Efficient Implementation of Pairing on Sensor Nodes

Tsukasa Ishiguro, Masaaki Shirase, *Tsuyoshi Takagi

Future University Hakodate, Japan

Sensor Node

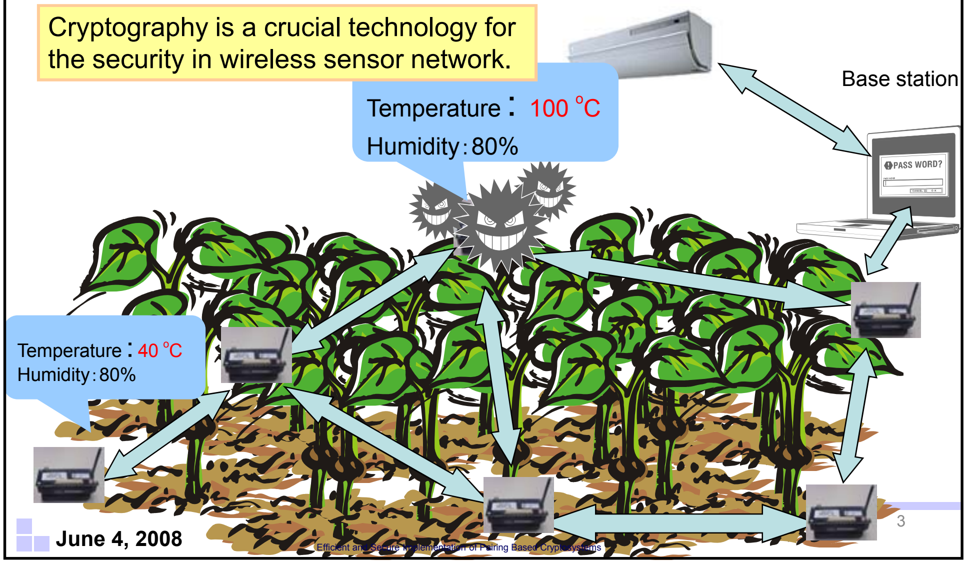
- MICAz
  - CPU: ATmega128L at 7.37MHz
  - ROM: 128kB
  - SRAM: 4kB
  - 8-bit CPU
  - Size: 62x35x27 (mm)

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

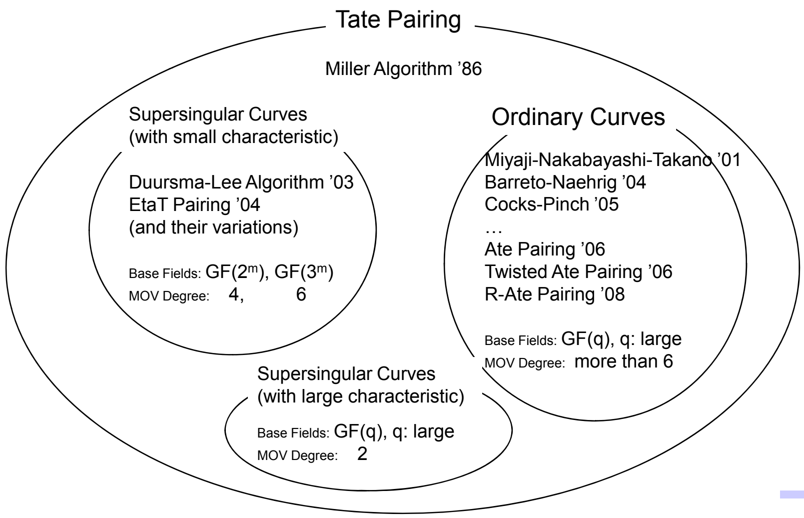
- •TinyPK, 14.5 sec, Watro et al. 2004
  RSA cryptosystem with e=3, n: 1024 bits, NesC
- •TinyECC, 1.9 sec, Liu et al. 2005
  ECC, SECG curve over GF(q), q: 160-bit prime, NesC+Assembly
- •TinyECCK, 1.1 sec, Chung et al. 2007
  ECC, Koblitz curve over $GF(2^{163})$, Nesc
- •TinyTate, 30.2 sec, Oliveira 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

