Computing Cryptographic Pairing in Sensors

Zhibin Zhou
Arizona State University
zhibin.zhou@asu.edu

Dijiang Huang
Arizona State University
dijiang@asu.edu

Abstract

In this paper, we present a methodology and preliminary result of computing the Tate Pairing on a supersingular curve over a prime field in the wireless sensors (MICAz Mote). The aim of this work is to study the feasibility of pairing based protocols and applications in sensors with limited computational resources. Tate pairing is the most computationally intensive computation in most pairing-based cryptographic algorithms. Our preliminary results showed that without hardware upgrades (esp. memory) or further optimization of algorithms and parameters, sensors not yet ready for computing pairing-based cryptographic algorithms.

1 Introduction

Pairing-based cryptography is based on elliptic curve operations. The central idea of pairing is the construction of a mapping between two finite groups. It allows for new cryptographic schemes using the transformation from one group to the other group. Much research has showed that pairing has great potential in wireless sensor networks. However, the really evaluation of pairing in sensors have not been studied in literature. In this paper, we show that the computation of pairing in wireless sensors (MICAz) is conditionally feasible. We have the following contributions:

1) We implement the Tate pairing algorithm using Nesc in 8-bit processor sensor, i.e., MICAz Mote.

2) We adopt majority of available optimizations for pairing operations and make several improvements of the pairing algorithms over a supersingular curve.

3) Based on our research results, we propose some future software based improvements for pairing implementations in sensors.

2 Related Work

Most of pairing algorithms are based on Miller Algorithm [8]. In [1, 3], the authors described in details of various solutions to implement Tate pairing in the characteristic 2 and 3 for large prime numbers. In the [6], the authors reported an implementation of Tate pairing in MICAz sensor which used 256-bit prime field with 128-bit sub-group size. In [9] and [2], the authors demonstrated the implementations and results of Tate pairing in lightweight devices such as PDA and smartcard, respectively.

3 Pairing Preliminaries and Testing Platform

Let n be a natural number coprime to q. The Tate pairing of order n is the bilinear map:

$$e_n(P,Q) : E(\mathbb{F}_q)[n] \times E(\mathbb{F}_q^k)[n] \rightarrow F_{q^k}^*. \qquad (1)$$

It has the following properties:

1) **Bilinearity**: $e_n(P_1+P_2,Q) = e_n(P_1,Q)\cdot e_n(P_2,Q)$ and $e_n(P,Q_1+Q_2) = e_n(P,Q_1)\cdot e_n(P,Q_2)$ for all $P,P_1,P_2 \in E(\mathbb{F}_q)[n]$ and all $Q,Q_1,Q_2 \in E(F_{q^k}[n])$.

2) **Nondegeneracy**: If $e_n(P,Q) = 1$ for all the $Q \in E(\mathbb{F}_{q^k}[n])$, then $P = \mathcal{O}$. Alternatively, for each $P \neq Q$ there exists $Q \in E(\mathbb{F}_{q^k}[n])$ such that $e_n(P,Q) \neq 1$.

MICAz sensor running TinyOS was used for implementing Tate pairing. The pairing implementation was developed in TinyOS operating system using NESC programming language. TinyOS is an open source embedded system operating system targeted for wireless sensor network. It is written in Nesc language as a set of cooperating tasks and processes. The hardware configurations are given in following Table:

Model	**MICAz**
Micro-controller	ATMega128L (8 bit)
Clock Speed	7.37 MHZ
RAM	4 KB
Program Memory	128 KB
Flash	512 KB
Radio Frequency	2400 MHZ

We have to make sure the size of memory required by the program variables does not exceed 4 KB. Since MICAz has only a 8 bit processor, the computation of pairing takes a long time because all primitive operations of pairing are *Big* integer operations and they have to be implemented using Multi-precision algorithms.

4 Results and Observations

In this section, we present the dome preliminary results for computing Tate pairing in MICAz sensor and some observations through our implementation.

