Computers & Security (2005) 24, 83–86

ELSEVIER

Computers & Security

www.elsevier.com/locate/cose

Multiplexer-based double-exponentiation for normal basis of GF(2^m)

Che Wun Chiou[a,*], Chiou-Yng Lee[b,1]

[a]*Department of Electronic Engineering, Ching Yun University, 229, Chien-Hsin Road, Chung-Li 320, Taiwan, ROC*
[b]*Program Coordination Department, Telecommunication Laboratories, Yang-Mei 326, Taiwan, ROC*

Received 16 July 2004; revised 16 July 2004; accepted 9 September 2004
Available online 28 January 2005

KEYWORDS
Cryptography;
Finite (Galois) field arithmetic;
Multiplier;
Normal basis;
Exponentiation

Abstract In many cryptographic protocols, double-exponentiation is a key arithmetic operation. In this study, we will present a multiplexer-based algorithm for double-exponentiation in GF(2^m). The proposed algorithm utilizes the concept of the modified Booth's algorithm. Multiplexers are employed for implementation of the proposed algorithm. The proposed double-exponentiation algorithm only requires m multiplications and saves about 66% time complexity while comparing with the ordinary binary method.

Introduction

Modular exponentiation is the main operation in public-key cryptosystems such as the RSA (Rivest et al., 1978) and Diffie and Hellman (1976). In recent years, many cryptographic protocols (Brickell and McCurley, 1991; NIST, 1991; Schnorr, 1990) use double-exponentiation computation. Moreover, double-exponentiation is also useful in the area of encoding the Reed and Solomon (1960) codes. The modular exponentiation is a difficult task to carry out efficiently, especially when performed in software for large fields. Therefore, it is desirable to have high-speed exponentiation algorithms.

Recently, finite (Galois) fields have received attention due to their important and practical applications in areas of communications such as error-correcting codes (MacWilliams and Slone, 1977), digital signal processing (Blahut, 1985), cryptography (Lidl and Niederreiter, 1994), and encoding of the Reed–Solomon codes (Berlekamp, 1982). One popular representation of elements in GF(2^m) is a normal basis (Massey and Omura, 1986; Wang et al., 1985; Reyhani-Masoleh and

* Corresponding author. Tel.: +886 3 4581196x5104; fax: +886 3 4588924.
E-mail addresses: cwchiou@cyu.edu.tw (C.W. Chiou), lchiou@m.ieice.org (C.-Y. Lee).
[1] Tel.: +886 3 4245215; fax: +886 3 4244168.

0167-4048/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2004.09.012

Hasan, 2002; Sunar and KoÇ, 2001; Takagi et al., 2001). The major advantage of the normal basis is that the squaring of an element is computed by a cyclic shift of the binary representation. Therefore, the normal basis is very effective for performing inverse, squaring, and exponentiation operations. In this study, a new double-exponentiation algorithm using normal basis and the modified Booth's algorithm (MacSorley, 1961) is presented. The proposed double-exponentiation algorithm only requires m multiplications while the conventional binary method (Knuth, 1981) takes $3m+1$ multiplications in average. Our proposed algorithm saves about 66% time complexity.

Preliminaries

Lidl and Niederreiter (1994) have showed that a normal basis always existed in GF(2^m) for all positive integers m. For an $\alpha \in$ GF(2^m), the set $\{\alpha^{2^0}, \alpha^{2^1}, \alpha^{2^2}, \ldots, \alpha^{2^{m-1}}\}$ is termed the normal basis N if $\alpha^{2^0}, \alpha^{2^1}, \alpha^{2^2}, \ldots,$ and $\alpha^{2^{m-1}}$ are linearly independent. Using a normal basis, any elements A and B in GF(2^m) can be represented as vectors $(a_0, a_1, a_2, \ldots, a_{m-1})$ and $(b_0, b_1, b_2, \ldots, b_{m-1})$, respectively, hold the following properties:

(a) $(A+B)^2 = A^2 + B^2$,
(b) $A^{2^{m-1}} = 1$ and $A^{2^m} = A$,
(c) $A^2 = a_{m-1}\alpha^{2^0} + a_0\alpha^{2^1} + a_1\alpha^{2^2} + \cdots + a_{m-2}\alpha^{2^{m-1}}$.

The normal basis representation of $A \in$ GF(2^m) can be represented as

$$A = \sum_{i=0}^{m-1} a_i \alpha^{2^i} \tag{1}$$

Let $T_i(X)$ perform the following function:

$$T_i(X) = X^{2^i} = x_{m-i}\alpha^{2^0} + x_{m-i+1}\alpha^{2^1} + x_{m-i+2}\alpha^{2^2} + \cdots + x_{m-1}\alpha^{2^{i-1}} + x_0\alpha^{2^i} + x_1\alpha^{2^{i+1}} + \cdots + x_{m-i+1}\alpha^{2^{m-1}},$$

where X is an element in GF (2^m and $X = x_0\alpha^{2^0} + x_1\alpha^{2^1} + \cdots + x_{m-1}\alpha^{2^{m-1}}$.