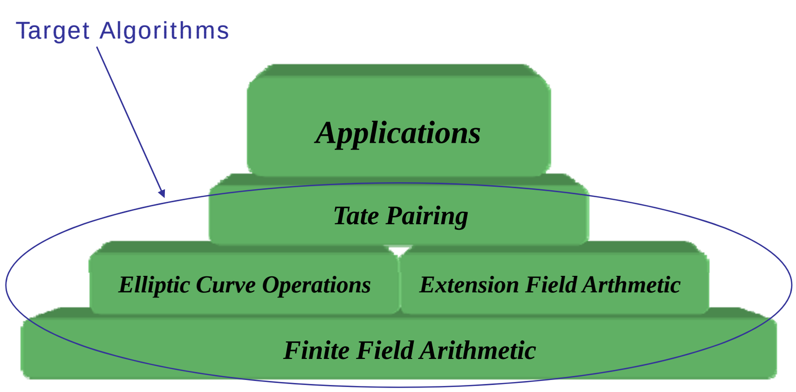
Pairing e: $G_1 \times G_2 \rightarrow G_T$

- Symmetric Pairing
  There is an efficient homomorphism $\varphi: G_1 \rightarrow G_2$ (or $G_1 = G_2$).
- Asymmetric Pairing
  There is no efficient homomorphism $\varphi: G_1 \rightarrow G_2$ (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



Future University-Hakodate

η_T Pairing (2004)

- Input: P = (x_p, y_p), Q = (x_q, y_q) ∈ *E(GF(3^m))*
- Output: η_T(P, Q) ∈ *GF(3^{6m})*

Initialization

$y_p \leftarrow -y_p,\quad f \leftarrow y_q\sigma - y_p + y_p(-x_q + \rho - x_p)$

Main loop

for $i \leftarrow 0$ to $(m-1)/2$ do

$\quad u \leftarrow x_p + x_q + 1$

$\quad g \leftarrow y_p\, y_q\sigma - u^2 - u\rho - \rho^2$

$\quad f \leftarrow f \cdot g$

$\quad x_p \leftarrow x_p^{1/3},\ y_p \leftarrow y_p^{1/3},\quad x_q \leftarrow x_q^3,\ y_q \leftarrow y_q^3$

end for

Final exponentiation

return $f^{(3^{3m}-1)(3^m+1)(3^m-3^{(m+1)/2}+1)}$

Addition (+), Subtraction (-),

Future University-Hakodate

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)

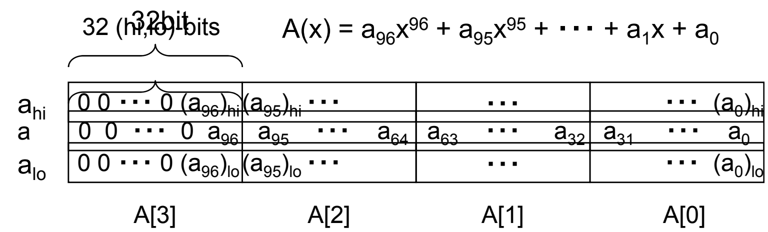
$$= a_5\sigma\rho^2 + a_4\sigma\rho + a_3\sigma + a_2\rho^2 + a_1\rho + a_0$$

$$(a_i \in GF(3^m),\ \rho^3 = \rho + 1,\ \sigma^2 = -1)$$

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)

$A(x) = a_{96}x^{96} + a_{95}x^{95} + \cdots + a_1 x + a_0$



Addition

$A(x), B(x)$ in $GF(3^{97})$

$C(x) = A(x) + B(x)$

$= (a_{97}+b_{97})\, x^{97} + (a_{96}+b_{96})\, x^{96} + \cdots + (a_0+b_0)$

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

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

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

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

$$\left(\begin{array}{l} a_i, b_i \in GF(3) \\ i = 0,1,\cdots,96 \end{array} \right)$$

1: Multiplication of polynomials

$$c'(x) = a(x) \times b(x)$$
$$= a_{96}b_{96}x^{192} + (a_{95}b_{96} + a_{96}b_{95})x^{191} + \cdots + a_0b_0x^0$$

