

Implementation of elliptic curve cryptography in constrained environments

Anthony Van Herrewege

Prof. Dr. Ir. I. Verbauwhede & Prof. Dr. Ir. B. Preneel

18 Februari 2009

Outline

- 1 Introduction
- 2 Elliptic Curve Pairings
- 3 Implementation
- 4 Conclusion

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclusion



Implement a compact hardware implementation of elliptic curve pairings.



ooooo

oooooo

ooo

Implement a compact hardware implementation of elliptic curve pairings.

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

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclusion

Overview

- 1 What?
- 2 Why?
- 3 How?

What?

■ Public key cryptography

What?

- Public key cryptography
- Identity-based cryptography

What?

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

Why?

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

Why?

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

Why?

- Identity-based cryptography
 - No public key lookup required:
eg. $P = \text{National identification number}$
 - Date-stamped encryption possible:
eg. $P = \text{Nin} + \text{"20091223"}$
 - Other positive aspects:
Fast (1 round) key establishment protocols
Ideal for eg. sensor networks

Why?

- Identity-based cryptography
 - No public key lookup required:
eg. $P = \text{National identification number}$
 - Date-stamped encryption possible:
eg. $P = \text{Nin} + \text{"20091223"}$
 - Other positive aspects:
Fast (1 round) key establishment protocols
Ideal for eg. sensor networks
 - Drawbacks as well:
no key revocation, still a central authority, ...

Why?

- Identity-based cryptography
 - No public key lookup required:
eg. $P = \text{National identification number}$
 - Date-stamped encryption possible:
eg. $P = \text{Nin} + \text{"20091223"}$
 - Other positive aspects:
Fast (1 round) key establishment protocols
Ideal for eg. sensor networks
 - Drawbacks as well:
no key revocation, still a central authority, ...
- Key strength comparison [bits]:
RSA 3072
ECC 256

How? - Underlying mathematics

- Discrete logarithm (DL) problem [hard]:

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

How? - Underlying mathematics

- Discrete logarithm (DL) problem [hard]:

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

- Computational DL problem [hard]:

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

How? - 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]:

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

How? - 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

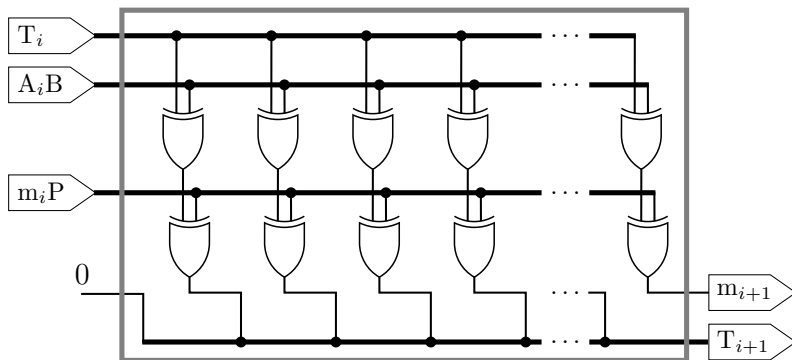
2 Elliptic Curve Pairings

3 Implementation

4 Conclusion

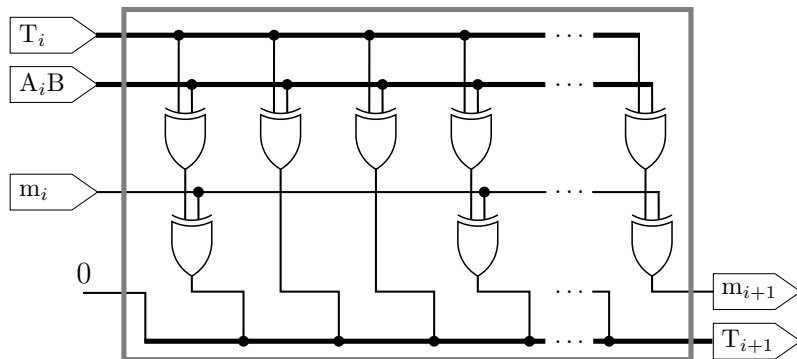
MALU

Modulo Arithmetic Logical Unit [general]:



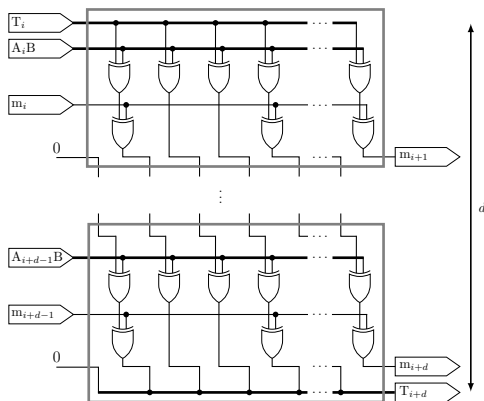
MALU

Modulo Arithmetic Logical Unit [optimized]:



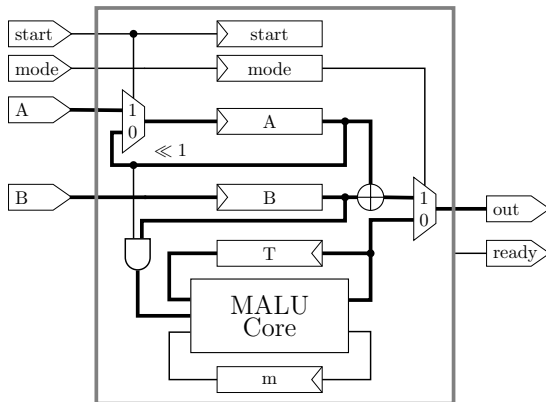
MALU

Modulo Arithmetic Logical Unit [optimized; d-bits wide]:



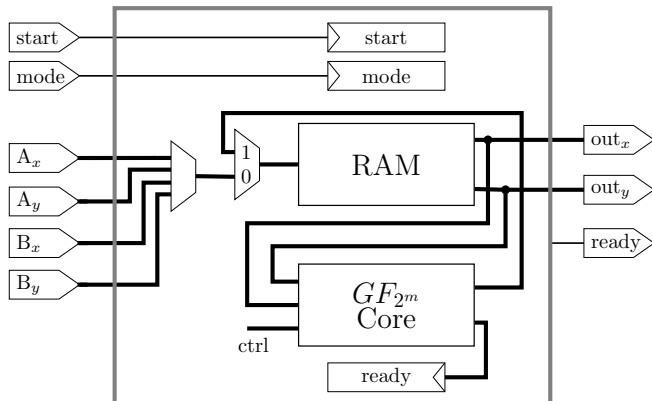
Wrappers - GF_{2^m}

GF_{2^m} Multiplication/Addition:



Wrappers - ECC

ECC Point Addition/Doubling:



State of the art

Some currently available implementations:

Name	Platform	Field	Speed
TinyTate	ATMega128L [7.4Mhz]	$\mathbb{F}_{2^{256}}$	30.2s
TinyPBC	ATMega128L [7.4Mhz]	$\mathbb{F}_{2^{256}}$	5.45s
Hankerson	P4 [2.8Ghz]	$\mathbb{F}_{2^{1223}}$	0.07s
Hankerson	P4 [2.8Ghz] (SSE)	$\mathbb{F}_{2^{1223}}$	0.03s

Outline

1 Introduction

2 Elliptic Curve Pairings

3 Implementation

4 Conclusion

Progress so far

■ MALU

Progress so far

- MALU
- ECC functions

Progress so far

- MALU
- ECC functions
- Pairing functions (partial)

To do

- Complete pairing functions

To do

- Complete pairing functions
- Bugfixing

To do

- Complete pairing functions
- Bugfixing
- Optimization (VHDL)

The end

Questions?