



# An Efficient General Purpose Elliptic Curve Cryptography Module for Ubiquitous Sensor Networks

---

Christof Paar, Axel Poschman, Leif Uhsadel  
Ruhr-Universität Bochum, Germany

June, 12th 2007

SPEED  
Software Performance Enhancement  
for Encryption and Decryption



# Outline

- Motivation
- Platform
- Bottlenecks I
- Algorithmic Setup
- Bottlenecks II
- Implementation
- Results

# Why high speed?

past



Mainframe  
(n : 1)

present



Personal  
(1 : 1)

future



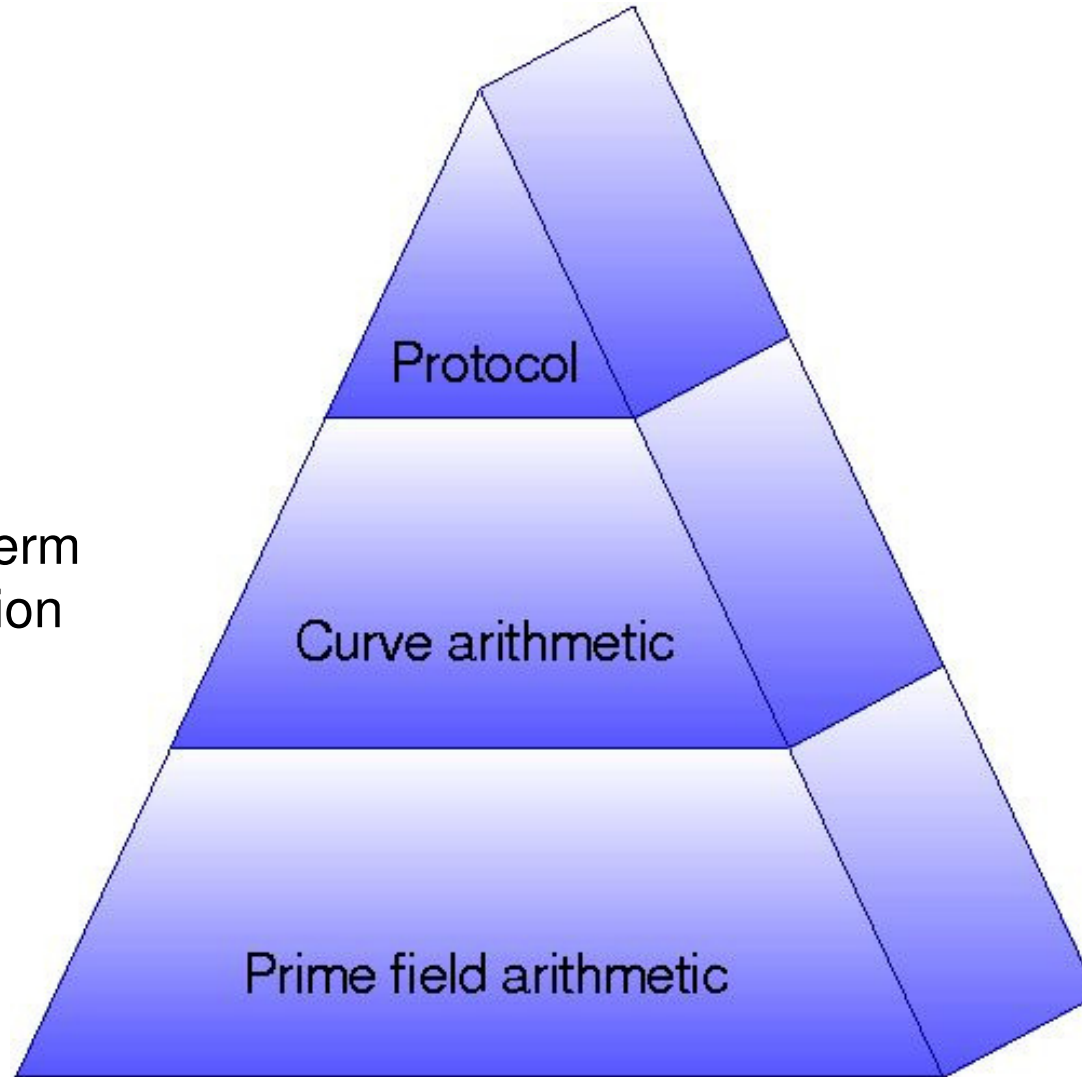
Ubiquitous  
(1 : n)

