Compact Multiplicative Inverter for Hardware Elliptic Curve Cryptosystem

M. Wong, Ka Man

Compact Multiplicative Inverter for Hardware Elliptic Curve Cryptosystem

M. M. Wong*[1], M. L. D. Wong[1, 2], and K. L. Man[2]

[1] School of Engineering, Computing and Science,
Swinburne University of Technology Sarawak Campus, Malaysia.
mwong@swinburne.edu.my
[2] Xi'an Jiaotong-Liverpool University, Suzhou, China.
Dennis.Wong@xjtlu.edu.cn, ka.man@xjtlu.edu.cn

Abstract. This paper presents a compact design of a multiplicative inverter for elliptic curve cryptosystems. Using a methodology based on the composite field arithmetic, we propose a combinatorial solution to mitigate the usage of look up tables as commonly adopted by the conventional software based approach. In particular, we perform further isomorphism in the subfield, such that the required arithmetic are constructed using logical AND and XOR gates only. In this work, we demonstrate our proposed methodology with the field $GF((2^8)^{41}) \cong GF((((2^2)^2)^2)^{41})$ in optimal normal type II basis. The chosen field is both secure and results in efficient computation. An analysis of the resultant hardware complexity of our inverter is reported towards the end.

Keywords: Elliptic curve (EC) cryptosystems, composite field arithmetic (CFA), Itoh and Tsujii inversion algorithm (ITIA), multiplicative inversion.

1 Introduction

Finite fields play an essential role in the modern cryptographic applications. As such, the complexity of its underlying field's arithmetic will determine the amount of resources required in the final cryptosystem. Therefore, the first, and the most essential step in constructing a compact and efficient elliptic curve (EC) hardware cryptosystem is to choose the suitable field for ECC computation. Therefore, composite field, which offers greater computational efficiency compared to other finite fields, is a favourable choice. The prior studies in composite field EC cryptosystems had emphasized on software implementations where look-up tables (LUTs) were utilized in the subfield arithmetic [1-4]. Consequently, the unbreakable delays of LUTs will determine the maximum attainable clock rate of the final hardware circuitry. This drawback can be avoided by employing combinatorial approaches, i.e. using only the combinatorial logic for the hardware construction.

* The work of M. M. Wong was supported by Swinburne University of Technology Sarawak Campus under a Ph.D. studentship

In particular, the scalar multiplication, kP, is the most crucial and yet the most complicated operation in any elliptic curve cryptography (ECC) [5, 6] applications. It involves a repetition of point additions and point doublings, which requires inversions over the finite field when defined in affine coordinate system [7]. Therefore, in this work, we propose a compact and efficient inversion circuit through the exploitation of composite field arithmetic (CFA) for EC hardware cryptosystem. Two main criteria are taken into consideration during the construction, which are the **security** aspect and the **complexity** of the underlying arithmetic. In short, we need to select an optimal field that is insusceptible to the known attacks and also results in combinatorial inversion circuitry without the need of LUTs.

2 Composite Field Inversion for Elliptic Curve Cryptography

Construction of the composite field inverter in EC cryptosystem requires three major steps. The first and also the most important step is choosing an appropriate field that would circumvent the cryptographic attacks on the elliptic curve discrete logarithm problem (ECDLP) [8-9]. ECDLP is defined as follows. Given an elliptic curve E, defined over a finite field $GF(q)$, a point $P \in E(GF(q))$ of order r, and a second point $Q \in \langle P \rangle$, determine the integer $l \in [0, r-1]$ such that $Q = lP$. The ECDLP is of particular interest because its apparent intractability would form the basis for the security of EC cryptographic schemes [10].

In 2000, Gaudry, Hess and Smart (GHS) [11] showed that the Weil descent attack methodology (see [12]), can be used to reduce any instance of the ECDLP to an instance of discrete logarithm problem (DLP) in the Jacobian of a hyperelliptic curve over $GF(2^N)$. Only for the case where $N \in [160{,}600]$ is prime, $GF(2^N)$ is secure from the GHS attack [13]. In other words, the use of elliptic curves over $GF(2^N)$ with N is a composite number is not recommended.