2: Reduction

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

Future University-Hakodate

Shift-Add Method

Shift-Add Method

$a(x) =$ [0 0 0 0 0 0 0 a_{96}] ⋯ [$a_{15}a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8$] [$a_7a_6a_5a_4a_3a_2a_1a_0$]

× $b(x) =$ [0 0 0 0 0 0 0 b_{96}] ⋯ [$b_{15}b_{14}b_{13}b_{12}b_{11}b_{10}b_9b_8$] [$b_7b_6b_5b_4b_3b_2b_1b_0$]

b_0 × [0 0 0 0 0 0 0 a_{96}] ⋯ [$a_{15}a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8$] [$a_7a_6a_5a_4a_3a_2a_1a_0$]

1 bit shift

b_1 × [0 0 0 0 0 0 $a_{96}a_{95}$] ⋯ [$a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8a_7$] [$a_6a_5a_4a_3a_2a_1a_0$ 0]

1 bit shift

⋮

b_{96}× ⋯ [$a_7a_6a_5a_4a_3a_2a_1a_0$] ⋯ [0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0]

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

96 shift operations

Comb Method

Comb Method

7 Shift operations

2.1 times faster

Cubing

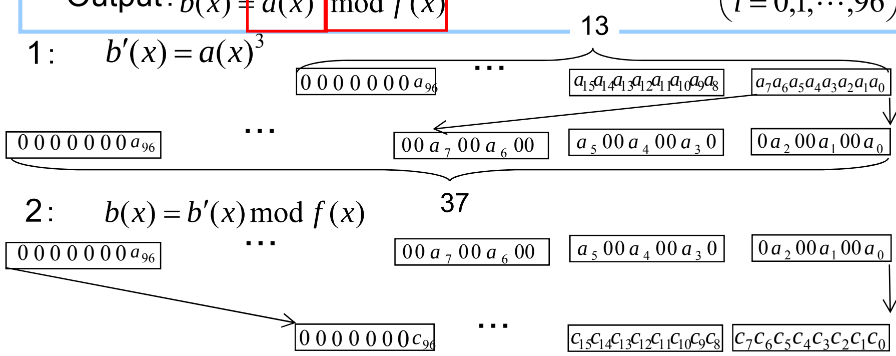
- Input: $a(x) = a_{96}x^{96} + a_{95}x^{95} + \cdots + a_0x^0 \in GF(3^{97})$

  $f(x) = x^{97} + x^{16} + 2$

  $\left(\begin{array}{l} a_i \in \{0,1,2\} \\ i = 0,1,\cdots,96 \end{array} \right)$
- Output: $b(x) = a(x)^3 \bmod f(x)$

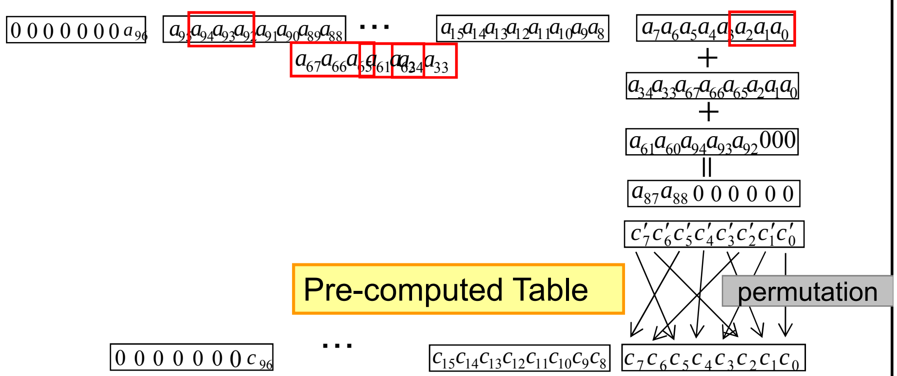
1: $b'(x) = a(x)^3$

2: $b(x) = b'(x) \bmod f(x)$



Faster Cubing

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



5.0 times faster

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

Future University-Hakodate

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 $GF(3^{97})$ is about 5.8 seconds.

Future works

- Implementation by inline assembly
- Cryptographic applications using pairing