The circuit for $T_i(X)$ is easily realized just by rewiring the bit positions. Therefore, the cost for $T_i(A)$ is negligible.

The multiplexer-based double-exponentiation algorithm

Let A and $B \in$ GF(2^m) be represented in normal basis, and let K and H be m-bit positive integers in binary representation. The double-exponentiation of the form $A^K B^H$ is computed by

$$\begin{aligned} &A^K B^H \\ &= A^{k_0 + k_1 2 + k_2 2^2 + \ldots + k_{m-1}2^{m-1}} B^{h_0 + h_1 2 + h_2 2^2 + \cdots + h_{m-1}2^{m-1}} \\ &= (A^{k_0} A^{k_1 2} A^{k_2 2^2} \cdots A^{k_{m-1}2^{m-1}})(B^{h_0} B^{h_1 2} B^{h_2 2^2} \cdots B^{h_{m-1}2^{m-1}}) \\ &= (A^{k_0} B^{h_0})(A^{k_1 2} B^{h_1 2})(A^{k_2 2^2} B^{h_2 2^2}) \cdots (A^{k_{m-1}2^{m-1}} B^{h_{m-1}2^{m-1}}) \\ &= (A^{k_0} B^{h_0})(A^{k_1} B^{h_1})^2 (A^{k_2} B^{h_2})^{2^2} \cdots (A^{k_{m-1}} B^{h_{m-1}})^{2^{m-1}}. \end{aligned}$$

The function $A^{k_i} B^{h_i}$ can be performed by using the modified Booth's algorithm (MacSorley, 1961). The major advantage of the modified Booth's algorithm is that two bits are processed in parallel. For enjoying the advantage of the Booth's algorithm, a 4-to-1 multiplexer (the truth table is listed in Table 1) is used to realize the function $A^{k_i} B^{h_i}$, and the circuit is shown in Fig. 1. The function block for realizing $A^{k_i} B^{h_i}$ using multiplexers is depicted in Fig. 2.

Table 1 Truth table for 4-to-1 multiplexer

Inputs		Output
k_i	h_i	y
0	0	1
0	1	B
1	0	A
1	1	AB

The exponentiation can be performed by repeated multiplications, such as the binary method (Knuth, 1981). Wang and Pei (1990) computed exponentiation A^K with normal basis representation for A and binary representation for K. Table 2 shows the comparison of our proposed double-exponentiation algorithm with other algorithms using the traditional binary method and the Wang–Pei algorithm. As compared to the ordinary binary method, our proposed method saves about 66% time complexity. Our multiplexer-based double-exponentiation algorithm also saves about 50% time complexity while comparing with the Wang–Pei algorithm.

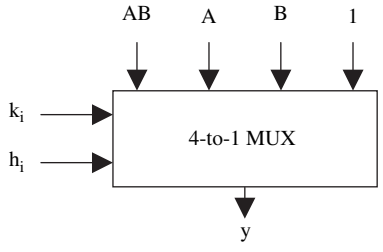

Figure 1 A 4-to-1 multiplexer for realizing $A^{k_i} B^{h_i}$.

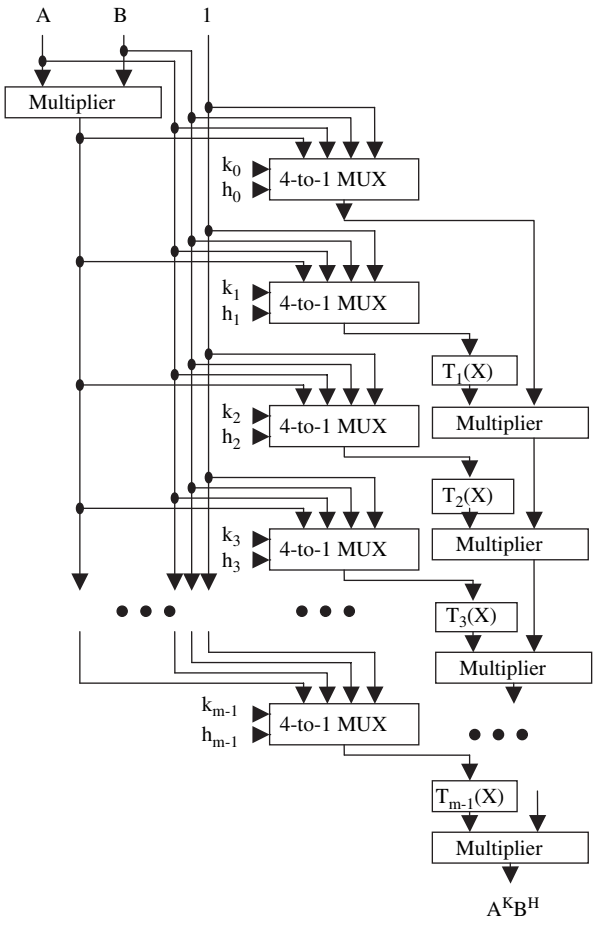

Figure 2 The proposed multiplexer-based double-exponentiation.

