



Efficient Implementation of Pairing on Sensor Nodes

Tsukasa Ishiguro, Masaaki Shirase, *Tsuyoshi Takagi

Future University Hakodate, Japan



Sensor Node

- **MICAz**

- CPU: ATmega128L at 7.37MHz
- ROM: 128kB
- SRAM: 4kB
- 8-bit CPU
- Size: 62x35x27 (mm)



Crossbow (<http://www.xbow.com>)

- **TinyOS:** operating system

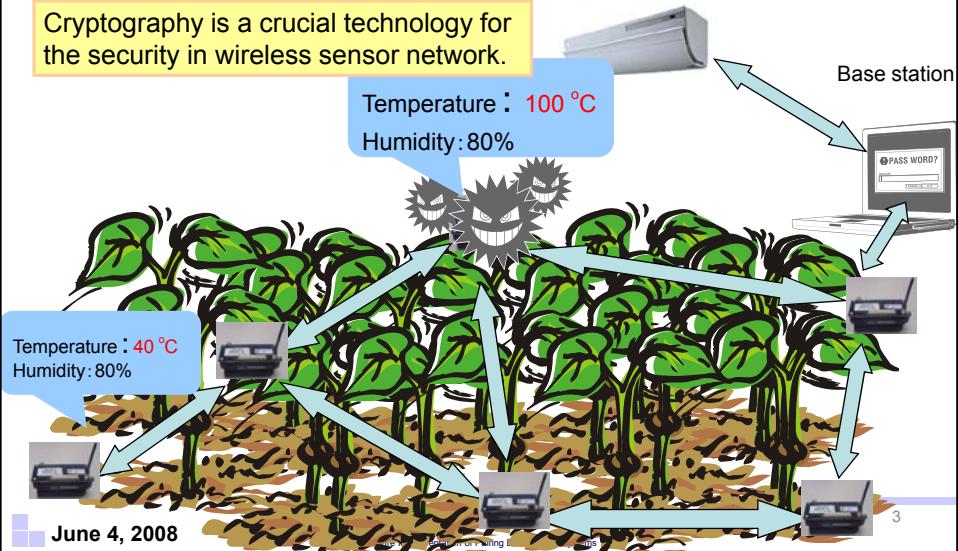
- **NesC:** C programming language





Security in Wireless Sensor Network

Cryptography is a crucial technology for the security in wireless sensor network.



Pairing Based Cryptography

- Novel Cryptographic Applications
 - ID Based Cryptography [Sakai et al. 2000, Boneh et al. 2001]
 - Efficient Broadcast Encryption [Boneh et al. 2005]
 - Keyword Searchable Encryption [Boneh et al. 2004]
 - etc
- Goal of This Research
 - Implementation of Pairing on ATmega128L





Recent Implementations of PKC on ATmega128L using NesC

- TinyPK, 14.5 sec, Watro et al. 2004
RSA cryptosystem with e=3, n: 1024 bits, NesC
- TinyECC, 1.9 sec, Liu et al. 2005
ECC, SECG curve over GF(q), q: 160-bit prime, NesC+Assembly
- TinyECCK, 1.1 sec, Chung et al. 2007
ECC, Koblitz curve over GF(2^{163}), NesC
- TinyTate, 30.2 sec, Oliveira et al. 2007
Tate pairing, Supersingular curves GF(q), q: 256-bit prime, NesC
- TinyPBC, 5.5 sec, Oliveira et al. 2007
 ηT pairing, supersingular curves GF(2^{271}), NesC
- Ours, 5.8 sec, Ishiguro et al. 2007
 ηT pairing, supersingular curves GF(3^{97}), NesC



Type of Pairing

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

- Symmetric Pairing

There is an efficient homomorphism $\phi: G_1 \rightarrow G_2$ (or $G_1 = G_2$).

- Asymmetric Pairing

There is no efficient homomorphism $\phi: G_1 \rightarrow G_2$ (and $G_1 \neq G_2$).

- Composite Order Pairing

$G_1 = G_2$ has a subgroup of composite order, e.g. RSA modulus.

