Additively Homomorphic UC Commitments With Optimal Amortized Overhead

Ignacio Cascudo, Ivan Damgård,
Bernardo David, Irene Giacomelli,
Jesper Buus Nielsen, Roberto Trifiletti

Aarhus University

Structure

1. Introduction

Structure

1. Introduction
2. Our Contributions

Structure

1. Introduction
2. Our Contributions
3. A general framework

Structure

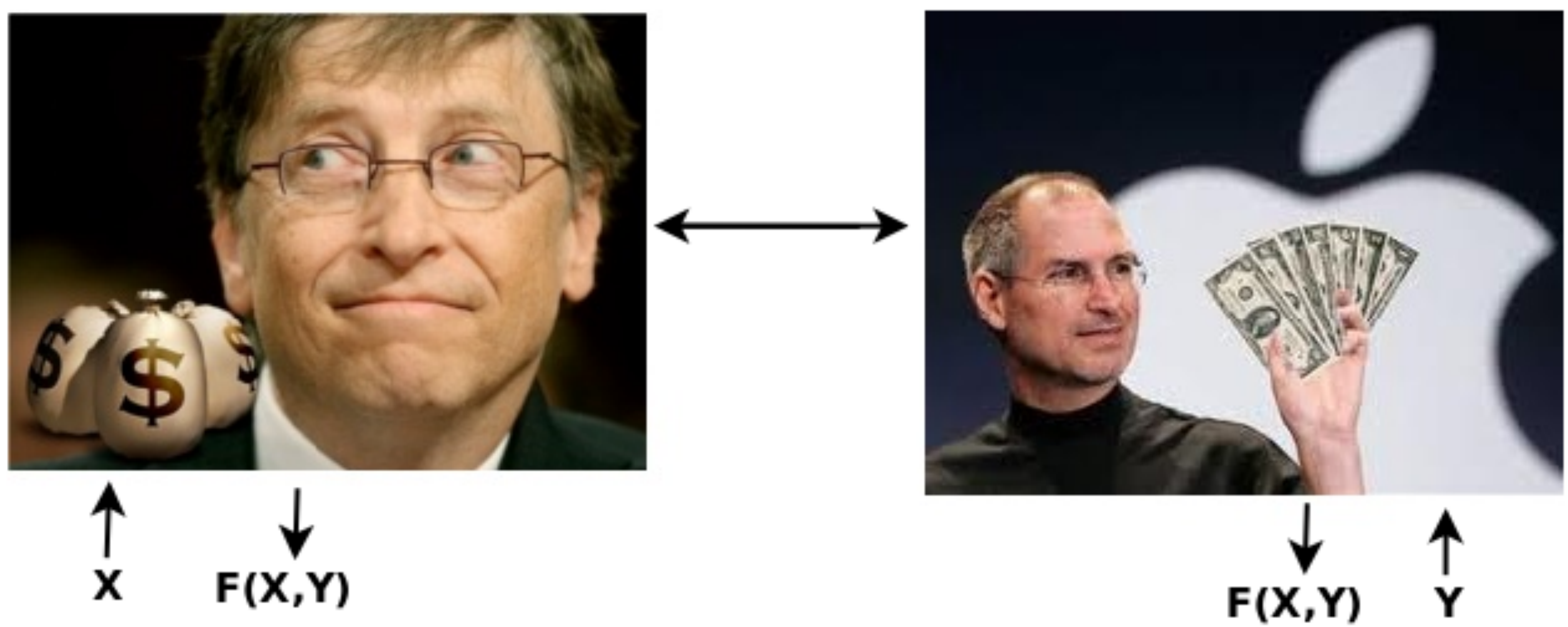
1. Introduction
2. Our Contributions
3. A general framework
4. Open Questions

Commitment Schemes

Commitment Schemes

Multiparty Computation

- The Millionaires’ Problem



Universal Composability

- Protocols remain secure in parallel concurrent executions and arbitrary composition.

Universal Composability

- Protocols remain secure in parallel concurrent executions and arbitrary composition.

- Commitments require setup assumptions [CF01].

Universal Composability

- Protocols remain secure in parallel concurrent executions and arbitrary composition.

- Commitments require setup assumptions [CF01].
- Commitments are complete [CLOS02].

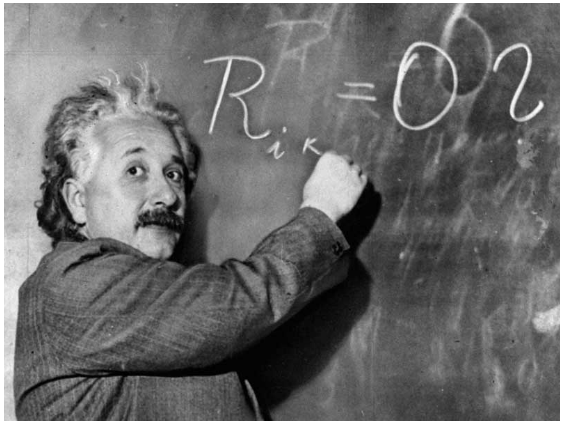
Related Works

- DDH based fast UC commitments: [Lindell11,BCPV13].
  - Use a Common Reference String (CRS).
  - High asymptotic communication and computational complexity.

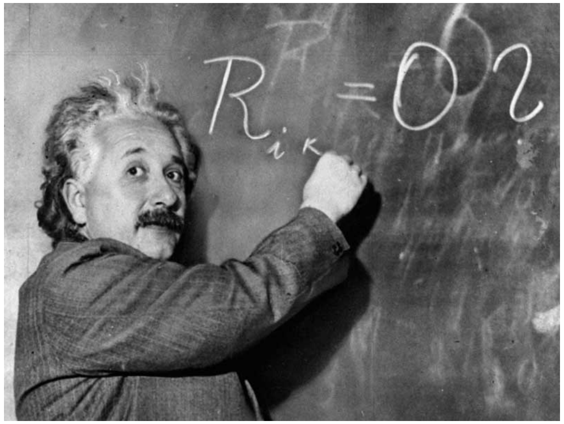
Related Works

- DDH based fast UC commitments: [Lindell11,BCPV13].
  - Use a Common Reference String (CRS).
  - High asymptotic communication and computational complexity.
- UC commitments with optimal rate: [DDGN14,GIKW14].
  - Use Oblivious Transfer as a setup assumption.
  - Require PRGs and general Linear Secret Sharing.

What do we do in theory?



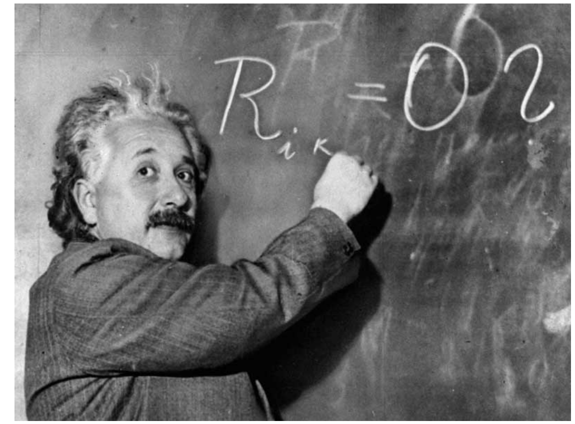
What do we do in theory?



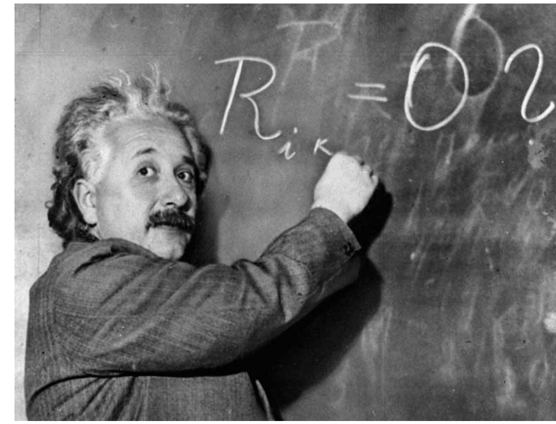
- Optimal communication

What do we do in theory?