Conclusions

We have presented a multiplexer-based algorithm for double-exponentiation in GF(2^m). The proposed double-exponentiation algorithm employs the concept of the modified Booth's algorithm and thus the advantage of the Booth's algorithm is preserved. Only m multiplications are required in our proposed double-exponentiation algorithm while $3m + 1$ and $2m + 1$ multiplications are required for the ordinary binary method and the Wang—Pei algorithm, respectively.

Table 2 Comparison of various double-exponentiation algorithms

Algorithms	Basis	No. of multiplications
Binary method (Knuth, 1981)	Binary representation	$3m + 1$
Using Wang and Pei (1990)	Normal basis	$2m + 1$
Fig. 2	Normal basis	m

References

Berlekamp ER. Bit-serial Reed—Solomon encoder. IEEE Trans Inf Theory Nov. 1982;IT-28:869—74.

Blahut RE. Fast algorithms for digital signal processing. Reading, Mass.: Addison-Wesley; 1985.

Brickell EF, McCurley KS. Interactive identification and digital signatures. AT&T Tech J 1991;73—86.

Diffie W, Hellman ME. New directions in cryptography. IEEE Trans Inf Theory 1976;IT-22(6):644—54.

Knuth DE. The art of computer programming. In: Seminumerical algorithms, vol. 2. Addison-Wesley; 1981.

Lidl R, Niederreiter H. Introduction to finite fields and their applications. New York: Cambridge University Press; 1994.

MacSorley L. High speed arithmetic in binary computers. Proc IRE Jan 1961;49.

MacWilliams FJ, Slone NJA. The theory of error-correcting codes. Amsterdam: North-Holland; 1977.

Massey JL, Omura JK. Computational method and apparatus for finite field arithmetic. U.S. Patent number 4,587,627; May 1986.

NIST. A proposed federal information processing standard for digital signature standard (DSS). Fed Reg 1991;56: 42980—2.

Reed IS, Solmon G. Polynomial codes over certain finite fields. SIAM J Appl Math 1960;8:300—4.

Reyhani-Masoleh A, Hasan MA. A new construction of Massey—Omura parallel multiplier over GF(2^m). IEEE Trans Comput May 2002;51(5):511—20.

Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 1978;21(2):120—6.

Schnorr CP. Efficient identification and signature for smart cards. In: Advances in cryptology, Crypto'89, Lecture notes in computer science, vol. 435. Springer-Verlag; 1990: 239—52.

Sunar B, KoÇ Ç.K. An efficient optimal normal basis type II multiplier. IEEE Trans Comput Jan. 2001;50(1):83—7.

Takagi N, Yoshiki J-I, Takagi K. A fast algorithm for multiplicative inversion in GF(2^m) using normal basis. IEEE Trans Comput May 2001;50(5):394—8.

Wang CC, Pei D. A VLSI design for computing exponentiations in GF(2^m) and its application to generate pseudorandom number sequences. IEEE Trans Comput Feb. 1990;39(2): 258—62.

Wang CC, Truong TK, Shao HM, Deutsch LJ, Omura JK, Reed IS. VLSI architectures for computing multiplications and inverses in GF(2^m). IEEE Trans Comput Aug. 1985;C-34(8): 709—17.

Che Wun Chiou received his B.S. degree in Electronic Engineering from Chung Yuan Christian University, Taiwan, in 1982, his M.S. degree and Ph.D. degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 1984 and 1989, respectively. From 1990 to 2000, he was with the Chung Shan Institute of Science and Technology in Taiwan. He joined the Department of Electronic Engineering, Ching Yun University, Taiwan, in 2000. He is currently the Dean of Division of Continuing Education in Ching Yun University. His current research interests include fault-tolerant computing, computer arithmetic, parallel processing, and cryptography.

Chiou-Yng Lee received his Bachelor's degree (1986) in medical engineering and M.S. degree in Electronic Engineering (1992), both from the Chung Yuan university, Taiwan, and the Ph.D. degree in Electrical Engineering from Chang Gung University, Taiwan, in 2001. From 1988 till date, he is a research associate with Chunghwa Telecommunication Laboratory in Taiwan. He joined the department of project planning. He taught those related field courses at Ching Yun University. His research interests include computations in finite fields, error-control coding, signal processing, and digital transmission system. Besides, he is a member of the IEEE and the IEEE Computer society. He is also an honorary member of Phi Tao Phi in 2001.