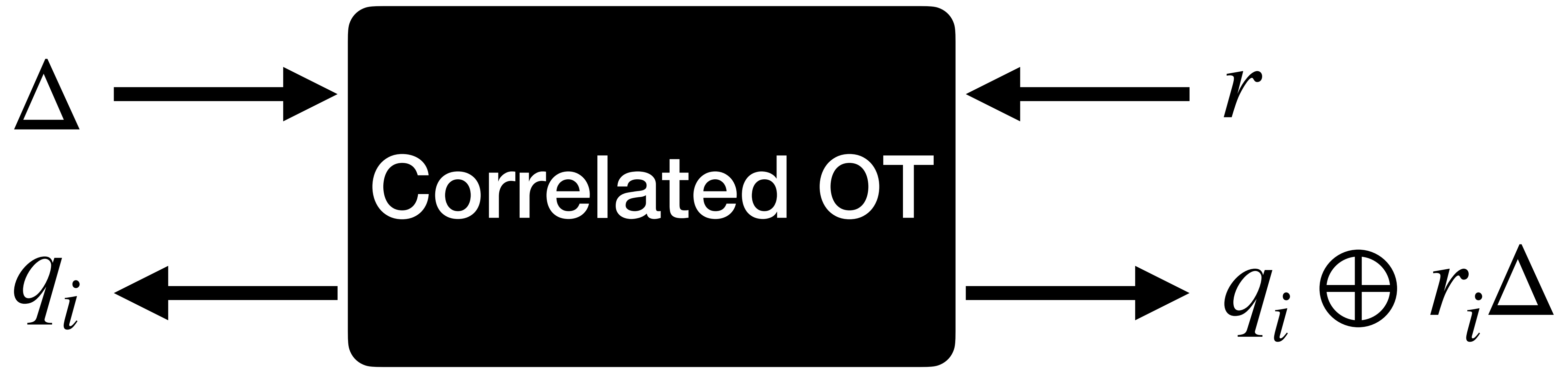
$$q = t \oplus \Delta \cdot r^{\intercal}$$

S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

R

$$\begin{array}{c|cccccccc|c}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

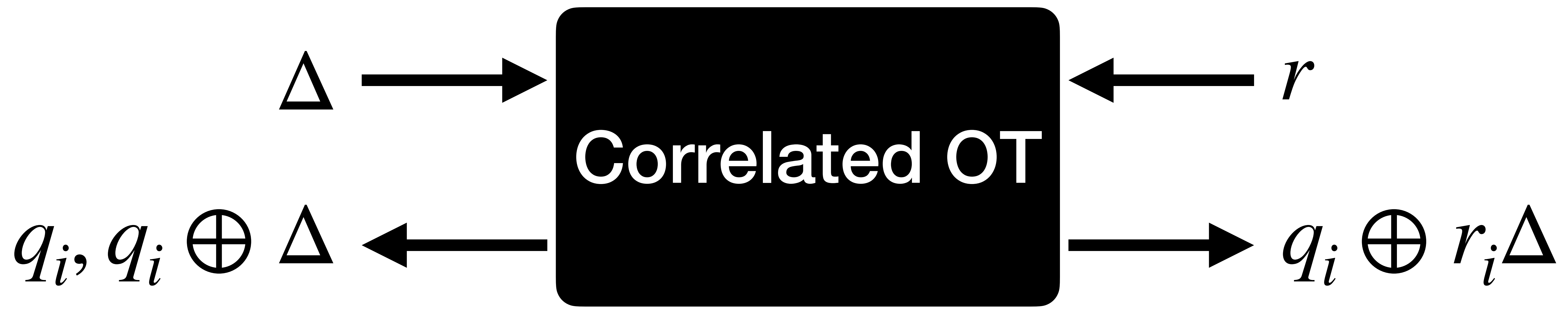


S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

R

$$\begin{array}{c|ccccccccc}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$