- Optimal communication
- Additively Homomorphic

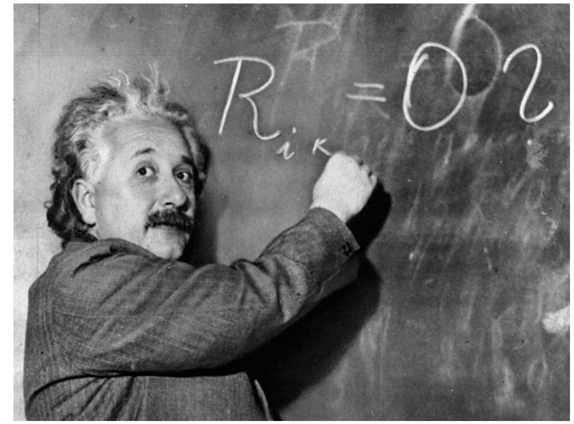


What do we do in theory?



- Optimal communication
- Additively Homomorphic
- Optimal computation 

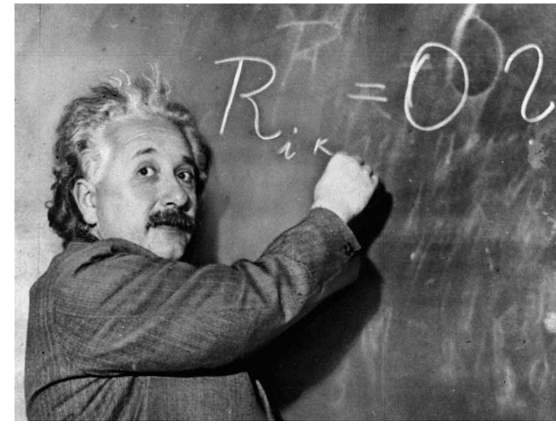
What do we do in theory?



- Optimal communication
- Additively Homomorphic
- Optimal computation 

- No need for general secret sharing 

What do we do in theory?

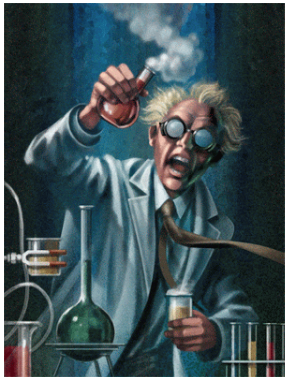


- Optimal communication
- Additively Homomorphic
- Optimal computation NEW!
- No need for general secret sharing NEW!

How do we do it?

ECC + PRG + OT

What do we do in practice?

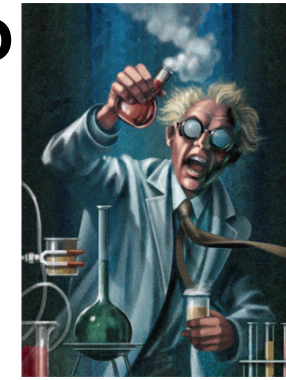


- Online Phase:

BCH [796,256,>=121] + PRG

2 Encodings: 1.5 μs

What do we do in practice?



- Online Phase:

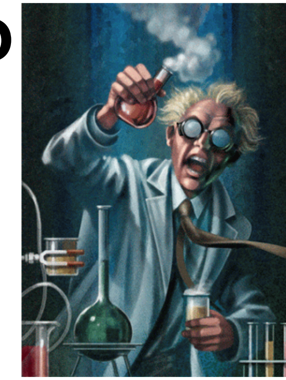
BCH [796,256,>=121] + PRG

2 Encodings: 1.5 μs

VS.

[Lindell11,BCPV13] -> 22 exponentiations: 8250 μs

What do we do in practice?



- Online Phase:

BCH [796,256,>=121] + PRG

2 Encodings: 1.5 μs

VS.

[Lindell11,BCPV13] -> 22 exponentiations: 8250 μs

=

- Practical scheme runs 5500 times faster

Practical Trade Offs…

Practical Trade Offs…

- No additive homomorphism.

Practical Trade Offs…

- No additive homomorphism.

- Setup phase cost:

796 OTs

8756 exponentiations using [PVW08]

398 [Lindell11,BCPV13] commitments

Building Blocks

- Error correcting codes:
  - Linear-time encodable codes [GI01,GI02,GI03,GI05,Spi96,DI14].

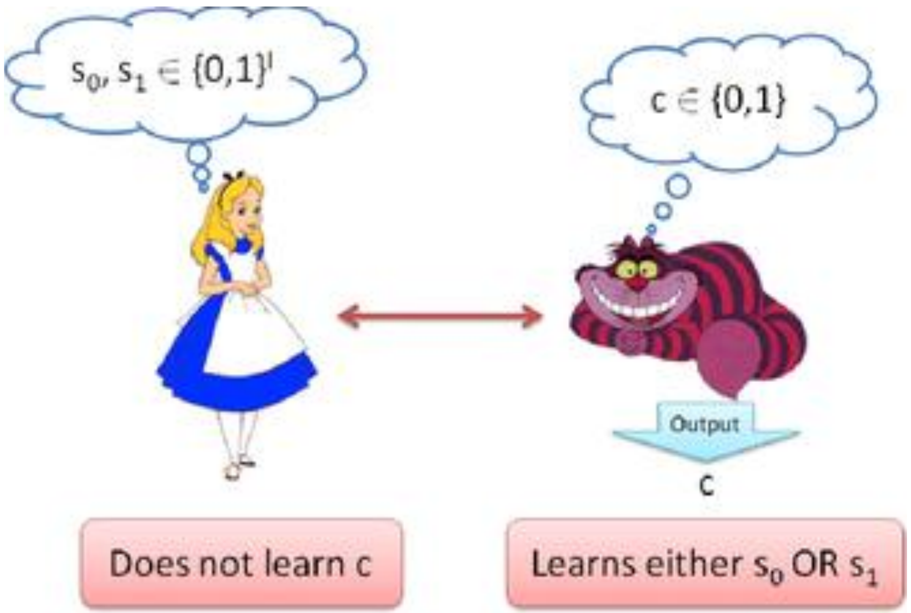
Building Blocks

- Error correcting codes:
  - Linear-time encodable codes [GI01,GI02,GI03,GI05,Spi96,DI14].
- UC Oblivious Transfer:
  - Any UC Oblivious Transfer protocol, e.g. [PVW08]

