Implementation of elliptic curve cryptography in constrained environments

Anthony Van Herrewege
Prof. Dr. Ir. I. Verbouwerende & Prof. Dr. Ir. B. Preneel

18 Februari 2009

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclussion

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclussion

Implement a compact hardware implementation of elliptic curve pairings.

Implement a compact hardware implementation of elliptic curve pairings.

- Program in GEZEL
- Optimize in VHDL
- Synthetize to FPGA/ASIC

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclussion

Overview

1. What?
2. Why?
3. How?

What?

- Calculations over elliptic curves

What?

- Calculations over elliptic curves
- Public key cryptography

What?

- Calculations over elliptic curves
- Public key cryptography
- Identity-based cryptography

Why?

- Identity-based cryptography
  - No public key lookup required:
    eg. $P =$ National identification number

Why?

- Identity-based cryptography
  - No public key lookup required:
    eg. $P =$ National identification number
  - Date-stamped encryption possible:
    eg. $P =$ Nin $+$ "20091223"

Why?

- Identity-based cryptography
  - No public key lookup required:
    eg. P = National identification number
  - Date-stamped encryption possible:
    eg. P = Nin + "20091223"
  - Drawbacks as well:
    key revocation, central authority, ...

Why?

- Identity-based cryptography
  - No public key lookup required:
    eg. P = National identification number
  - Date-stamped encryption possible:
    eg. P = Nin + "20091223"
  - Drawbacks as well:
    key revocation, central authority, ...
- Key strength [bits]:

RSA	
ECC	256

Underlying mathematics

- Discrete logarithm (DL) problem [hard]:

$$\text{Given: } g, h \in G : h \stackrel{?}{=} g^a \pmod{n}$$

Underlying mathematics

- Discrete logarithm (DL) problem [hard]:

$$\text{Given: } g, h \in G : h \stackrel{?}{=} g^a \pmod{n}$$

- Computational DL problem [hard]:

$$\text{Given: } g, g^a, g^b, \in G : h \stackrel{?}{=} g^{ab} \pmod{n}$$

Underlying mathematics

- Discrete logarithm (DL) problem [hard]:

$$\text{Given: } g, h \in G : h \stackrel{?}{=} g^a \pmod{n}$$

- Computational DL problem [hard]:

$$\text{Given: } g, g^a, g^b, \in G : h \stackrel{?}{=} g^{ab} \pmod{n}$$

- Decision DL problem [easy]:

$$\text{Given: } g, g^a, g^b, g^c \in G : g^c \stackrel{?}{=} g^{ab} \pmod{n}$$

Pairings

Q: What group satisfies CDL_{hard} and DDL_{easy}?
A: Elliptic curve pairing e:

$$e : G_1 \times G_1 \rightarrow G_2$$

Mapping needs to be bilinear, non-degenerate & efficiently computable. Several available pairings:

Weil, Tate, ate, eta, ...