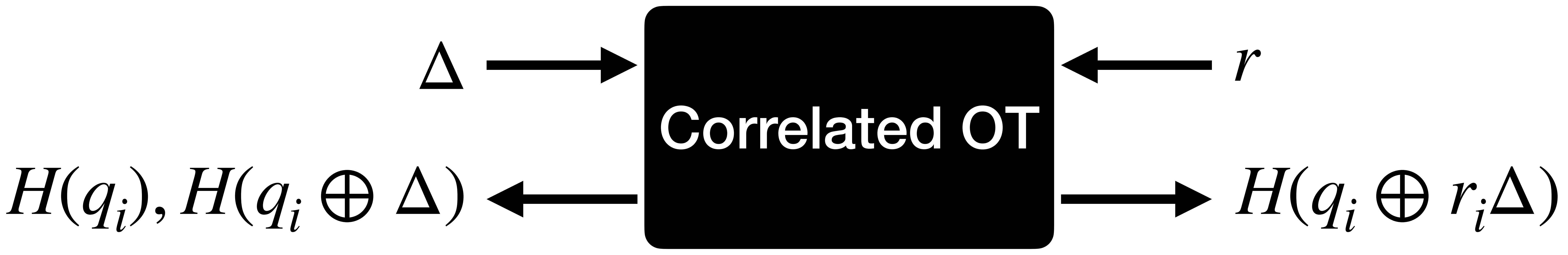


S

$$\begin{array}{c|cccccccc}
\Delta & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & 1 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
q & 0 & \mathbf{1} & 1 & 0 & 1 & \mathbf{1} & \mathbf{1} & 1 \\
 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & 1 & \mathbf{0} & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & 0 \\
 & 1 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 1 \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

R

$$\begin{array}{c|ccccccccc}
r & & & & & & & & & \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & t \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots &
\end{array}$$



S

R

$$\begin{tikzcd}
\Delta \arrow[r] & \boxed{\text{Correlated OT}} \arrow[dl] \arrow[dr] & r \arrow[l] \\
H(q_i), H(q_i \oplus \Delta) \arrow[d] & & H(q_i \oplus r_i\Delta) \\
\text{two random strings…} & & \text{of which R knows one, according to } r
\end{tikzcd}$$

Extending Oblivious Transfers Efficiently

Yuval Ishai[1], Joe Kilian[2], Kobbi Nissim[2*], and Erez Petrank[1**]

[1] Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel. {yuvali|erez}@cs.technion.ac.il
[2] NEC Laboratories America, 4 Independence Way, Princeton, NJ 08550, USA. {joe|kobbi}@nec-labs.com

Abstract. We consider the problem of extending oblivious transfers: Given a small number of oblivious transfers "for free," can one implement a large number of oblivious transfers? Beaver has shown how to extend oblivious transfers given a one-way function. However, this protocol is inefficient in practice, in part due to its non-black-box use of the underlying one-way function.
We give efficient protocols for extending oblivious transfers in the random oracle model. We also put forward a new cryptographic primitive which can be used to instantiate the random oracle in our constructions. Our methods suggest particularly fast heuristics for oblivious transfer that may be useful in a wide range of applications.

1 Introduction

Is it possible to base oblivious transfer on one-way functions? Partial answers to this question were given by Impagliazzo and Rudich [22] and Beaver [1]. Impagliazzo and Rudich [22] showed that a *black-box* reduction from oblivious transfer to a one-way function (or a one-way permutation) would imply P≠NP. They gave an oracle that combines a random function and a PSPACE oracle and proved that relative to this oracle one-way functions exist, but secret-key agreement is impossible. In other words, even an *idealized* one-way function (a random oracle) is insufficient for constructing secret-key agreement and hence oblivious transfer. A number of papers have continued this line of research and drew the limits of black-box reductions in cryptography, mapping the separations between the power of cryptographic primitives in relativized worlds [34, 15, 16, 25, 14].

It is not known whether a *non-black-box* reduction from oblivious transfer to one-way functions exists. Impagliazzo and Rudich's result strongly suggests that with the current knowledge in complexity theory we cannot base oblivious transfer on one-way functions. However, a remarkable theorem of Beaver [1] shows that a 'second-best' alternative *is* achievable – one-way functions are sufficient to *extend* a few oblivious transfers into many, i.e. it is possible to implement a large number of oblivious transfers given just a small number of oblivious transfers:

* Work partially done while the second author was at DIMACS, Rutgers University, 96 Frelinghuysen Road Piscataway, NJ 08854, USA.
** This research was supported by the E. AND J. BISHOP RESEARCH FUND.

Receiver chooses a large random matrix and secret shares it

Sender chooses random columns of the secret sharing

When each party transposes their matrix, the result is a large correlated secret share

Parties use a hash function H to generate random strings, which they can use to encrypt the messages they actually care about

Today's objectives

Introduce notion of OT extension

Show feasibility of OT extension

Construct efficient OT extension