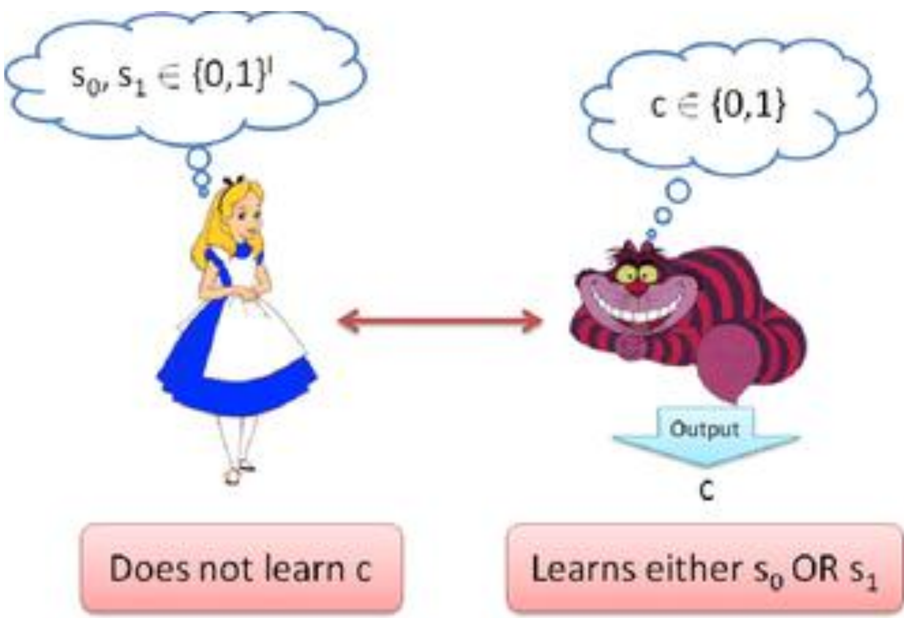
Building Blocks

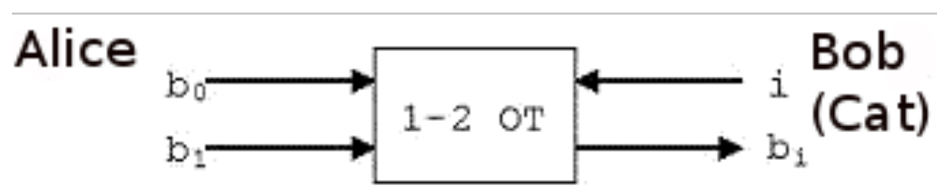
- Error correcting codes:
  - Linear-time encodable codes [GI01,GI02,GI03,GI05,Spi96,DI14].
- UC Oblivious Transfer:
  - Any UC Oblivious Transfer protocol, e.g. [PVW08]
- Pseudorandom Generator:
  - Linear-time PRG, e.g. [VZ12]