4.1 Implementation Results

Our preliminary implementation is based on a modified Miller Algorithm to compute Tate pairing on a MICAz Mote. The MICAz Mote runs TinyOS using NESC programming language. The primitive operations of pairing, i.e., field operations on $\mathbb{F}_p$ were implemented using the software package RSAREF [5] provided by RSA Laboratories. In addition, we implemented our own field operations on $\mathbb{F}_{p^2}$ and we modified the point addition function provided by TinyECC [7] to calculate the slope function which is one of the time consuming operations of pairing. The following parameters were used in our implementation. The cost

Curve	Super Singular curve : $y^2 = x^3 + x$
Embedded Degree	k : 2
Sub-group size	l : 160 bit Solinas prime
Prime	p : 256 bit

of running one Tate pairing operation in MICAz is given in the following Table. The cost is measured in terms of computational time and memory usage. The memory usage is further divided to indicate the amount of the program memory and data memory (RAM) used by the program.

Tate Pairing on MICAz	**Cost**	**Memory consumption percentage**
Time	31.25 seconds	-
Data memory (RAM)	974 bytes	24%
Program memory (Flash)	16686 bytes	13%

4.2 Observations

The following are some of the observations we made through the pairing implementation.

1) Dynamic allocation of memory posed some problems while computing pairing in MICAz though it is efficient in terms of memory usage than static allocation. This is because pairing by itself is time consuming operation for sensor and dynamic memory allocation would further increase the time.

2) We also observed that calculating slope of point addition is faster using affine coordinates system compared to using projective coordinates system.

3) We tried to compute Tate pairing using 512 bit prime number p instead of 256-bit p to achieve higher security. We observed that 512-bit field operations are very expensive in terms of memory for MICAz. With the given hardware, pairing could not run to completion.

5 Future Improvements

To further improve the efficiency, we can adopt the following strategies:

1) *Adopt precomputation in the pairing implementation.*

2) *Implement more efficient field multiplications for sensors.* In the paper [4], the author proposed a multiplication algorithm which combines column-wise and row-wise multiplication to reduce the number of memory access during the computation. We can actually implement and test the algorithm in the MICAz Mote.

3) *Adopt memory efficient system parameters.* As a matter of fact, there is a tradeoff between the memory usage and the computation speed. If we use the finite field with the characteristic 2 or 3, the embedded degree k increase to 4 or 6, which means the pairing is mapped to $\mathbb{F}_{2^m}$ or $\mathbb{F}_{3^m}$, respectively. However, to reach the same security level as $\mathbb{F}_p$ where p is a 512-bit prime, each group element can be 256 bits or 160 bits, compared to 512-bit in the $\mathbb{F}_p$.

4) *Implement pairing in Imote sensors* Imote sensor are equipped with more powerful hardwares and thus is more suitable for test platform.

6 Conclusion

From our preliminary implementation of pairing in MICAz, we showed that computing secure Tate pairing in MICAz is quite expensive. We have also identified various optimizations to improve the computational time and memory usage of pairing in sensors. Therefore without hardware upgrades especially memory or further optimizations it may be infeasible to implement 512 bit pairing in Micaz. Our future work will be implement more optimized tate pairing in Micaz and Imote sensors.

References

[1] P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. *Advances in Cryptology–Crypto*, 2442:354–368, 2002.

[2] G. Bertoni, L. Chen, P. Fragneto, K. Harrison, and G. Pelosi. Computing Tate Pairing on Smartcards. *White Paper STMicroelectronics*, 2005.

[3] S. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate pairing. *Algorithmic Number Theory 5th International Symposium, ANTS-V*, 2369:324–337.

[4] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. *Proceedings of Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004)*, 2004.

[5] R. Laboratories. Rsaref: A cryptographic toolkit, version 2.0. 1994.

[6] E. M. F. D. J. L. Leonardo B. Oliveira, Diego Aranha and R. Dahab. TinyTate: Identity-Based Encryption for Sensor Networks.

[7] A. Liu and P. Ning. Tinyecc: Elliptic curve cryptography for sensor networks.

[8] V. Miller. Short programs for functions on curves. *Unpublished manuscript*, 1986.

[9] A. Ramachandran, Z. Zhou, and D. Huang. Computing Cryptographic Algorithms in Portable and Embedded Devices. *to appear at IEEE PORTABLE 2007*.