In the later date, the applicability of the GHS attack on the ECDLP for elliptic curves over $GF(2^N)$ for composite $N \in [160{,}600]$ was further analyzed by Maurer et al. in [10]. The elliptic curves of composite field $GF((2^n)^m)$ that are susceptible to the GHS attack were identified and listed precisely in their paper. Therefore, this allows us to select the composite field that is not weak under GHS attack.

For security purposes, the extension field, m, has to be a considerably large prime number, while the subfields, n, is chosen to be relatively smaller in order to simplify the computation. Hence, we have chosen $GF((2^8)^{41})$ for our design.

In the second step, after the field selection, we consider algorithmic optimization to achieve area reduction in the inverter design. While the previous studies focused on two-level isomorphism composite field, we propose to perform further isomorphisms in the subfield $GF(2^8)$, such that it is further reduced to $GF(((2^2)^2)^2)$. With this, we can derive a combinatorial inverter circuitry without the use of LUTs. Furthermore, normal basis representation is often a preferred choice over the polynomial basis representation in hardware implementation. Among the normal bases, the optimal normal basis (ONB) manages to further reduce the complexity of the complicated normal

basis multipliers. As we have decided the extension field, m, to be a prime number, ONB type II representation is sought here.

Last, we employ the Itoh and Tsujii inversion (ITI) algorithm [14, 15] to perform the efficient and compact multiplicative inversion over the selected composite field. The ITI algorithm presented below as Theorem 1 is a Fermat's Little Theorem (FLT)-based inversion algorithm which can efficiently reduce the inversion in the extension field $GF((2^n)^m)$ to the inversion in its subfield, $GF(2^n)$.

Theorem 1 (Itoh & Tsujii Inversion [14]). *Let* $A \in GF((2^n)^m)$, $A \neq 0$ *and* $r = \frac{(n^m-1)}{(n-1)}$. *The inverse of an element* A *can be computed as* $A^{-1} = (A^r)^{-1}.A^{r-1}$, *with* $A \in GF(2^n)$.

Overall, in this work, we derive a combinatorial inverter over $GF((((2^2)^2)^2)^{41})$ for EC cryptosystems in ONBII representation using the ITI algorithm. Detailed description of our proposed inverter will be presented in the next section. To the best of our knowledge, this is the first reported work on using ITI for the aforementioned configuration.

3 Design and Implementation

Our composite field inverter using ITI algorithm can be accomplished through the following four steps. Here after, we denote our field as $GF(q^m)$ with $q = (((2^2)^2)^2)$ and $m = 41$.

Step 1: Exponentiation of $A^{r-1} \in GF(q^m)$. The exponent $r-1$ can be expressed as a sum of powers $q^{40} + q^{39} + q^{38} + \cdots + q^2 + q$. Through a series of repeated power raising and multiplication, the exponentiation is accomplished as follows;

$$
\begin{aligned}
A^{q^2} &= (A^q)^q \\
A^q.A^{q^2} &= A^{q^2+q} \\
\left(A^{q^2+q}\right)^{q^2}.\left(A^{q^2+q}\right) &= A^{q^4+q^3+q^2+q} = A^{\Sigma_{i=1}^{4} q^i} \\
\left(A^{\Sigma_{i=1}^{4} q^i}\right)^{q^4}.\left(A^{\Sigma_{i=1}^{4} q^i}\right) &= A^{q^8+q^7+\cdots+q} = A^{\Sigma_{i=1}^{8} q^i} \\
\left(A^{\Sigma_{i=1}^{8} q^i}\right)^{q^8}.\left(A^{\Sigma_{i=1}^{8} q^i}\right) &= A^{q^{16}+q^{15}+\cdots+q} = A^{\Sigma_{i=1}^{16} q^i} \\
\left(A^{\Sigma_{i=1}^{16} q^i}\right)^{q^{16}}.\left(A^{\Sigma_{i=1}^{16} q^i}\right) &= A^{q^{32}+q^{31}+\cdots+q} = A^{\Sigma_{i=1}^{32} q^i} \\
A^{r-1} &= \left(A^{q^{32}+q^{31}+\cdots+q}\right)^{q^8}.A^{q^8+q^7+\cdots+q}
\end{aligned}
\tag{1}
$$

The complexity to compute A^{r-1} using addition chain (see (1)) is found to be 6 multiplications in $GF(q^{41})$ and 40 exponentiations to the q^{th} power. While the exponentiation requires only q cyclic shifts, the $GF(q^{41})$ multiplier needs to be implemented using a normal basis multiplier.