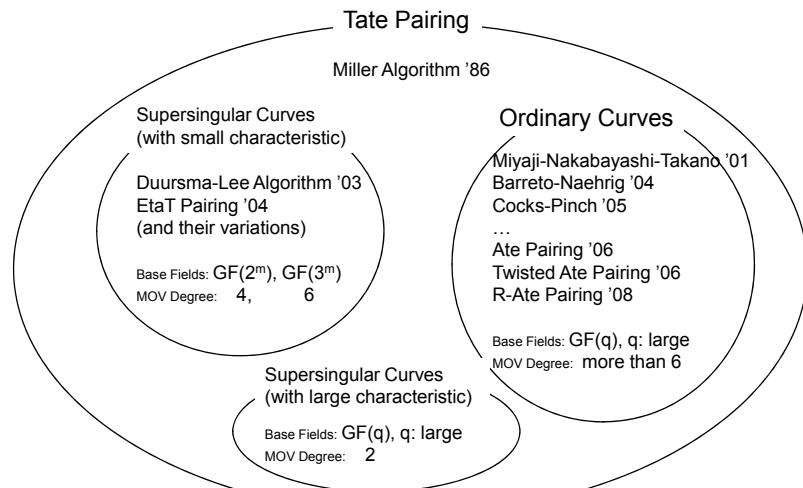
- Hyperelliptic Curve Pairing

pairing using hyperelliptic curves.

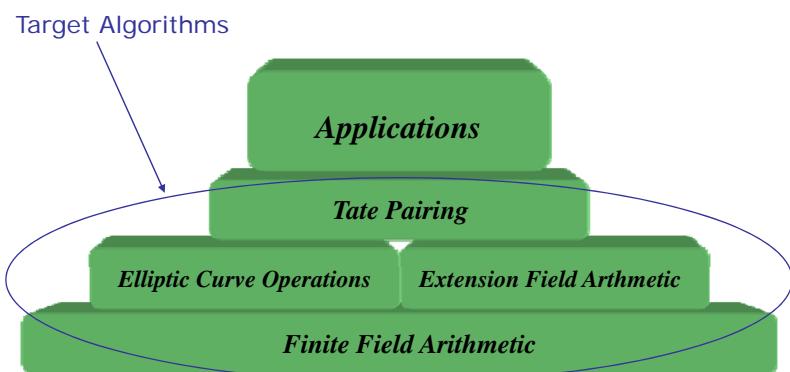




Several Pairings using Elliptic Curves



Implementation of PBC





η_T Pairing (2004)

- Input: $P = (x_p, y_p), Q = (x_q, y_q) \in E(GF(3^m))$
- Output: $\eta_T(P, Q) \in GF(3^{6m})$

```

 $y_p \leftarrow -y_p, f \leftarrow y_q\sigma - y_p + y_p(-x_q + \rho - x_p)$  Initialization
for  $i \leftarrow 0$  to  $(m-1)/2$  do
     $u \leftarrow x_p + x_q + 1$  Addition (+), Subtraction (-),
     $g \leftarrow y_p y_q\sigma - u^2 - u\rho - \rho^2$  Addition (+), Subtraction (-),
     $f \leftarrow f \cdot g$  Multiplication (in  $GF(3^m)$ )
     $x_p \leftarrow x_p^{1/3}, y_p \leftarrow y_p^{1/3}, x_q \leftarrow x_q^3, y_q \leftarrow y_q^3$  Main loop
end for
return  $f^{(3^{3m}-1)(3^m+1)(3^m-3^{(m+1)/2}+1)}$  Final exponentiation
  
```



Finite Fields $GF(3^m)$ (or F_{3^m})

- Arithmetic of $GF(3^m)$
 - Addition/Subtraction (A), Multiplication (M), Cubing (C), Inversion (I)
 - We can implement them using AND,OR,XOR.
- Extension Fields $GF(3^{6m})$
 - We can implement it using (A,M,C,I) of $GF(3^m)$
 - Elements $A = (a_5, a_4, a_3, a_2, a_1, a_0)$
 $= a_5\sigma\rho^2 + a_4\sigma\rho + a_3\sigma + a_2\rho^2 + a_1\rho + a_0$
 $(a_i \in GF(3^m), \rho^3 = \rho + 1, \sigma^2 = -1)$