Oblivious Transfer

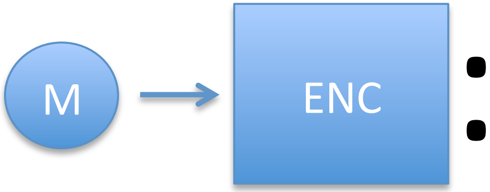


Oblivious Transfer

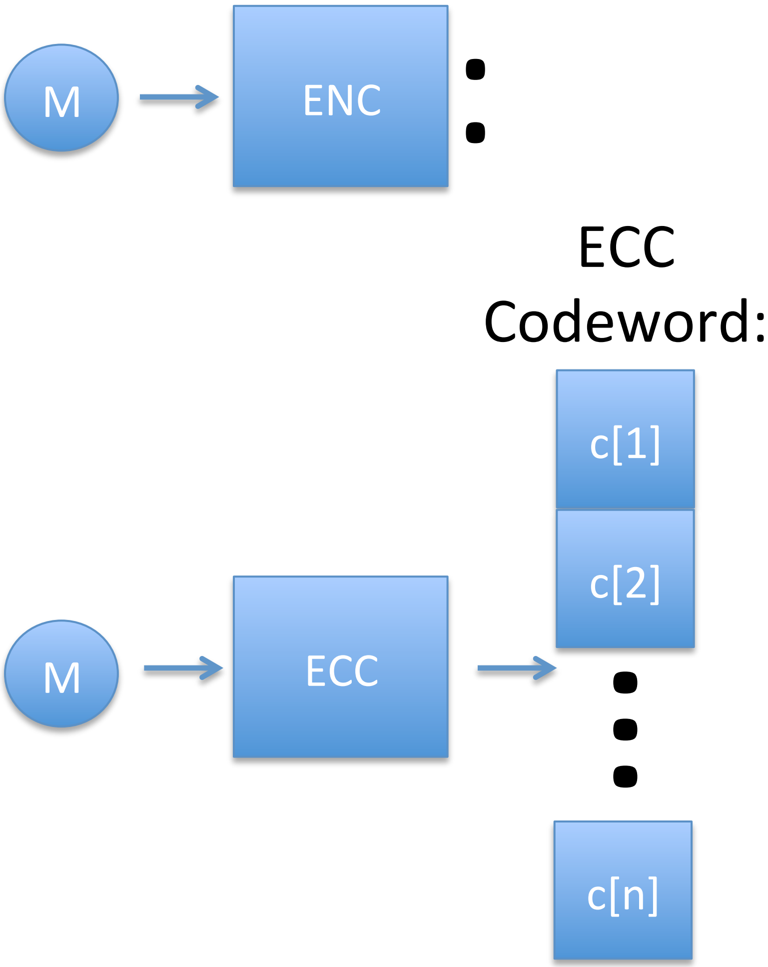




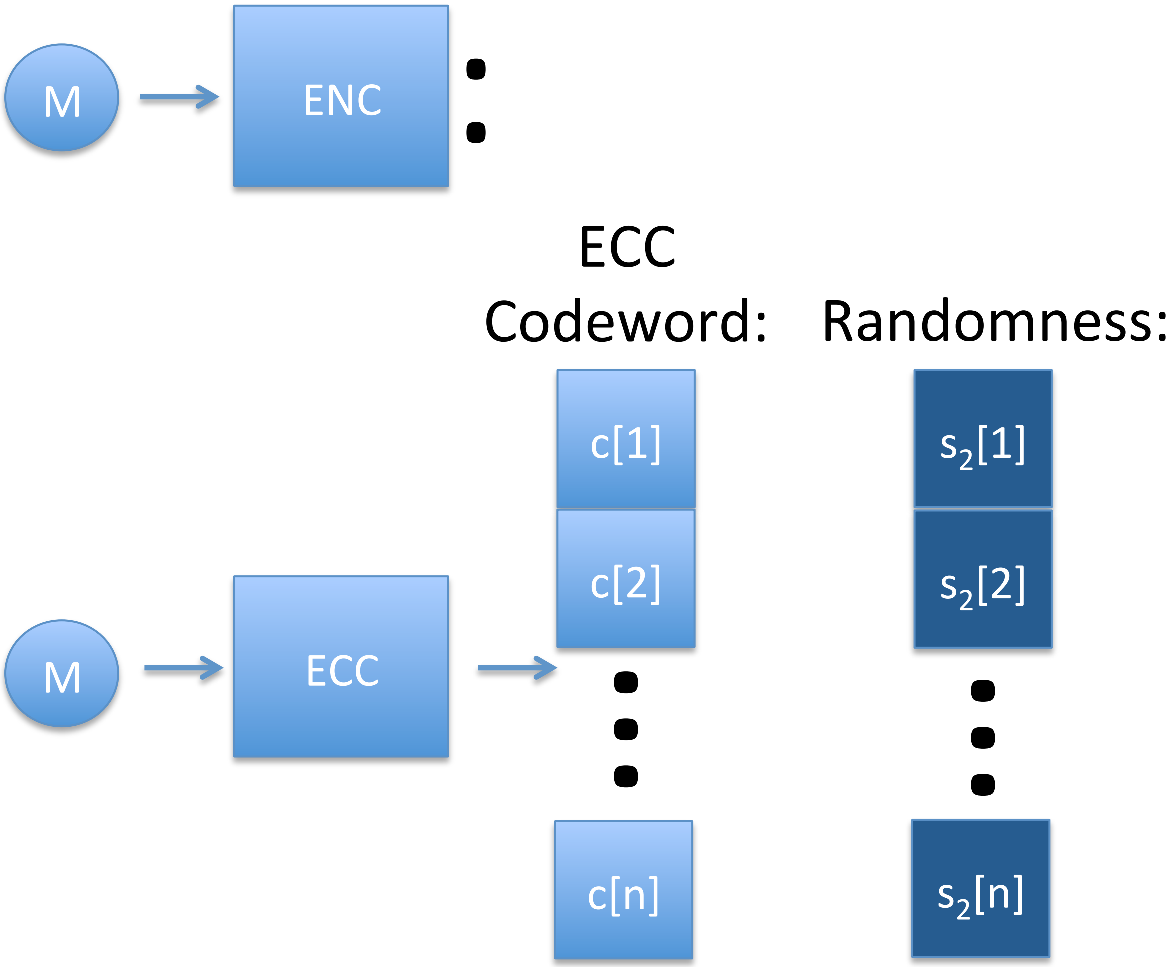
Encoding Scheme



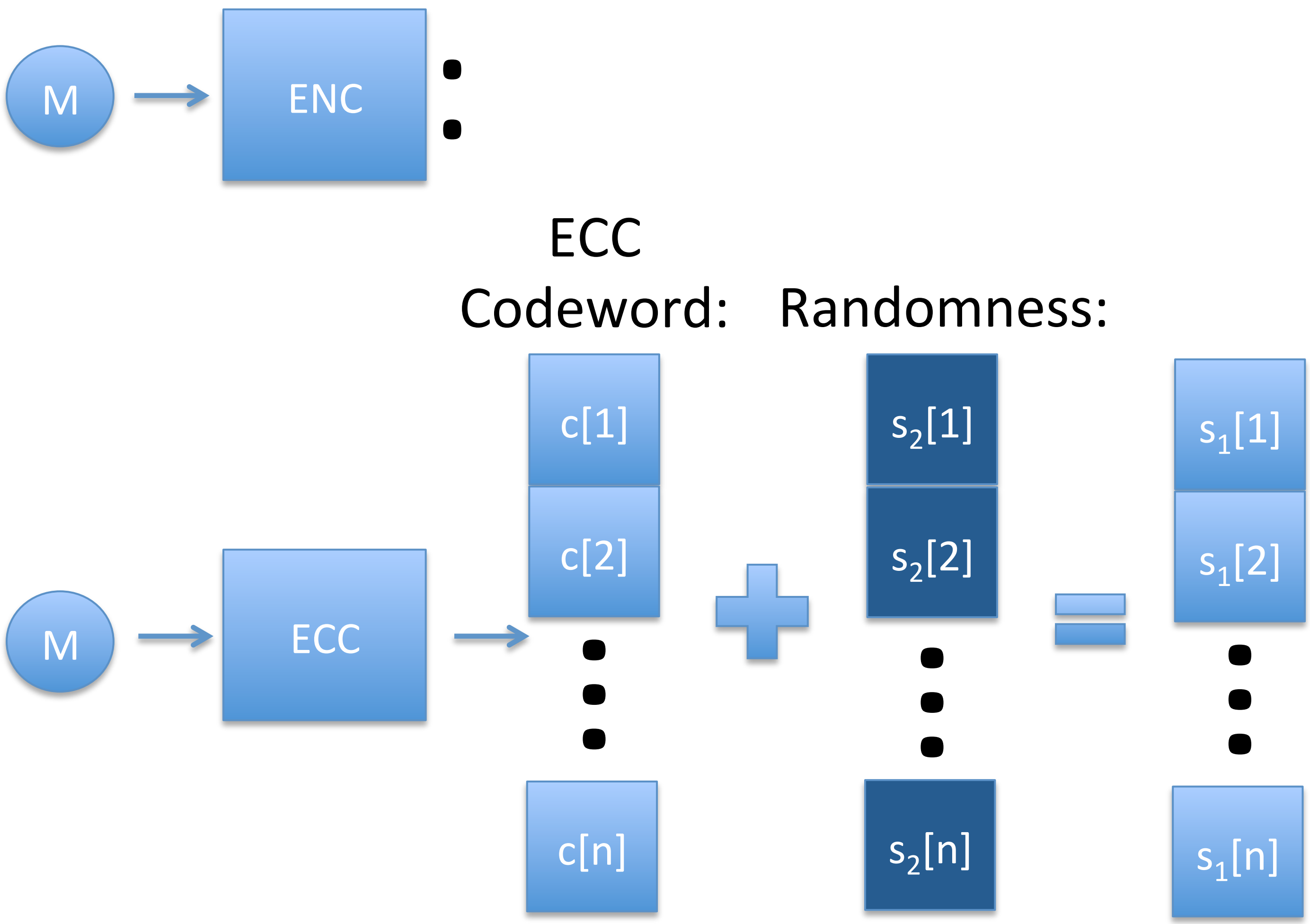
Encoding Scheme



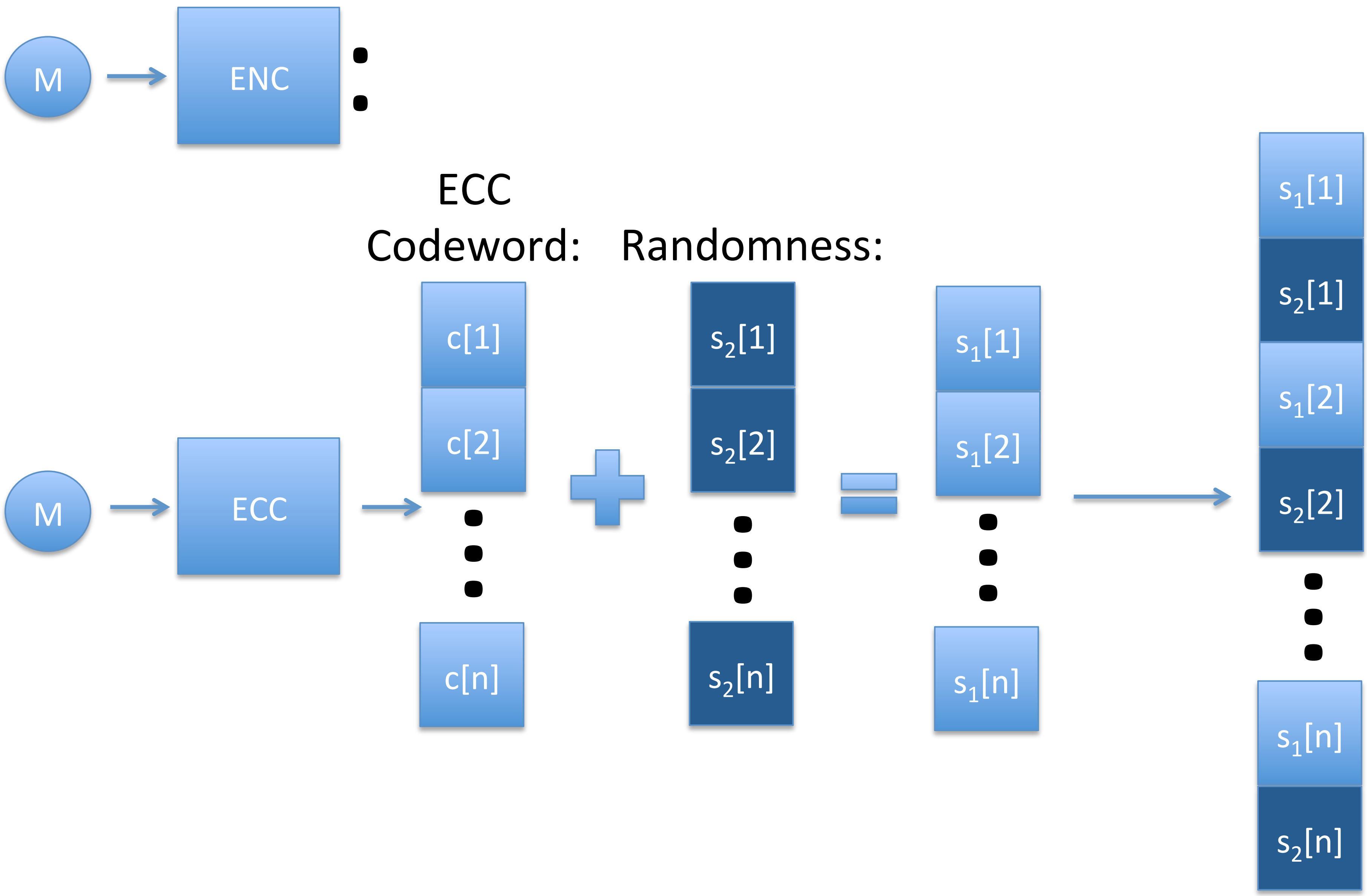
Encoding Scheme



Encoding Scheme



Encoding Scheme



General Framework

- Setup phase:

- Commitment/Open Phases:

General Framework

- Setup phase:
  - Independent from the inputs

- Commitment/Open Phases:

General Framework

- Setup phase:
  - Independent from the inputs
  - Constant number of OTs for unbounded number of commitments.