Ubiquitous = wireless + embedded + **energy efficient**  
= constrained in CPU, memory, battery



# General Purpose Module

77% long term  
multiplication





Asymmetric Cryptography is quite useful for key distribution



Asymmetric Cryptography is supposed to be too demanding for constrained devices



SUN: Fast but not public



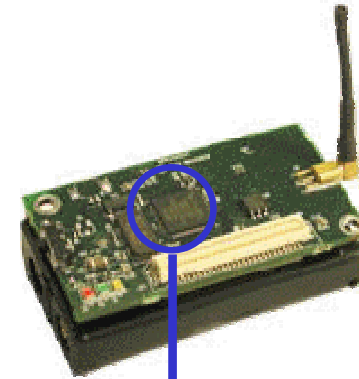
TinyECC: Open source

## Goal:

- Fast and free prime field for constrained devices
- Main task: **efficient 160-bit modular multiplication**

## MicaZ

Sensor Board	Temperature, Light, Humidity,...
Power Supply	2xAA Batteries (1000 mAh)
Microcontroller	8-bit Atmega128L $\mu$ Processor

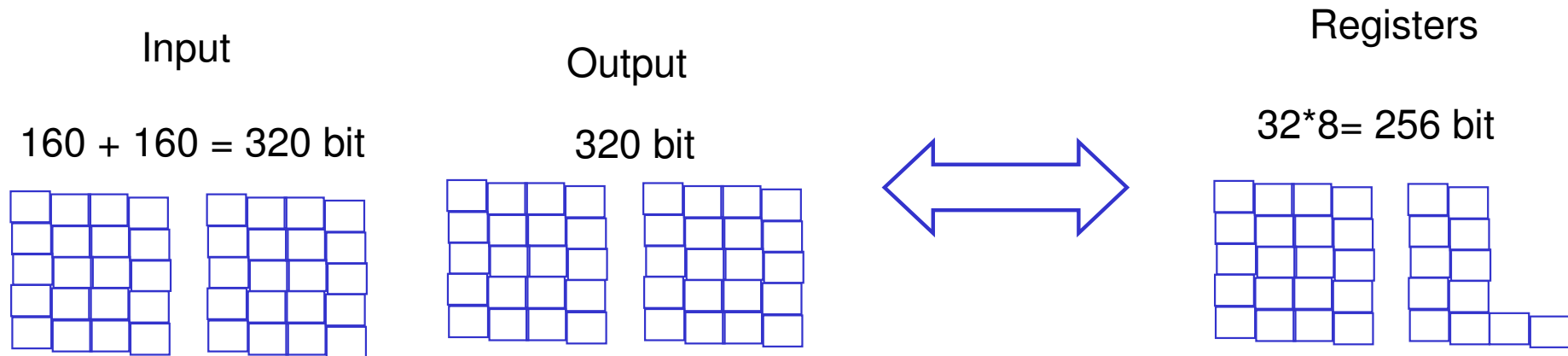


## ATMega128L

Microcontroller	8-bit Atmega128L $\mu$ Processor
Max Frequency	7,37 Mhz
Flash Memory	128 KB
Configuration EEPROM	4 KB
SRAM	4 KB
Registers	32
Measurement Flash	512 KB