Polynomial Base (m=97)

- Polynomial Base $GF(3^m) = GF(3)[x]/(x^{97}+x^{16}+2)$
 - $GF(3)$ is represented by (hi,lo)-bit.
 - $a = (a_{hi}, a_{lo})$, a in $GF(3) = \{0, 1, 2\}$.
 - $0 = (0,0)$, $1 = (0,1)$, $2 = (1,0)$

$\overbrace{\qquad\qquad\qquad}^{32 \text{ bit}} \quad A(x) = a_{96}x^{96} + a_{95}x^{95} + \cdots + a_1x + a_0$

a_{hi}	$0\ 0\ \cdots\ 0\ (a_{96})_{hi}$	$(a_{95})_{hi}\ \cdots$	\cdots	$\cdots\ (a_0)_{hi}$
a	$0\ 0\ \cdots\ 0\ a_{96}$	$a_{95}\ \cdots\ a_{64}$	$a_{63}\ \cdots\ a_{32}$	$a_{31}\ \cdots\ a_0$
a_{lo}	$0\ 0\ \cdots\ 0\ (a_{96})_{lo}$	$(a_{95})_{lo}\ \cdots$	\cdots	$\cdots\ (a_0)_{lo}$

$A[3] \qquad A[2] \qquad A[1] \qquad A[0]$

Addition

A(x), B(x) in GF(3⁹⁷)

$$\begin{aligned} C(x) &= A(x) + B(x) \\ &= (a_{97}+b_{97}) x^{97} + (a_{96}+b_{96}) x^{96} + \cdots + (a_0+b_0) \end{aligned}$$

Algorithm

- 1) $t = (a_{hi} \mid a_{lo}) \& (b_{hi} \mid b_{lo})$
 - 2) $c_{hi} = t \wedge (a_{hi} \mid b_{hi})$
 - 3) $c_{lo} = t \wedge (a_{lo} \mid b_{lo})$

Boolean Gates: AND(&), OR(|), XOR(^)



Multiplication

Input: $a(x) = a_{96}x^{96} + a_{95}x^{95} + \dots + a_0x^0 \in GF(3^{97})$

$b(x) = b_{96}x^{96} + b_{95}x^{95} + \dots + b_0x^0 \in GF(3^{97})$

$f(x) = x^{97} + x^{12} + 2$

Output: $c(x) = a(x) \times b(x) \bmod f(x)$

$$\begin{cases} a_i, b_i \in GF(3) \\ i = 0, 1, \dots, 96 \end{cases}$$

1: Multiplication of polynomials

$$c'(x) = a(x) \times b(x)$$

$$= a_{96}b_{96}x^{192} + (a_{95}b_{96} + a_{96}b_{95})x^{191} + \dots + a_0b_0x^0$$

2: Reduction

$$c(x) = c'(x) \bmod f(x)$$

13



June 4, 2008

Efficient and Secure Implementation of Pairing Based Cryptosystems



Shift-Add Method

Shift-Add Method

$$\begin{array}{r}
 a(x) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ a_{96}] \quad \cdots \quad [a_{15}a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8] \quad [a_7a_6a_5a_4a_3a_2a_1a_0] \\
 \times b(x) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ b_{96}] \quad \cdots \quad [b_{15}b_{14}b_{13}b_{12}b_{11}b_{10}b_9b_8] \quad [b_7b_6b_5b_4b_3b_2b_1b_0]
 \end{array}$$

$$\begin{array}{r}
 b_0 \times [0\ 0\ 0\ 0\ 0\ 0\ a_{96}] \quad \cdots \quad [a_{15}a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8] \quad [a_7a_6a_5a_4a_3a_2a_1a_0] \\
 b_1 \times [0\ 0\ 0\ 0\ 0\ 0\ a_{96}a_{95}] \quad \cdots \quad [a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8] \quad [a_6a_5a_4a_3a_2a_1a_0]
 \end{array}$$