- Commitment/Open Phases:

General Framework

- Setup phase:
  – Independent from the inputs
  – Constant number of OTs for unbounded number of commitments.
  – Constant communication complexity.
- Commitment/Open Phases:

General Framework

- Setup phase:
  - Independent from the inputs
  - Constant number of OTs for unbounded number of commitments.
  - Constant communication complexity.
- Commitment/Open Phases:
  - Linear communication complexity.

General Framework

- Setup phase:
  - Independent from the inputs
  - Constant number of OTs for unbounded number of commitments.
  - Constant communication complexity.
- Commitment/Open Phases:
  - Linear communication complexity.
  - Only require a PRG and the encoding scheme.

General Framework

- Setup phase:
  - Independent from the inputs
  - Constant number of OTs for unbounded number of commitments.
  - Constant communication complexity.
- Commitment/Open Phases:
  - Linear communication complexity.
  - Only require a PRG and the encoding scheme.
  - Non interactive.

Setup Phase

Sender

Receiver

Random
Seeds:

Setup Phase

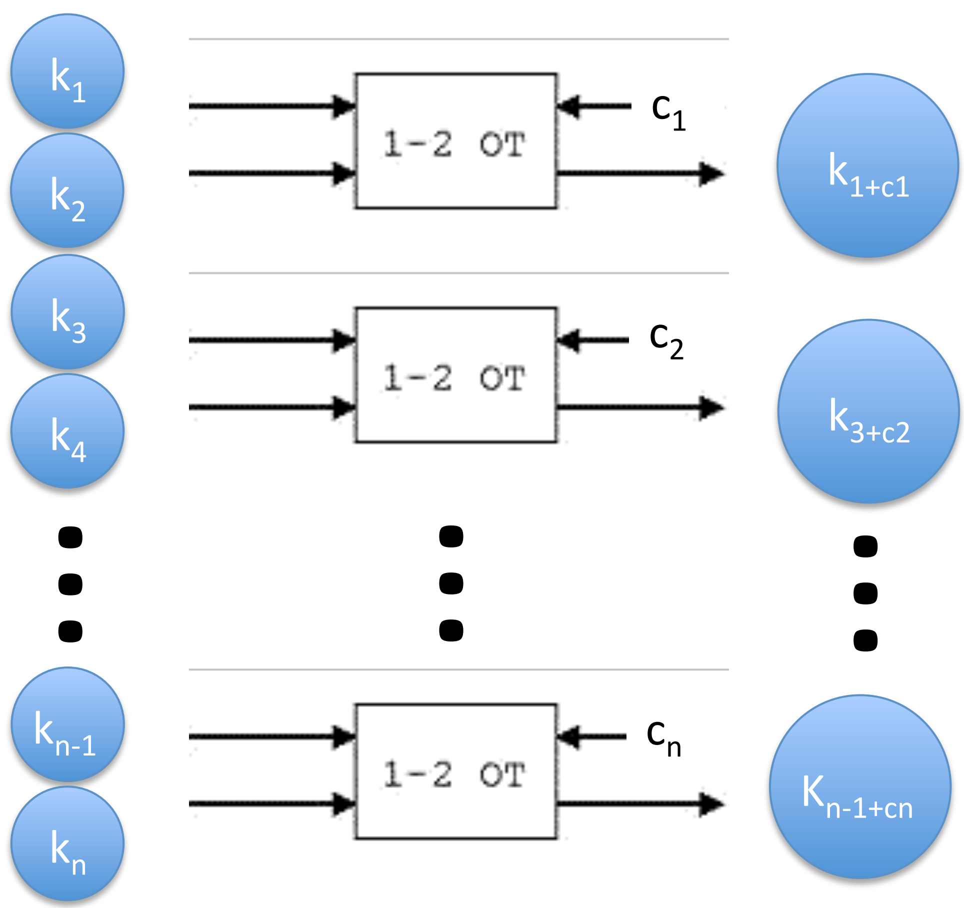
Sender

Receiver

Random Seeds:

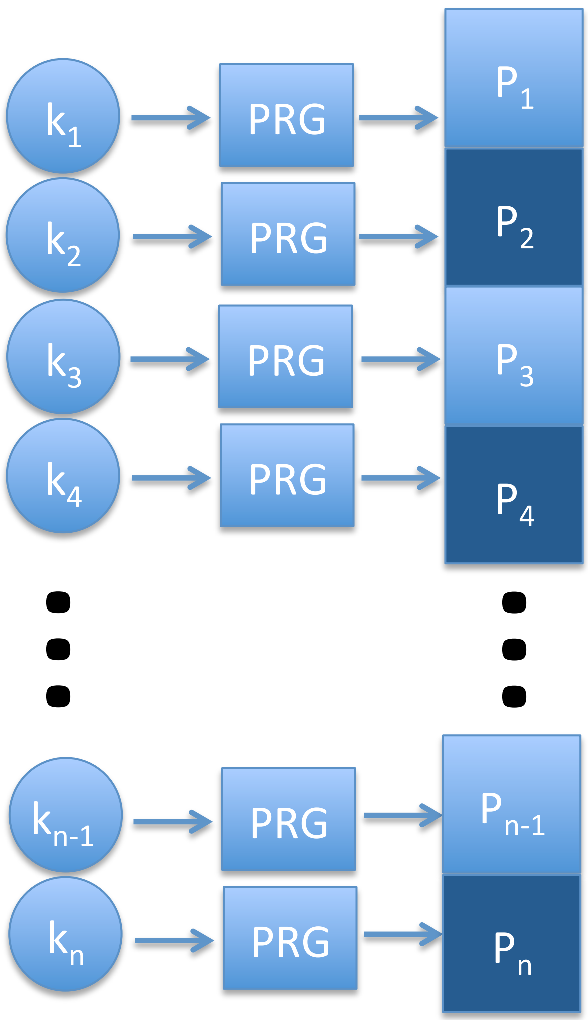
Random Choices:

Received Seeds:



Commitment Phase (Sender)