# Bottleneck SRAM access



- **SRAM operation: 2 clock cycles**
- **8-bit multiplication: 2 clock cycles**



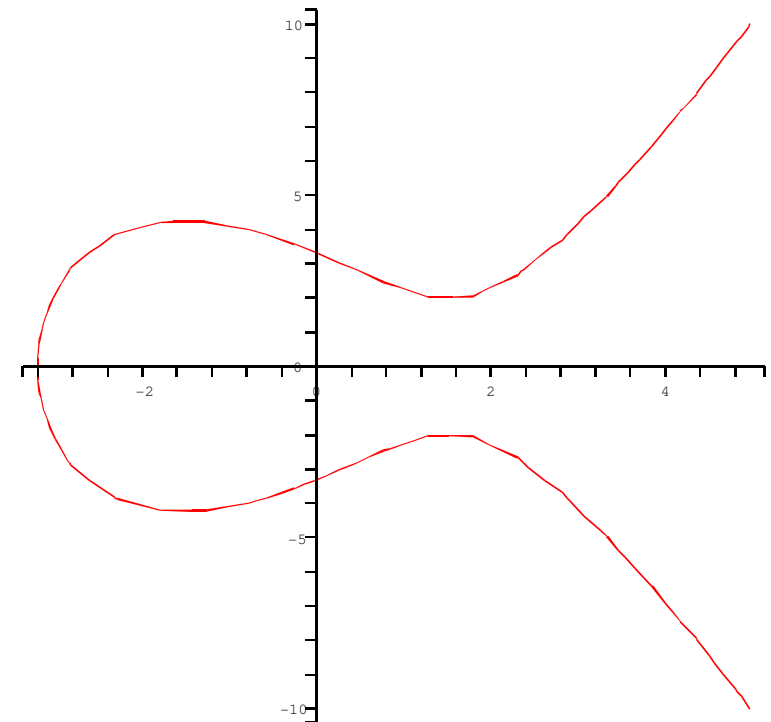
# Algorithmic Setup

- Primefield based on a 160-bit Mersenne Prime

## Alternatives:

- Karatsuba Offman
  - trade 1 mul for 4 add
  - recursive nature
- Hybrid Schoolbook
  - optimized for low SRAM access