Step 2: Multiplication of A and A^{r-1} that yield $A^r \in GF(q)$. In the second step, multiplication of two operands $A, A^{r-1} \in GF(q^{41})$ will result in $A^r \in GF(2^q)$. Subsequently, we need a specific multiplier that compute the first coefficient in the general multiplication in $F(q^{41})$. This step can be accomplished with 81 multiplications and 81 additions over $GF(2^q)$. Note that in the finite field of characteristic 2, both subtraction and addition are implemented using a XOR operation.

Step 3: Inversion in $GF(2^n)$ yields $(A^r)^{-1}$. Instead of using LUTs, we utilize a combinatorial circuitry to perform the inversion over the composite field $GF(((2^2)^2)^2)$. The inversion involves three level of isomorphisms which requires three field polynomials stated (in a general form) below:

$$r(y) = y^2 + y + \nu, \text{ extension of } GF(2^8)/GF(2^4) \quad (2)$$
$$s(z) = z^2 + Tz + 1, \text{ extension of } GF(2^4)/GF(2^2) \quad (3)$$
$$t(w) = w^2 + w + 1, \text{ extension of } GF(2^2)/GF(2) \quad (4)$$

The inverter architecture is described with reference to their respective field polynomials in general. First, for the isomorphism between $GF(2^8)/GF(2^4)$, we have the element of field $GF(2^8)$, δ, expressed as $\gamma_1 Y^{16} + \gamma_0 Y$, where $\gamma_0, \gamma_1, \nu \in GF(2^4)$ and using both roots of $r(y)$ as $r(y) = (y + Y)(y + Y^{16})$. Second, for the isomorphism between $GF(2^4)/GF(2^2)$, we have the element of field $GF(2^4)$, Δ, expressed as $\Gamma_1 Z^4 + \Gamma_0 Z$, where $\Gamma_1, \Gamma_0, T \in GF(2^2)$ and $s(z) = (z + Z)(z + Z^4)$. Last, for the isomorphism between $GF(2^2)/GF(2)$, we let element of field $GF(2^2)$, d, expressed as $g_1 W^2 + g_0 W$, where $g_0, g_1 \in GF(2)$ and $t(w) = (w + W)(w + W^2)$.

Hence, the multiplicative inverse of $\gamma_1 Y^{16} + \gamma_0 Y$ can be computed as stated in (5),

$$(\gamma_1 Y^{16} + \gamma_0 Y)^{-1} = [\gamma_1 \Theta] Y^{16} + [\gamma_0 \Theta] Y \quad (5)$$

where $\Theta = [\gamma_0 \gamma_1 + (\gamma_0^2 + \gamma_1^2)\upsilon]^{-1}$. The arithmetic in (5) can be decomposed into several subfield operations, namely the multiplications and the inversions. To summarize, the arithmetic required over the inversion is tabulated in Table 1 and as depicted in Figure 1. The total complexity of our inverter is 36 ANDs and 96 XORs.

Step 4: Multiplication of $(A^r)^{-1}.A^{r-1}$. In this final step, we need to multiply $A^{r-1} \in GF(q^m)$ (from Step 1) and $(A^r)^{-1} \in GF(\mathrm{q})$ (from Step 3) to deduce A^{-1} .This step requires $m = 41$ multiplications in $GF(\mathrm{q})$. Let $g, h \in GF(((2^2)^2)^2)$ be $\{\gamma_1 Z^8 + \gamma_0 Z\}$ and $\{\delta_1 Z^8 + \delta_0 Z\}$ respectively. Multiplication of g and h is then derived in (6),

$$\begin{aligned}&(\gamma_1 Z^8 + \gamma_0 Z)(\delta_1 Z^8 + \delta_0 Z)\\ &= (\gamma_1 \delta_1)(Z^8)^2 + (\gamma_1 \delta_0 + \gamma_0 \delta_1) Z^8 Z + (\gamma_0 \delta_{01}) Z^2\\ &= [(\gamma_1 + \gamma_0)(\delta_1 + \delta_0)\nu + \gamma_1 \delta_1] Z^8 + [(\gamma_1 + \gamma_0)(\delta_1 + \delta_0)\nu + \gamma_0 \delta_0] Z \end{aligned} \quad (6)$$

with a complexity of 27 ANDs and 81 XORs.