⋮

$$\begin{array}{r}
 b_{96} \times \cdots [a_7a_6a_5a_4a_3a_2a_1a_0] \quad \cdots \quad [0\ 0\ 0\ 0\ 0\ 0\ 0] \quad [0\ 0\ 0\ 0\ 0\ 0\ 0]
 \end{array}$$

$$c(x) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ c_{192}] \quad \cdots \quad [c_{15}c_{14}c_{13}c_{12}c_{11}c_{10}c_9c_8] \quad [c_7c_6c_5c_4c_3c_2c_1c_0]$$

1 bit shift
1 bit shift

96 shift operations

14



June 4, 2008

Efficient and Secure Implementation of Pairing Based Cryptosystems

Comb Method

Comb Method

$$\begin{aligned}
 a(x) = & \quad [0\ 0\ 0\ 0\ 0\ 0\ 0\ a_{96}] \quad \cdots \quad [a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_0] \\
 \times b(x) = & \quad [0\ 0\ 0\ 0\ 0\ 0\ 0\ b_{96}] \quad \cdots \quad [b_1\ b_1\ b_1\ b_1\ b_1\ b_1\ b_1\ b_0] \\
 & \quad b_0 \times [0\ 0\ 0\ 0\ 0\ 0\ 0\ a_{96}] \quad \cdots \quad [a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_0] \\
 & \quad b_8 \times [0\ 0\ 0\ 0\ 0\ 0\ 0\ a_{96}] \quad \cdots \quad [a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_0] \\
 & \quad \quad \quad \vdots \\
 & \quad b_1 \times [0\ 0\ 0\ 0\ 0\ 0\ a_{96}] \quad \cdots \quad [a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_0] \\
 & \quad \quad \quad \vdots \\
 c(x) = & \quad [0\ 0\ 0\ 0\ 0\ 0\ 0\ c_{192}] \quad \cdots \quad [c_{15}\ c_{14}\ c_{13}\ c_{12}\ c_{11}\ c_{10}\ c_9\ c_8] \quad [c_7\ c_6\ c_5\ c_4\ c_3\ c_2\ c_1\ c_0]
 \end{aligned}$$

1 bit shift

June 4, 2008 7 Shift operations 2.1 times faster 15

Efficient and Secure Implementation of Pairing Based Cryptosystems

Cubing

$$\begin{aligned}
 \bullet \text{ Input: } & a(x) = a_{96}x^{96} + a_{95}x^{95} + \cdots + a_0x^0 \in GF(3^{97}) \\
 & f(x) = x^{97} + x^{16} + 2 \\
 \bullet \text{ Output: } & b(x) = a(x)^3 \mod f(x)
 \end{aligned}$$

$a_i \in \{0,1,2\}$
 $i = 0, 1, \dots, 96$

1: $b'(x) = a(x)^3$ 13

$$\begin{array}{ccccccc}
 & & & & & & \\
 & [0\ 0\ 0\ 0\ 0\ 0\ 0\ a_{96}] & \cdots & [a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_1\ a_0] & & & \\
 & [0\ 0\ 0\ 0\ 0\ 0\ a_{96}] & \cdots & [0\ a_7\ 0\ 0\ a_6\ 0\ 0] & [a_5\ 0\ 0\ a_4\ 0\ 0\ a_3\ 0] & [0\ a_2\ 0\ 0\ a_1\ 0\ 0\ a_0] & \\
 & & & \downarrow & & & \\
 & & & [0\ 0\ 0\ 0\ 0\ 0\ c_{96}] & \cdots & [c_{15}\ c_{14}\ c_{13}\ c_{12}\ c_{11}\ c_{10}\ c_9\ c_8] & [c_7\ c_6\ c_5\ c_4\ c_3\ c_2\ c_1\ c_0]
 \end{array}$$