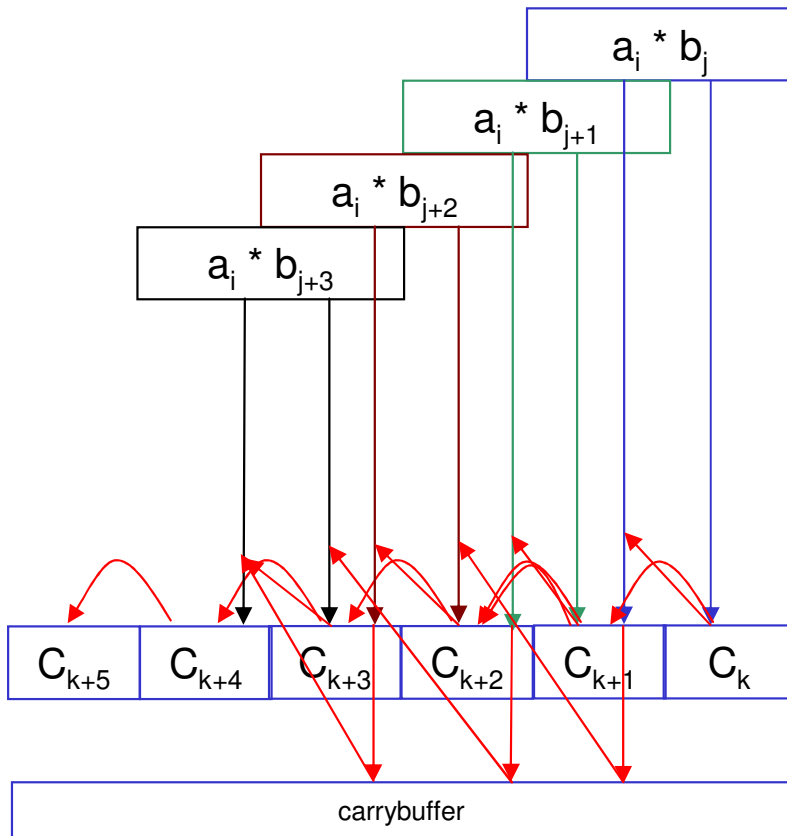
Standard curve secp160r1





# Implementation

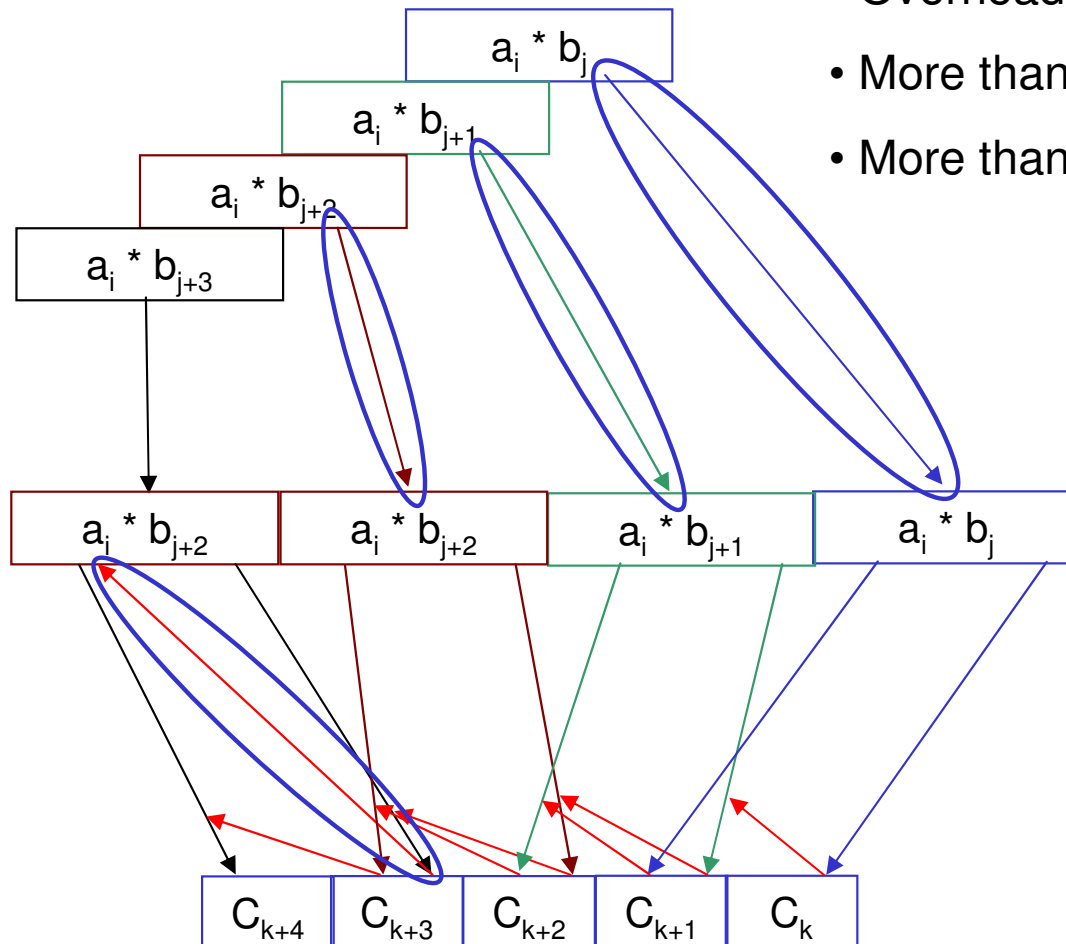
## Why are carries a bottleneck ..?



- Addition overwrites carry flag
- Add with carry not possible
- Carry must be buffered

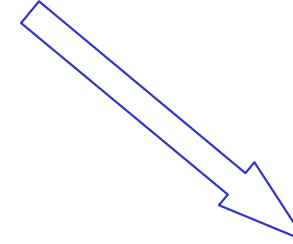
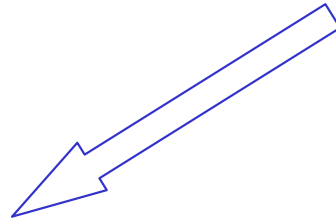
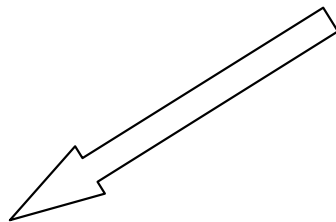
- Overhead per 8-bit multiplication:
- More than 3 clock cycles
- 400 8-bit multiplications are done

# Implementation Handling carries



- Overhead per 4 8-bit multiplication:
- More than 4 clock cycles
- More than 1 clock cycle per 8-bit multiplication

160-bit Integer Multiplication	
sun	this work
assembly	assembly
3106 clock cycles	2913 clock cycles
0.39 ms @ 8 MHz	0.36 ms @ 8 MHz



Binary EC multiplication	
sun	this work
assembly	C
0.81s	1.15s



Sliding Window (w=4) EC multiplication	
tinyecc (ECDSA sig)	continued project
hybrid	C
1.9s	0.89s





- Questions?
- Comments?

[uhsadel@crypto.rub.de](mailto:uhsadel@crypto.rub.de)