Table 1. Multiplicative Inverse for $GF(((2^2)^2)^2)$

Operation	Equation
Inversion in $GF(2^8)$	$\delta_1 = [\gamma_1\gamma_0 + (\gamma_1^2 + \gamma_0^2)\nu]^{-1}\gamma_0$ $\delta_0 = [\gamma_1\gamma_0 + (\gamma_1^2 + \gamma_0^2)\nu]^{-1}\gamma_1$
	$\nu\gamma^2 = [(\Gamma_0 + \Gamma_1)^2]Z^4 + N^2\Gamma_0^2$
Inversion in $GF(2^4)$	$\Delta_1 = [\Gamma_1\Gamma_0 T^2 + (\Gamma_1^2 + \Gamma_0^2)]^{-1}\Gamma_0$ $\Delta_0 = [\Gamma_1\Gamma_0 T^2 + (\Gamma_1^2 + \Gamma_0^2)]^{-1}\Gamma_1$
	$\Gamma^2 = g_0W^2 + g_1W$ $\Gamma T = (g_0 + g_1)W^2 + g_1W$
Inversion in $GF(2^2)$	$d_1 = g_0$ $d_0 = g_1$
Multiplication in $GF(2^4)$	$(\Gamma_1\Delta_1)(Z^4)^2 + (\Gamma_1\Delta_0 + \Gamma_0\Delta_1)Z^4Z + (\Gamma_0\Delta_0)Z^2$
Multiplication in $GF(2^2)$	$[(g_1 + g_0)(d_1 + d_0) + g_1d_1]W^2$ $+[(g_1 + g_0)(d_1 + d_0) + g_0d_0]W$

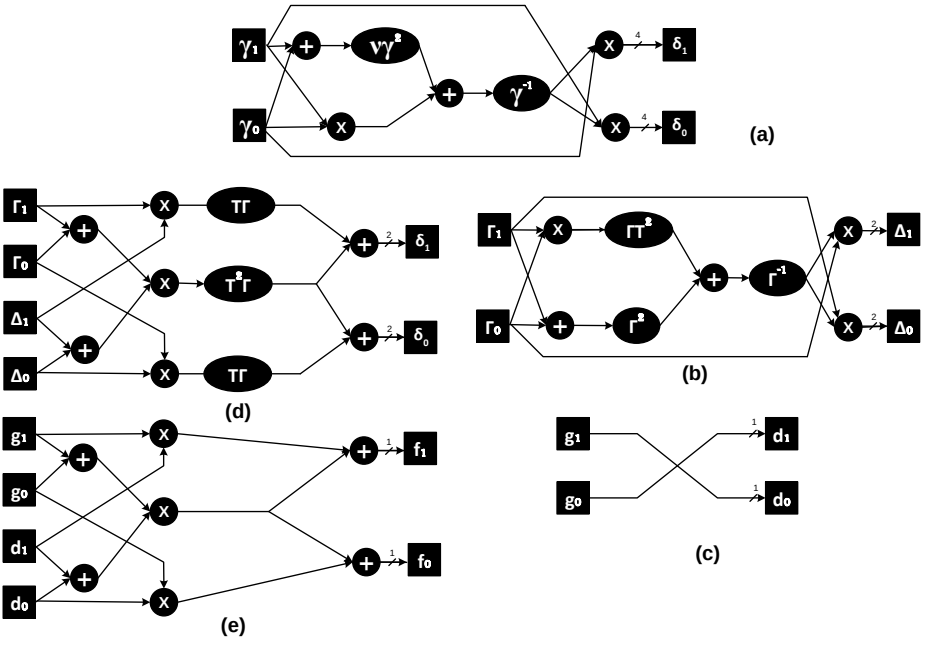

Fig. 1. Inversion over $GF(2^8)$ using CFA. (a) Inversion in $GF(2^8)$, (b) Inversion in $GF(2^4)$, (c) Inversion in $GF(2^2)$, (d) Multiplication in $GF(2^4)$, (e) Multiplication in $GF(2^2)$