2: $b(x) = b'(x) \mod f(x)$ 37

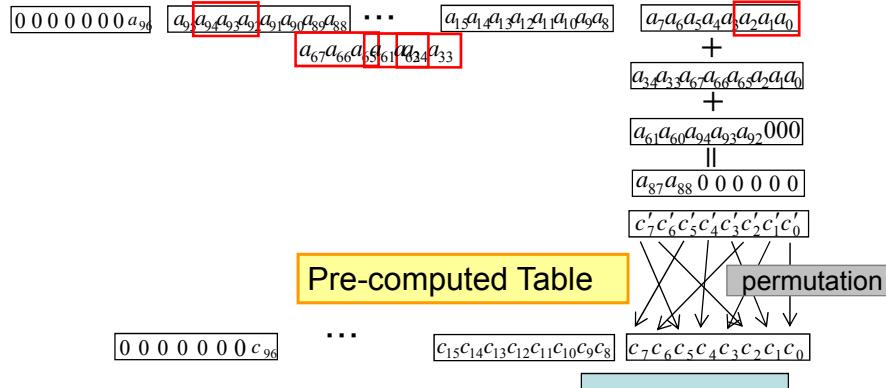
$$\begin{array}{ccccccc}
 & & & & & & \\
 & [0\ 0\ 0\ 0\ 0\ 0\ a_{96}] & \cdots & [0\ a_7\ 0\ 0\ a_6\ 0\ 0] & [a_5\ 0\ 0\ a_4\ 0\ 0\ a_3\ 0] & [0\ a_2\ 0\ 0\ a_1\ 0\ 0\ a_0] & \\
 & & & \downarrow & & & \\
 & & & [0\ 0\ 0\ 0\ 0\ 0\ c_{96}] & \cdots & [c_{15}\ c_{14}\ c_{13}\ c_{12}\ c_{11}\ c_{10}\ c_9\ c_8] & [c_7\ c_6\ c_5\ c_4\ c_3\ c_2\ c_1\ c_0]
 \end{array}$$

Efficient and Secure Implementation of Pairing Based Cryptosystems



Faster Cubing

Reduction trinomial $x^{97}+x^{16}+2$ allows us a simple pre-computation.



Timing

[IST07] Tsukasa Ishiguro, Masaaki Shirase, Tsuyoshi Takagi, "Efficient Implementation of the Pairing on ATmega128L", IPSJ Computer Security Symposium, CSS 2007, Nara, pp.187-192, October, 2007.

	MICAz (non-improved)	MICAz (improved)	PC (improved)
addition	0.047ms	0.047ms	0.12μs
multiplication	14ms	6.2ms	5.91μs
Cubing	2ms	0.4ms	0.32μs
Inversion	108ms	96ms	670.9μs
Pairing	≈ 30s	5.8s	5.41ms

MICAz, NesC
CPU: ATmega128L 7.37MHz
RAM: 4kB

PC, gcc
CPU: Core2Duo6300 1.87GHz
RAM: 1GB





Timing for larger degrees

	F_3^{97}	F_3^{167}	F_3^{193}	F_3^{239}
Addition	0.047	0.076	0.084	0.092
Cube	0.40	0.069	1.15	1.16
Multiplication	6.2	18.2	25.5	35.75
Inversion	96	1.6890	1.4480	2.3040
η_T Pairing (sec)	5.8	15.3	34.6	60.2

[milliseconds]

MICAz, NesC

CPU: ATmega128L 7.37MHz



Comparison

	TinyPK	TinyECC	TinyECCK	TinyTate	TinyPBC	Ours
Language	NesC	NesC,asm	NesC	NesC	NesC	NesC
Cryptosystem	RSA	ECC (char. p)	ECC (char. 2)	Tate (char.p)	η_T (char. 2)	η_T (char. 3)
ROM (bytes)	12,408	13,858	5,592	18,384	47,948	17,284
RAM (bytes)	1,167	1,440	1,002	1,831	368 (Stack 2,867)	628
Timing (sec)	14.5	1.9	0.9	30.2	5.5	5.8





Conclusion

- We implemented the η_T pairing of char. 3 on sensor node.
- The implementation is optimized for ATmega128L.
- The timing for GF(3⁹⁷) is about 5.8 seconds.

Future works

- Implementation by inline assembly
- Cryptographic applications using pairing

