ISSN 2047-3338

FPGA Based Low Footprint Security Chip for Portable Embedded Systems

Sunil Devidas Bobade[1] and Vijay R. Mankar[2]
[1]S.G.B. Amravati University Amravati, India
[2]R.B.T.E. Pune Region, Pune, India

***Abstract*— Elliptic Curve Cryptography is one of the most preferred public key Cryptography algorithms known for its security strength and reduced key size. This makes ECC most suitable for implementing the hardware security chip for providing security services in portable embedded devices. This paper presents an area efficient Security Chip housing ECC algorithm for data encryption over binary field and hybrid multiplier for performing cryptographic multiplications. In proposed FPGA security chip, revised and simplified double point multiplication algorithm is adopted and to achieve further reduction, new low footprint hybrid multiplier is integrated. This hybrid multiplier integrates traditional Karatsuba Multiplier with systolic multiplier. The proposed hybrid multiplier does the initial recursion using the systolic algorithm while final small sized multiplications are accomplished using the Karatsuba algorithm The complete Security Chip housing ECC encryption engine and multiplier module is synthesized and simulated using Xilinx 14.4 software and is implemented on Xilinx Virtex-4 xc4vlx200ff1513 FPGA. Experimental results show proposed Security Chip utilizes very little FPGA resources, when comparing with other such architectures.**

***Index Terms*— ECC, Double Point Multiplication, Karatsuba Multiplier and Systolic Multiplier**

I. INTRODUCTION

SYSTEM security is an increasingly important design criterion for many embedded systems. These systems are often portable and more vulnerable and are easily attacked than traditional desktop and server computing systems. Cryptography is the process of making the transfer and storage of data secure, to avoid the eaves drops. For the rapid growth of electronics in nowadays, it needs more secure data transmission. Confidential data exchange over public computer network needs authentication [1]. Low-cost embedded devices like RFID tags and smart cards are rapidly becoming pervasive in daily life. Well known embedded applications include electronic passports, contactless payments, product tracking, access control and supply chain management just to name a few. But the small programmable chips that passively respond to every reader have raised concerns among researchers about privacy and security breaches. A considerable body of research has been focused on providing embedded systems with cryptographic functionality, while scarce computational and storage capabilities of low cost embedded devices make the problem challenging.

Digital signatures and secure channels meant for valid data exchange and cryptography gives a solution for this. The cryptography is divided into mainly two, Public key cryptography and symmetric key cryptography. RSA is the well established public key cryptography. In public key cryptography, the senders and receivers key is different. For symmetric cryptography, both the keys involved are same [2]. Cryptography process consists of the encryption and decryption of data, the encryption key and decryption key.

Elliptic curve cryptography (ECC) is superior to RSA. Elliptic curve cryptosystems possess a number of degrees of freedom like Galois field characteristic, extension degree, elliptic curve parameters, or the fixed point generating the working subgroup on the curve. The beauty of this new field is potentially related to the simplicity of the operators used in the encryption process, to the non secure transmission constraints used in the exchange of the keys and to the enhanced complexity that might face hackers when unwanted information goes out of the organization [3].

ECC employs a relatively short encryption key. It is faster and requires less computing power than other first-generation encryption public key algorithms such as RSA, Diffie-Hellman. For example, a 160-bit ECC encryption key provides the same security as a 1024-bit RSA encryption key and can be up to 15 times faster, depending on the platform on which it is implemented. ECC is extremely helpful for use on low memory and low computing environments of embedded systems such as mobile devices, wireless devices etc. Elliptic curve cryptography has proven to be a promising solution for the implementation of public-key cryptosystems in embedded systems. As widespread use of the internet and mobile devices continues to increase, transferring information with less computation and in a more secure manner has been the primary focus. With smaller key sizes and lower processing requirements, elliptic curve cryptography serves the purpose on embedded devices.