4 Discussion and Results

To demonstrate the efficacies of our inverter in EC hardware cryptosystem, its computational cost is benchmarked with the previous works. To our best knowledge, the most recent and comparable work from the literature was presented by Guajardo and Paar in 1997 [4]. They employed ITI algorithm for inversion over two levels composite field of $GF((2^n)^m)$ in polynomial basis representation, and the subfield $GF(2^n)$ arithmetic was computed using the LUT approach.

The LUT approach employed in the previous works of the composite field EC cryptosystems [1-4] was performed using log and $antilog$ conversion. In this approach, three and two tables of 2,048 bits were used to calculate the multiplication and the inversion of the field elements respectively. Meanwhile, without using any LUT, our $GF(((2^2)^2)^2)$ inverter and multiplier are constructed using 36 ANDs and 96 XORs, and 27 ANDs and 81 XORs respectively.

Due to the large amount of subfield multiplier are required, the complexity of the subfield multiplier determines the hardware cost (area and power) and the performance of the inverter architecture. Here, we point out the advantages of using combinatorial $GF(((2^2)^2)^2)$ multiplier as opposed to the LUT approach in hardware implementation. Having both architectures implemented in Cyclone III EP3C120F780I7 FPGA, the summary of the hardware requirements are tabulated in Table 2. Based on the result in Table 2 our combinatorial circuitry is capable of promoting a significant saving in term of hardware resources and with higher overall performance compared to the conventional LUT approach, which is based on log and $antilog$ conversion method.

Table 2. Hardware analysis of FPGA implementation for $GF(((2^2)^2)^2)$ multiplier using (i) combinatorial circuity as proposed in our work and (ii) log and antilog conversions.

	(i) Combinatorial Circuity	(ii) Log and Antilog Conversions
Total LE	51	432
Total Combinatorial Functions	51	432
Dedicated Logic Register	0	0
Total Register	8	0
Total Memory Bits	0	2,048
Fmax (MHz)	142.76	95.15
Total Thermal Power Dissipation (mW)	79.83	80.61
Core Dynamic Thermal Power Dissipation (mW)	2.90	3.65
Core Static Power Dissipation (mW)	68.26	68.27
I/O Thermal Power Dissipation (mW)	8.67	8.70

Furthermore, we also include the existing EEA-based inverter architectures over binary field, $GF(2^m)$ [16-18] for benchmarking. The complexity of these architectures working in $GF(2^{328})$, together with the work by Guajardo and Paar in

$GF((2^8)^{41})$ and our work are summarized in Table 3. The analytical results in Table 3 proved that composite field results in compact architecture design compared to the binary field. Therefore, composite field that is insusceptible towards cryptographic attacks is highly desirable in hardware EC cryptosystem implementation.

Table 3. Analytical comparison of various inverter architectures.

	Guajardo and Paar [14]	Guo and Wang [16]	Wu at al. [17]	Yan and Sarwate [18]	Our Work
Finite Field	$GF((2^8)^{41})$	$GF(2^{328})$	$GF(2^{328})$	$GF(2^{328})$	$GF((((2^2)^2)^2)^{41})$
OR gates	0	0	1,312	0	0
NOT gates	0	0	656	0	0
AND gates	0	645,504	654,504	430,336	275,652
XOR gates	315,688	430,336	215,168	430,336	986,340
XOR3 gates	0	215,168	215,168	0	0
Adder	7,812	656	0	0	0
Mux	0	860,672	654,504	645,504	0
LUT (2048 bits)	8,279	0	0	0	0

5 Conclusion

This work presented of a secure and compact combinatorial inverter for EC cryptosystems over $GF((((2^2)^2)^2)^{41})$ in ONBII representation. Unlike the previous works, we performed further isomorphisms in the subfield, $GF(2^8) \cong GF(((2^2)^2)^2)$, such that the need for LUTs can be eliminated completely. Using the ONBII representation, we chose the extension field m, to be a prime number while allowing the 40 exponentiations be implemented easily using simple cyclic shifts. In addition to that, we have shown the advantages of using combinatorial circuitry for EC hardware cryptosystem as opposed to the LUT approach. Furthermore, we has proven our composite field inverter is more compact than those binary field inverters which were reported in the literature.