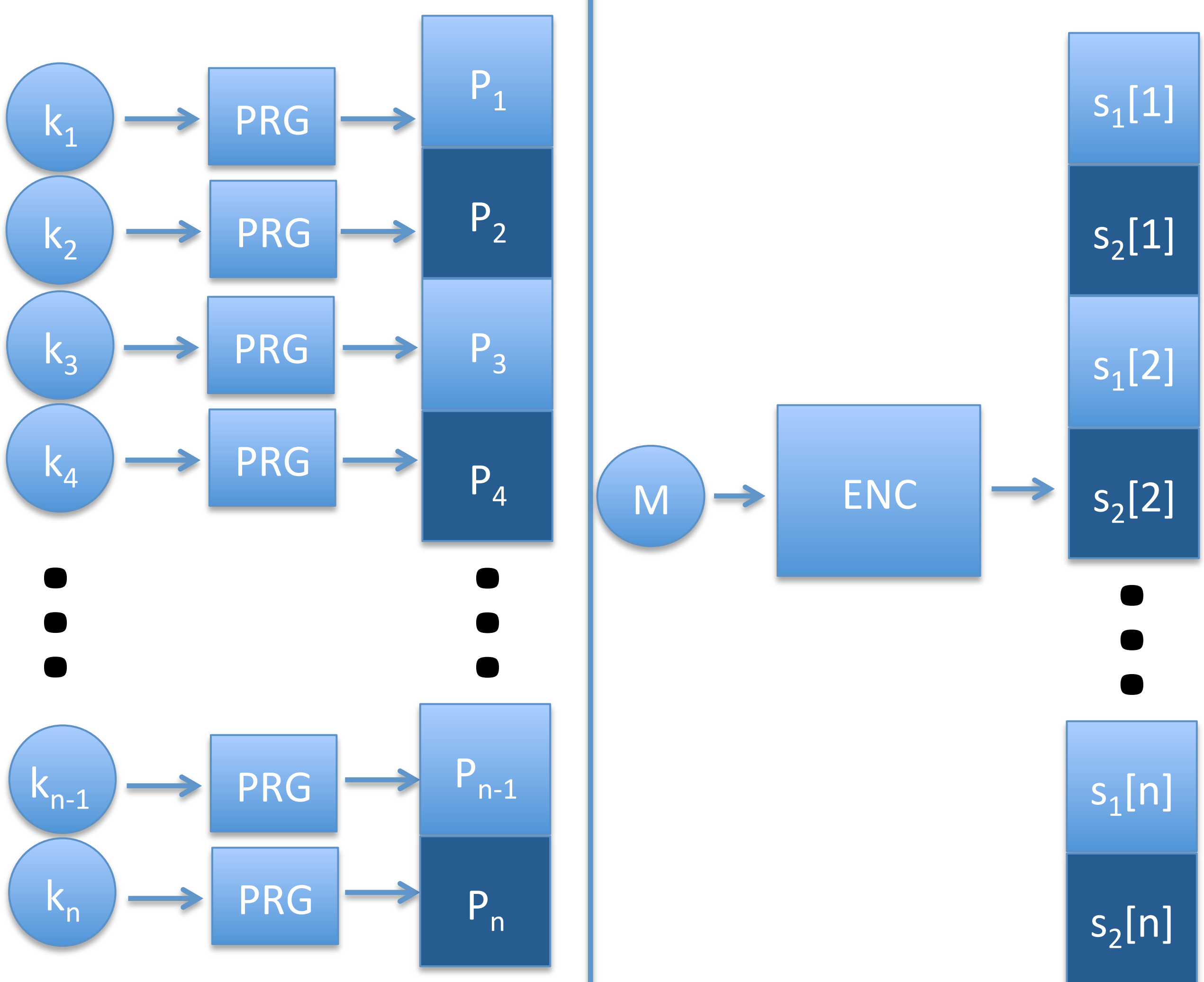
Generate one-time pads:



Commitment Phase (Sender)

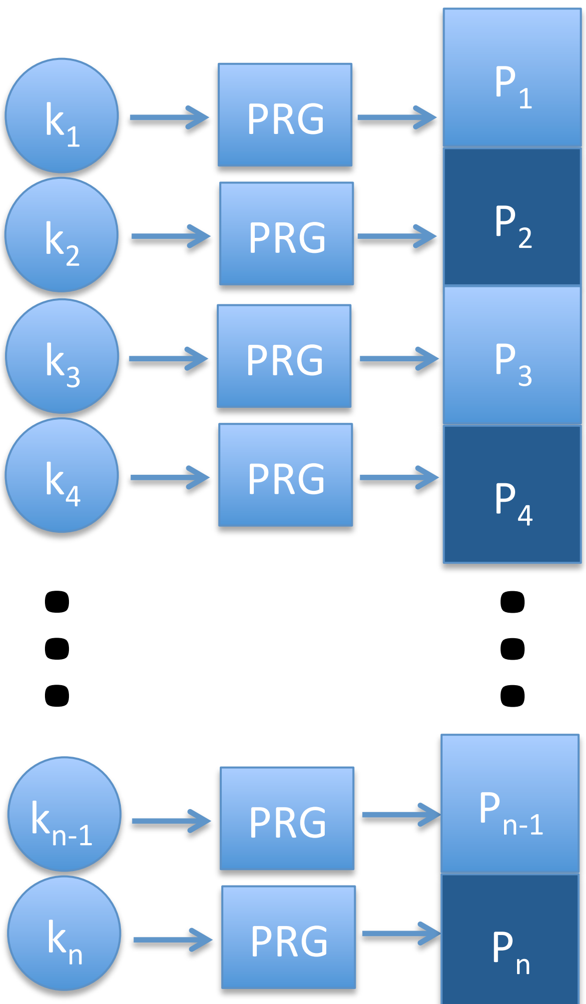
Generate one-time pads:

Encode messages and encrypt with one-time pads:

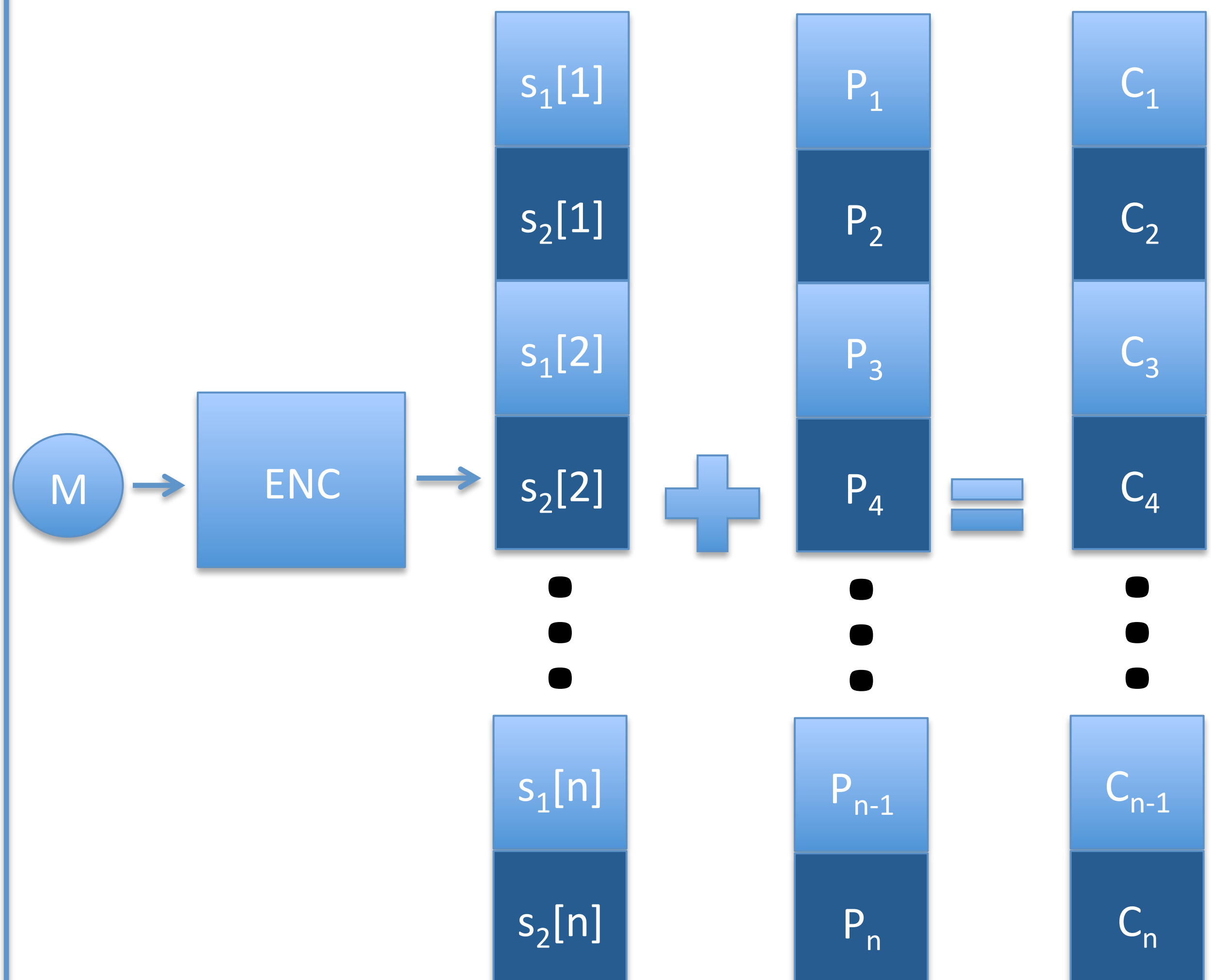


Commitment Phase (Sender)