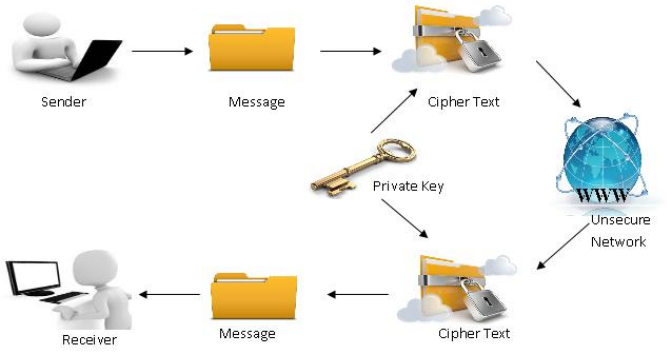

Fig. 1: Encryption and Decryption Process

II. Research Methodology

Many hardware designs of elliptic curve cryptography have been developed, aiming to accelerate the scalar multiplication processes, mainly based on the platform of field programmable gate arrays (FPGAs) [4]. The application field of FPGAs has clearly outgrown the prototype-only use. More and more FPGA implementations are now being implemented in an environment which used to be ASIC-only territory. When these applications are implemented on an FPGA, they need secure data communication [5]. In this rapidly changing environment, the reconfigurability of an FPGA is a very useful feature which is not available on an ASIC. In the last decade, the approach of hardware implementing elliptic curve cryptography (ECC) algorithm knew a very concentrated contest, due essentially to the requirements of security, speed, and area constraints[6]-[8].

The application of elliptic curves to the field of cryptography has been relatively recent. Hardware realization of cryptography tasks costs time, money, and energy [4]. The core theme of work is based on implementing an elliptic curve cryptography processor utilizing different techniques which can be used to enhance its performance [5], [6]. Such a processor can influence applications in different ways:

By increasing the speed, it enables more people to use the system in the same time and increases the availability. It can reduce the overall system costs. If energy consumption is minimized, the processor can decrease the total energy, and for example increase the battery lifetime in cell phones.

Most of the hardware implementation of cryptographic engine that can perform computations with high throughputs lacks hardware efficiency and are hence not suitable for portable devices [7]. Since hardware structure decides the market value for embedded devices, a reduced chip size is mostly preferred. Keeping in mind the memories, controllers and the various other peripheral circuits that are essential in building a chip, it is advantageous to design the crypto processor fitting in a small area. Even though the software implementation of the cryptographic algorithms using the RISC processors reports high throughput and low power and area consumption, the users and the manufactures are still concerned about the susceptibility of its software implementation to side channel attacks. Hence it is mandatory to design the crypto processor using the hardware circuits that meet the prescribed requirement and can withstand to these attacks. Hence with all the restraints discussed above, it is very challenging to design and implement a cryptographic algorithm with area, power and performance efficient.

The novelty of proposed work is to design and build FPGA based security chip meant exclusively for protecting contents of portable embedded systems with minimum number of slices and LUTs as possible. The chip houses ECC encryption engine based on revised and simplified version of double point multiplication algorithm proposed in [9]. For further lowering the foot print of the complete chip design, the multiplier used in work is replaced with an area efficient hybrid multiplier, which is the main contribution of work. This reduced footprint chip offers confidentiality to the data being stored in embedded storage.

The rest of the paper is organized as follows. Section 3 discusses some previous approaches for elliptic curve hardware realization. In Section 4, the proposed Security chip comprising of ECC engine and hybrid multiplier is presented. Section 5 evaluates the results and comparisons for all the designs synthesized in Xilinx. The work is concluded in Section 6.

III. Related Work

Several architectures of ECC processor are available in the literature. Few speed optimized and few are area optimized. Vladimir Tujillo-Olaya and Jaime Velasco-Medina built area efficient ECC processor by performing scalar point multiplication activity using Lopez- Dahab algorithm [10]. The processor employed serial multiplier with a bit size of 30. The design by Chiou-Yng Lee and Pramod Kumar Meher in cryptoprocessor [11] adapted a new divide and conquer technique for computing TMVPIP. This novel approach drastically reduced the delay time to (1+3log2 log2 m) Tx+ Ta and made it highly speed efficient.

Jen-Wei Lee et al in [12] designed an Elliptic Curve Cryptographic by adapting Priority-oriented scheduling of right-to-left double and add-always EC scalar multiplication with randomized processing technique and dual-PE design. The cryptographic processor provided protection against power analysis attacks. An Energy-adaptive dual-field processor was implemented by Jyu-Yuan Lai and Chih-Tsun Huang in [13]. In implementation, an advanced field inversion method and the scheduler- controlled data path are integrated adapting the TSMC 130-nm CMOS cell based technology. The implemented design exhibited an increase in throughput with reduce area, power, and energy consumption.