6 Reference

1. Harper, G., Menezes, A., Vanstone, S.: Public-key cryptosystems with very small key lengths. In: Proceedings of the 11th annual international conference on Theory and application of cryptographic techniques. EUROCRYPT'92, Berlin, Heidelberg, Springer-Verlag (1993) 163-173
2. Beauregard, D.: Efficient algorithms for implementing elliptic curve public-key schemes. In: Master's thesis, ECE Dept., Worcester Polytechnic Institute. (1996)
3. Win, E.D., Bosselaers, A., Vandenberghe, S., Gersem, P.D., Vandewalle, J.: A fast software implementation for arithmetic operations in GF(2^n). In: Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology, London, UK, Springer-Verlag (1996) 65-76

4. Guajardo, J., Paar, C.: Efficient algorithms for elliptic curve cryptosystems. In Kaliski, B., ed.: Advances in Cryptology CRYPTO '97. Volume 1294 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (1997) 342-356 10.1007/BFb0052247.
5. Koblitz, N.: Constructing elliptic curve cryptosystems in characteristic 2. In Menezes, A., Vanstone, S., eds.: Advances in Cryptology CRYPT0 '90. Volume 537 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (1991) 156-167 10.1007/3-540-38424-3-11.
6. Miller, V.S.: Use of elliptic curves in cryptography. In: Lecture notes in computer sciences; 218 on Advances in cryptology-CRYPTO 85, New York, NY, USA, Springer-Verlag New York, Inc. (1986) 417-426
7. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer-Verlag, New York, Inc (2004)
8. Menezes, A., Teske, E., Weng, A.: Weak fields for ECC. In: CT-RSA'04. (2004) 366-386
9. Menezes, A., Teske, E.: Cryptographic implications of Hess' generalized GHS attack. Applicable Algebra in Engineering, Communication and Computing 16 (2006) 439-460 10.1007/s00200-005-0186-8.
10. Maurer, M., Menezes, A., Teske, E.: Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. In: Proceedings of the Second International Conference on Cryptology in India: Progress in Cryptology. INDOCRYPT '01, London, UK, UK, Springer-Verlag (2001) 195-213
11. Gaudry, P., Hess, F., Smart, N.: Constructive and destructive facets of Weil descent on elliptic curves. Journal of Cryptology 15 (2002) 19-46 10.1007/s00145-001-0011-x.
12. Frey, G.: Applications of arithmetical geometry to cryptographic constructions. In: Proceedings of the Fifth International Conference on Finite Fields and Applications, Springer 128-161
13. Menezes, A., Qu, M.: Analysis of the Weil descent attack of Gaudry, Hess and Smart. In: Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA. CT-RSA 2001, London, UK, UK, Springer-Verlag (2001) 308-318
14. Itoh, T., Tsujii, S.: A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. Inf. Comput. 78 (September 1988) 171-177
15. Guajardo, J., Paar, C.: Itoh-Tsujii inversion in standard basis and its application in cryptography and codes. Designs, Codes and Cryptography 25 (2002) 207-216 10.1023/A:1013860532636
16. Guo, J.H., Wang, C.L.: Hardware-efficient systolic architecture for inversion and division in GF(2^m). Computers and Digital Techniques, IEE Proceedings - 145(4) (jul 1998) 272 - 278
17. Wu, C.H., Wu, C.M., Shieh, M.D., Hwang, Y.T.: Systolic VLSI realization of a novel iterative division algorithm over GF(2^m): a high-speed, low-complexity design. In: Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on. Volume 4. (may 2001) 33-36 vol. 4
18. Yan, Z., Sarwate, D.: New systolic architectures for inversion and division in GF(2^m). Computers, IEEE Transactions on 52(11) (nov. 2003) 1514 -1519