New Parallel Architecture for Modular Multiplication and Squaring Based on Cellular Automata

Kyo-Min Ku[1], Kyeoung-Ju Ha[2], Hyun-Sung Kim[3], and Kee-Young Yoo[4]

[1] Daegu National University of Education,
1797-6, DaeMyung-Dong, Nam-Gu, Daegu, Korea, 705-715
kmku@dnue.ac.kr
[2] KyungSan University, 75 San, JumChon-Dong,
Kyungsan, KyungPook, Korea, 712-715
kjha@kyungsan.ac.kr
[3] KyungIl University, Hayangup, Kyungsansi, KyungPook, Korea
kim@kiu.ac.kr
[4] KyungPook National University,
1370 Sankyuk-Dong, Puk-Gu, Daegu, Korea 702-701
yook@knu.ac.kr

Abstract. Modular exponentiation in a finite field is the basic computation involved in most public key crypto systems, such as Diffie-Hellman key exchange, ElGamal, etc. The current paper presents a new parallel architecture whereby the modular multiplication and squaring can be processed simultaneously in GF(2^m) in m clock cycles using a cellular automata. Since the proposed cellular automata architecture is simple, regular, modular, cascadable, it can also be utilized efficiently for the implementation of VLSI.

1 Introduction

With the recent rapid expansion of the internet it is now possible to readily obtain various forms of information and information services. However, unfortunately, potentially dangerous and destructive malfunctions also accompany such convenience and profitability. Accordingly, this has increased the need for information protection, resulting in the development of many types of security technologies and an increased public interest in crypto systems. For the past 30 years, studies on finite fields have been conducted in many areas, including crypto systems [1], and most public key crypto systems, such as Diffie-Hellman key exchange and ElGamal, are based on modular exponentiation computations in a finite field [2], [3]. Such modular exponentiation uses a modular multiplier as the basic structure for its implementation. The Elliptic Curve Cryptosystem is also based on constant multiplication [4]. Examples of the algorithms used to implement multipliers include the LSB-first multiplication algorithm [5], MSB-first multiplication algorithm [6], and Montgomery algorithm [7]. Previous research and development on modular multiplication is as follows: First, for a

J. Fagerholm et al. (Eds.): PARA 2002, LNCS 2367, pp. 359–369, 2002.

one-dimensional systolic array, in the case of an LSB-first algorithm, the modular multiplication is performed within $3m$ clock cycles using m cells [5]. While in the case of an MSB-first algorithm, the modular multiplication can be performed within $3m$ clock cycles using m cells [6]. With an LFSR structure, the modular multiplication can be performed within $2m$ clock cycles using m cells [10], the modular multiplication can be performed within m clock cycles using m cells and the modular squaring can be performed within m clock cycles using m cells [11]. The structures proposed in [5], [6], [10], [12] are simple modular multipliers. However, when computing exponentiation, such structures must be repeated twice for modular multiplication and squaring. In case of [11], the structures of multiplication and squaring must be used together to simultaneously perform the modular multiplication and squaring.

The purpose of the current paper is to reduce the time and the space, and to investigate and develop a simple, regular, modular, and cascadable architecture for the VLSI implementation of exponentiation in GF(2^m) based on cellular automata, which is the basic computation in any public key crypto system. A cellular automata, particularly a 3-neighbor additive cellular automata, can satisfy such requirements very well and has already been applied in many areas, such as the encryption, decryption, etc. of a crypto-system [8], [9].

Accordingly, this paper proposes a new parallel architecture in which a 3-neighbor cellular automata is used to simultaneously process modular multiplication and squaring for effective exponentiation in GF(2^m). The proposed architecture can simultaneously perform multiplication and squaring in m clock cycles using m cells, $3m$ AND gates, $3m-1$ XOR gates, and $5m$ registers. Based on the properties of LSB-first multiplication, the parts of modular multiplication and squaring that can be performed in common are identified, then the remainder is processed in parallel. As a result, exponentiation can be performed much more efficiently as regards time and space compared to repeating the structure as proposed in [5], [6], [10] and can be performed much more efficiently as regards space compared to repeating the structure as proposed in [11] [12].

The remainder of the paper is as follows: Chapter 2 gives an overview of the concept of cellular automata, while Chapter 3 reviews the structure of exponentiation in GF(2^m). Chapter 4 introduces the structure of the proposed multiplier/squarer for efficient exponentiation using a cellular automata. Chapter 5 gives an analysis. Finally, Chapter 6 offers some conclusions.

2 Cellular Automata(CA)

Cellular automata consist of numbers of interconnected cells arranged spatially in a regular manner [8], [9]. The next state of a cell depends on the present states of 'k' of its neighbors, for a k-neighborhood CA. Example of one rule of a 2-state 3-neighbor 1-dimensional CA is shown below.

State of neighbor:	111	110	101	100	011	010	001	000	
Next state:	0	1	0	1	1	0	1	0	(Rule 90)