Outline

1 Introduction

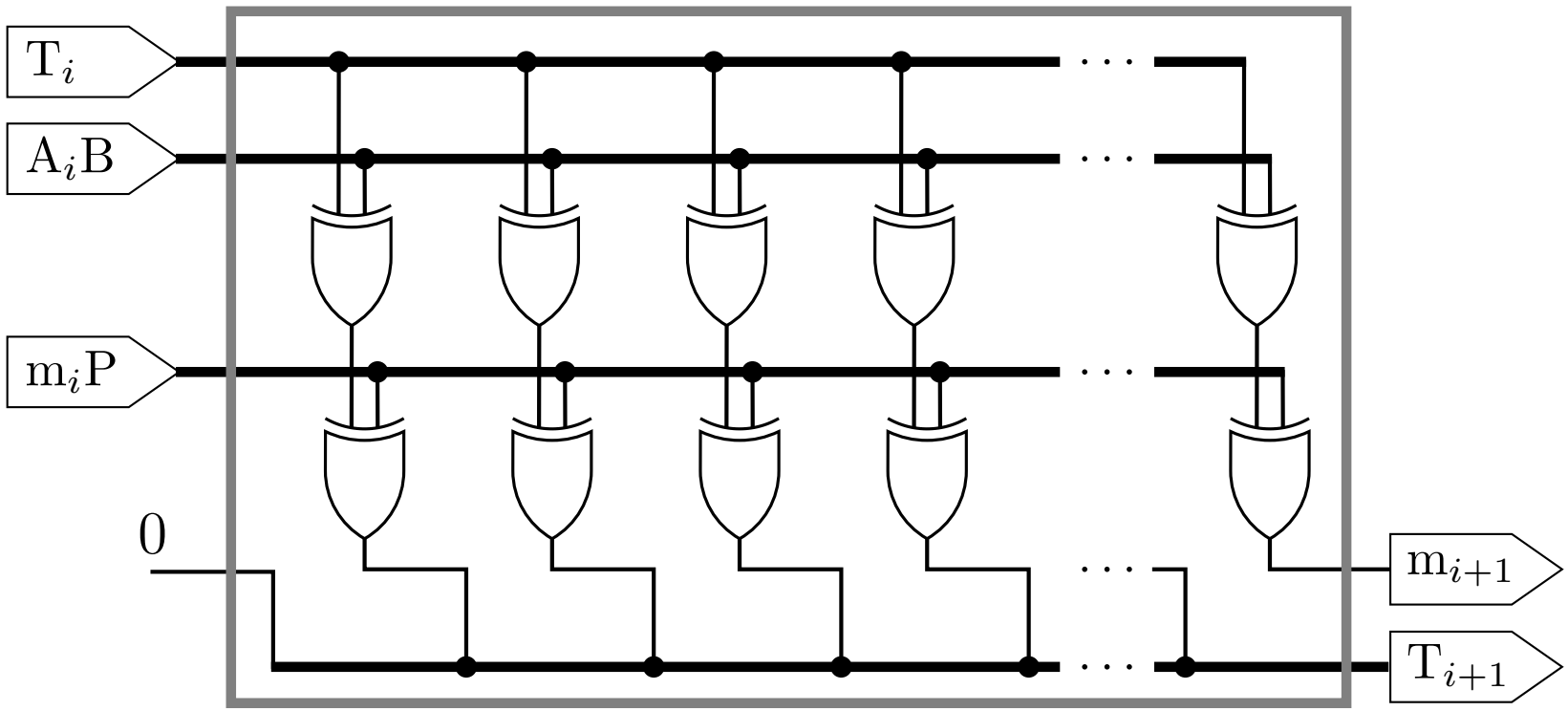
2 Elliptic Curve Pairings

3 Implementation

4 Conclussion

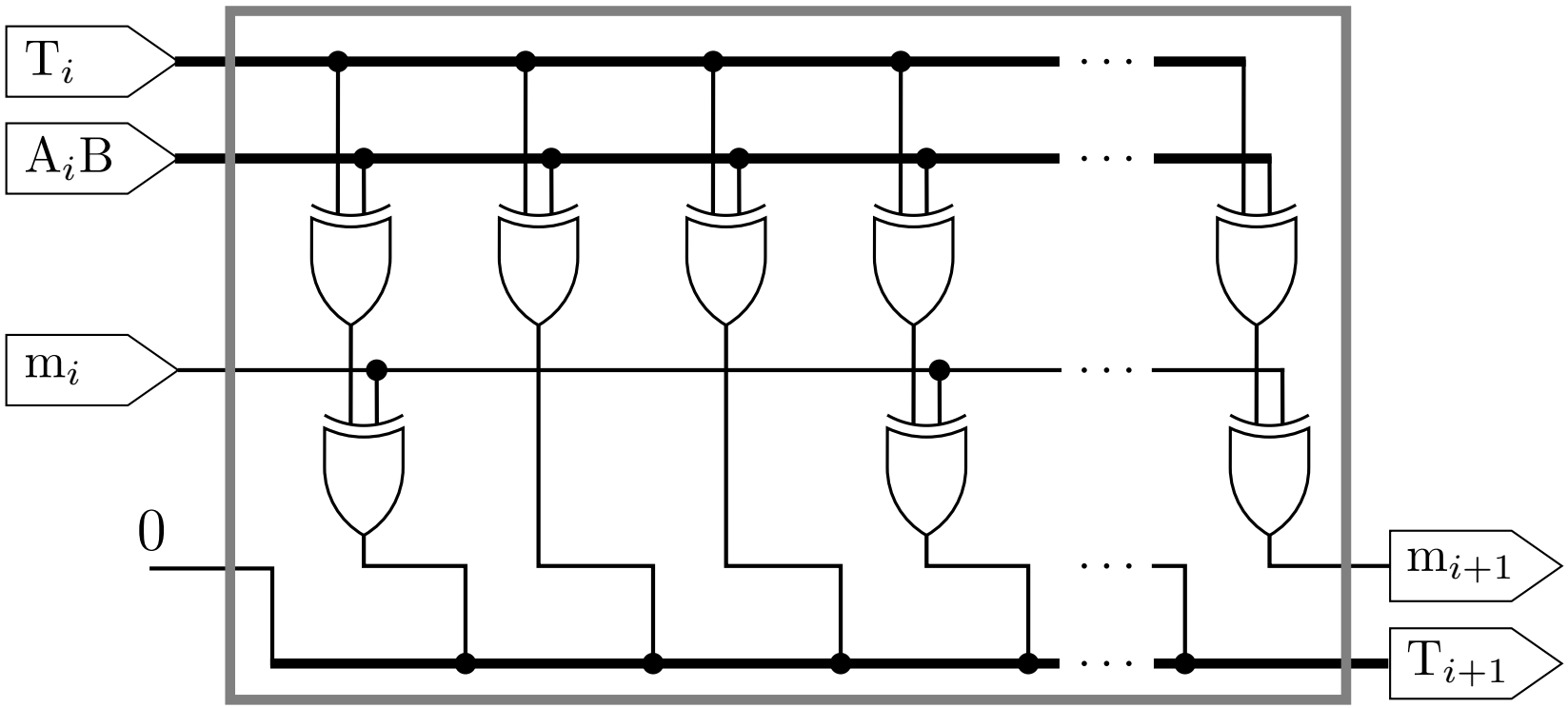
MALU

Modulo Arithmetic Logical Unit [general]:



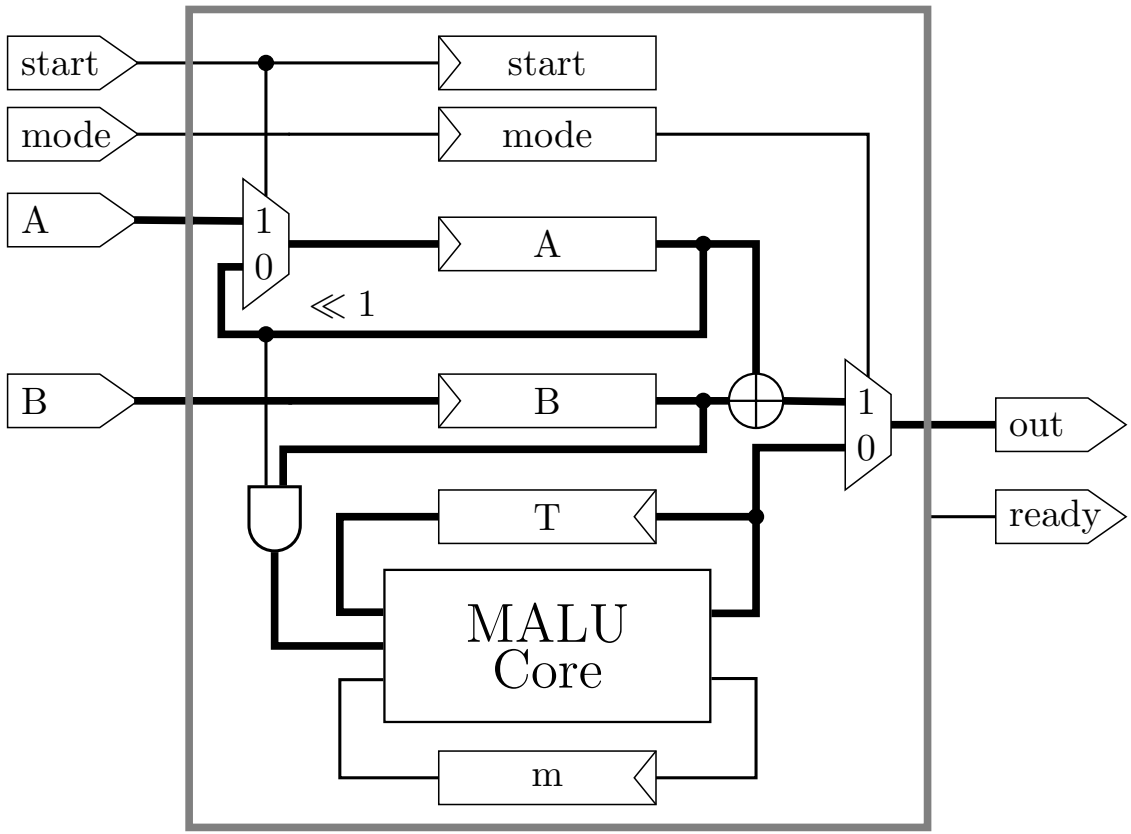
MALU

Modulo Arithmetic Logical Unit [optimized]:



Wrappers

Multiplication/Addition:



State of the art

Current available implementations:

Name	SW/HW	Speed
TinyTate	SW	5s

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclussion

Progress so far

- MALU

Progress so far

- MALU
- ECC functions

Progress so far

- MALU
- ECC functions
- Pairing functions

To do

- Bugfixing

To do

- Bugfixing
- Optimization (VHDL)

The end

Questions?