In [14], Hossein Mahdizadeh and Massoud Masoumi presented a Novel architecture for efficient FPGA implementation of elliptic curve cryptographic processor over GF(2^{163}).In implementation, the execution delay of the LD algorithm was reduced by parallelization of the multipliers. The hardware implementation of the elliptic curve cryptographic processor was designed with the objective of area reduction. The final output reported slice occupancy of 17929, 33414 LUTs, and an efficiency of 508. Ashkan Hosseinzadeh Namin et al in [15] implemented a word-level finite field multiplier using normal basis with smaller critical

path delay, circuit complexities, and high speed. The implemented design shows a critical path delay of 6.698ns with 15 clock cycles. The circuit complexity of the design is low when compared with other methods that exist.

A Full-custom VLSI design of a unified multiplier for elliptic curve cryptography on RFID tag was implemented by Johann Grobschadl in [16]. The implemented design shows a power saving and energy-delay with area constraints as added advantage. The implemented design shows a Delay (poly) of 66.8 nSec for radix-2, 38.0 nSec for radix-4 and an average current of 3.66 mA for radix-2 , 3.49 mA for radix-4.

IV. Proposed Low Footprint Security Chip Architecture

A. Elliptical Curve Cryptography Process

ECC algorithm uses two points $P_1(x_1, y_1), P_2(x_2, y_2)$ picked up from the elliptic curve E and a two randomly selected integers k_1, k_2 acting as private key for the sender and receiver from a set of $1-(n-1) \in Z$. The public key Q is computed by performing double point multiplication operation over P_1, P_2 and k_1, k_2 (see algorithm 1).

$$\text{Algorithm}: 1$$
$$Input \quad : P_1, P_2 \in E$$
$$k_1, k_2 \in \{1-(n-1)\}$$
$$Output : Q \in E$$
$$1. Choose\, k_1 \; and \; k_2$$
$$2. Compute\, Q = k_1 P_1 + k_2 P_2$$
$$3. return\, Q$$

Using this public key, encryption of message is carried out. The sender first selects a random number $d \in \{1-(n-1)$ and the message M to be encrypted. The cipher text C_1 is obtained by adding the two points P_1, P_2 and multiplying the added point with d. The next cipher text C_2 is computed by adding the message M with point by adding the two points $k_1 P_1$ and $k_2 P$, that are computed in the key generation stage and multiplying it with the randomly selected integer d. The cipher texts thus generated are communicated through unsecure network. (See algorithm 2).

$$\text{Algorithm}: 2$$
$$Input \quad : P_1, P_2, M, k_1 P_1, k_2 P_2 \in E$$
$$d \in \{1-(n-1)\}$$
$$Output : C_1, C_2 \in E$$
$$1. Choose\, d$$
$$2. Compute\; C_1 = d \times (P_1 + P_2)$$
$$3. Compute\, e = d \times (k_1 P_1 + k_2 P_2)$$
$$4. compute\, C_2 = M + e$$
$$5. return\, (C_1, C_2)$$

The role of internal registers is to hold the values computed during the process of double point multiplication using algorithm 3 and the values are updated as per Table 1.

$$\text{Algorithm}: 3$$
$$Input \quad : P, Q \in E$$
$$a, b \in \{1-(n-1)\}$$
$$Output : aP + bQ$$
$$1. if\; a \neq b$$
$$2. Compute\, the\, values\, as\, per\, table$$
$$3. else$$
$$4. compute\, a(P+Q)$$
$$5. end$$
$$4. return\, ap + bQ$$

TABLE 1: UPDATE RULES FOR DOUBLE POINT MULTIPLICATION PROCESS

Rule	Case	a	b	P	Q	P-Q
R1	a>b and (a-b)/2 is Even	(a-b)/2	b	2P	P+Q	P-Q
R1*	a<b and (b-a)/2 is Even	a	(b-a)/2	P+Q	2P	P-Q
R2	a>b (or) a<b a/2 is Even	a/2	b	2P	Q	P+Q
R2*	a>b (or) a<b b/2 is Even	a	b/2	P	2Q	P+(-Q)