Generate one-time pads:



Encode messages and encrypt with one-time pads:



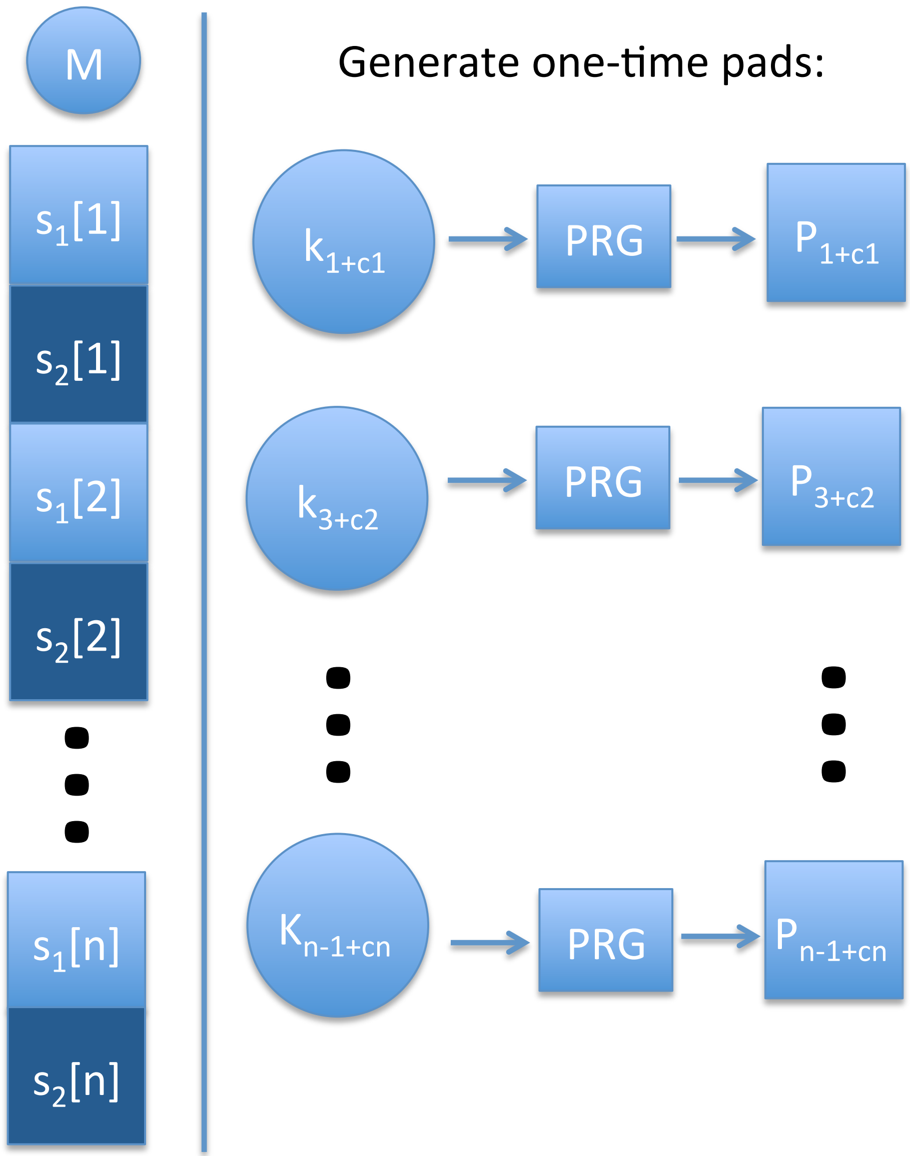
Open Phase (Receiver)

Opening Message:

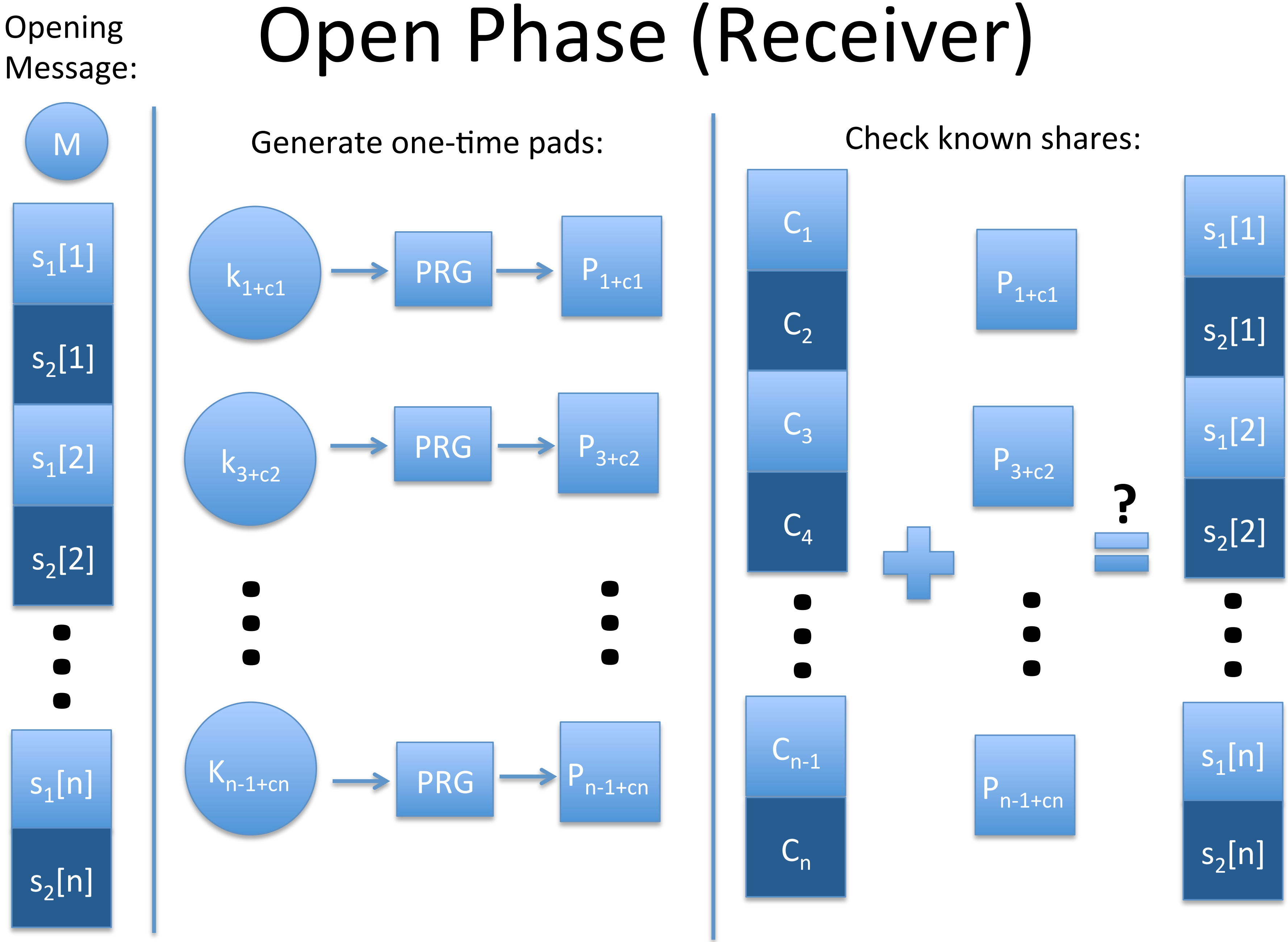
Open Phase (Receiver)

Opening Message:

Generate one-time pads:

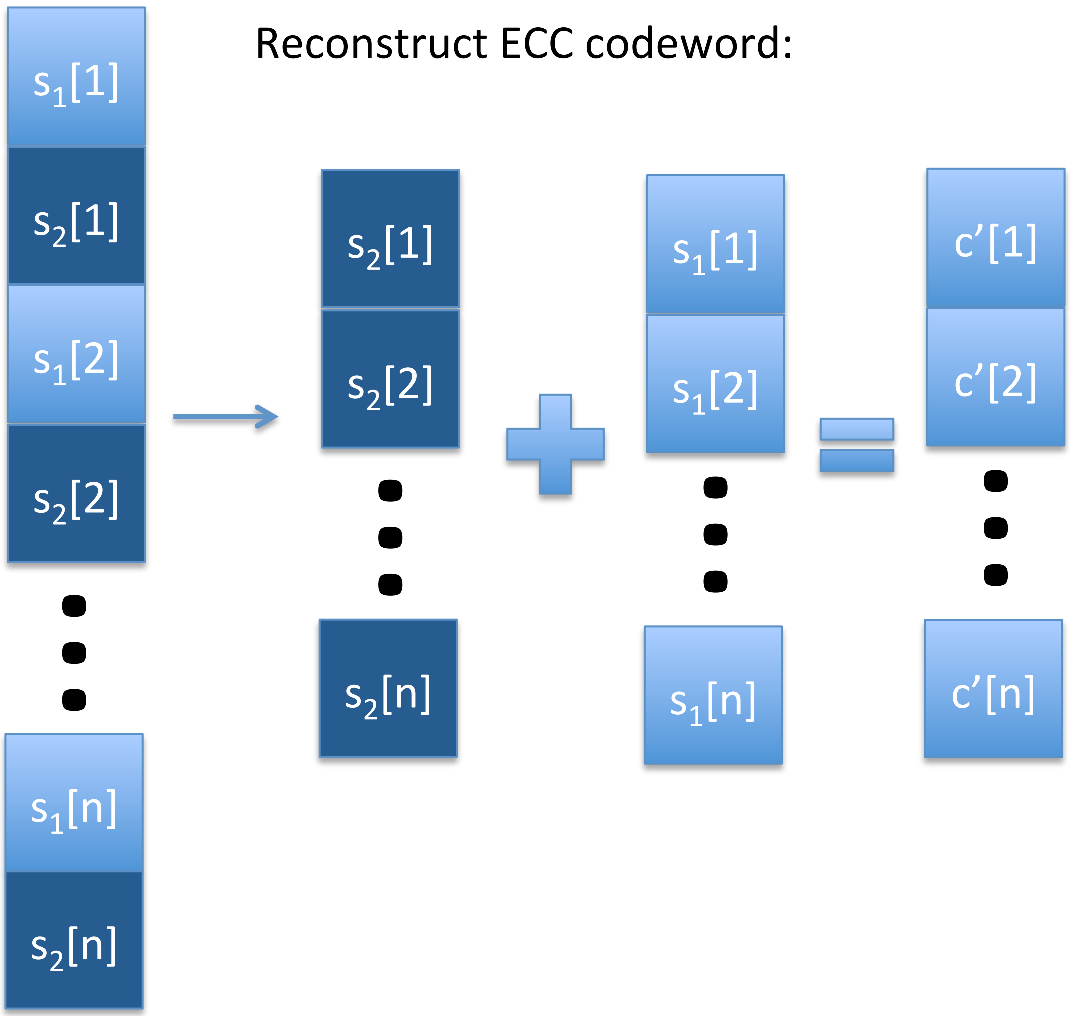


Open Phase (Receiver)

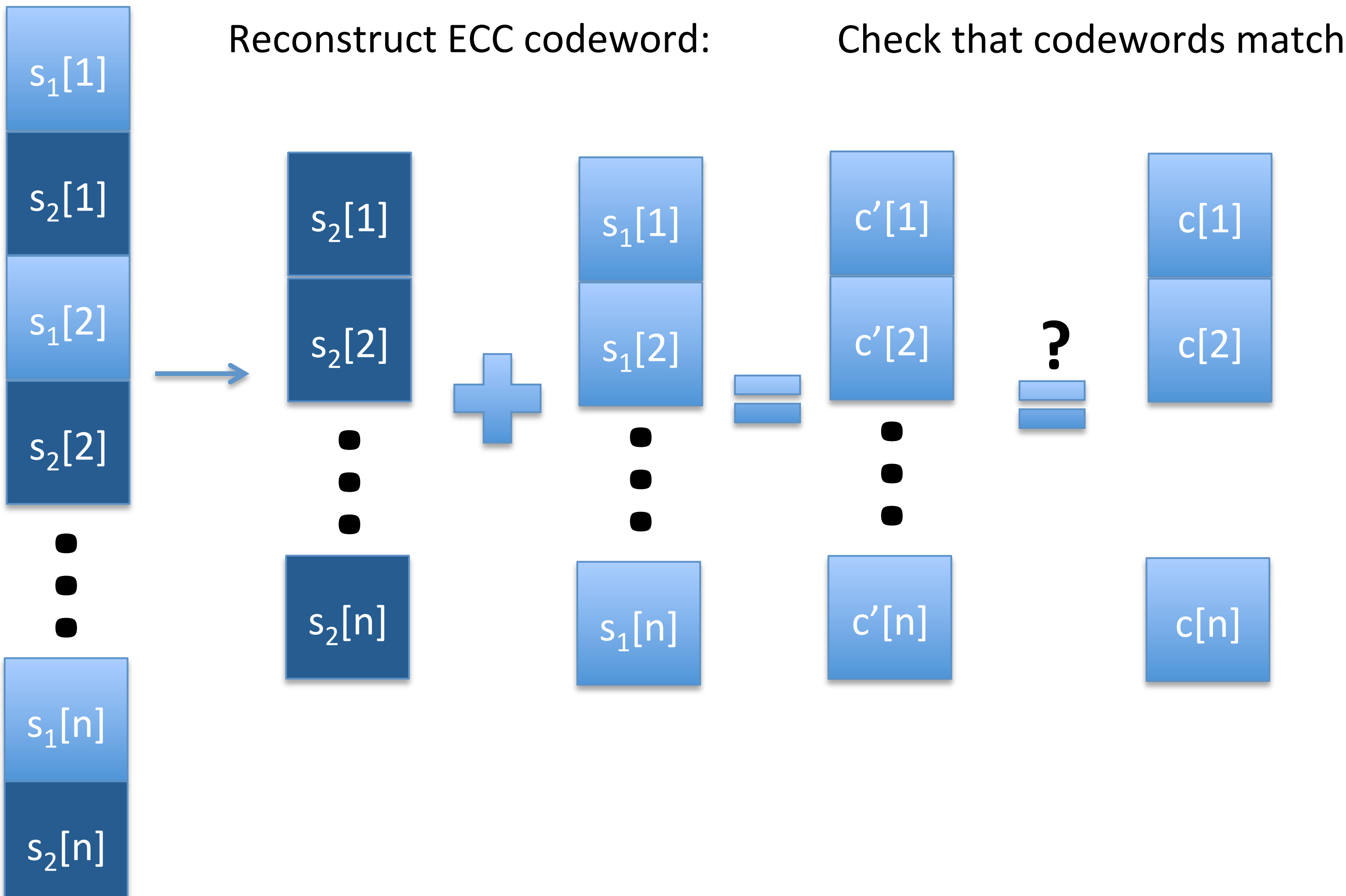


Open Phase (Receiver)

Reconstruct ECC codeword:



Open Phase (Receiver)



Open Problems

- Can we get optimal rate?
- Can optimal fully homomorphic commitments be constructed without general LSSS?
- Can we get additive homomorphism in this construction without VSS?
- Can we increase concrete efficiency in both setup and online phases?

THANK YOU!

READ THE FULL PAPER:

https://eprint.iacr.org/**2014/829**