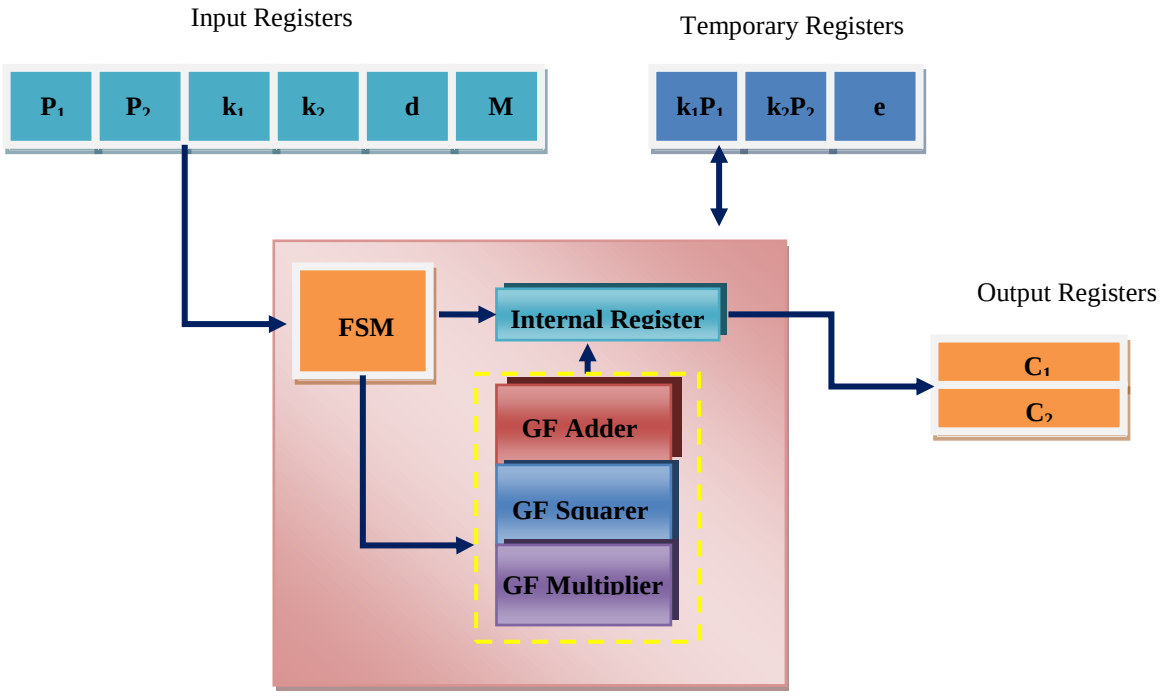

Fig. 2: Architecture of the low footprint security chip

Fig. 2 shows the architecture of the low footprint security chip based on the above ECC algorithm. The architecture based on revised and simplified double point multiplication algorithm, comprises of Input registers, Output Registers, Temporary registers and internal registers and three modules GF Adder, GF Squarer and GF Multiplier. The input registers hold the input parameters $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{d}, \boldsymbol{M}$. The temporary register holds the intermediate results generated during the process of key generation and encryption and are used for the final cipher text generation.

The sub modules in Security chip are GF Adder, GF Squarer and GF Multiplier. GF Adder performs modulo-2 addition and can be implemented by using an array of $\boldsymbol{l}$ XOR gates.

Algorithm : 4

Input : $\alpha(x) \in GF(2^l)$ *of order* $l-1, f(x)$

Output : $\alpha(x)^2$

1. *Initially* $\delta[k] = 0 \forall k : 0 \text{ to } 2l-1$

2. *for* $k : 0 \text{ to } l-1$

3. $\delta[2k] \leftarrow \alpha[k]$

4. *end*

4. *compute* $a(x)^2 = \delta(x) \bmod f(x)$

5. *return* $\alpha(x)^2$

GF Squarer simply inserts zeros between each bit in the bit-vector representation of the element followed by a reduction operation using the irreducible polynomial $\boldsymbol{f(x)}$ with respect to the bit length l .The algorithm 4 given below unfolds the squaring operation performed in proposed work.

Since the hardware complexity of a Security chip is based on the GF multiplier employed, area efficiency can be improved by adopting area efficient multiplier design. Area efficiency is achieved in multiplier by integrating two different versions of multiplier, Karatsuba and Systolic version [17].

B. Low footprint Hybrid Multiplier Design

Multiplication and Inversion are the most complex design in Security chip processor design and majority of the available resources are consumed by these blocks itself. Although, the inversion requires more resources and time when comparing with the multiplication operation, this can be realized simply by using single inversion for the complete scalar multiplication process, by adopting suitable point representations. It is additionally essential to advance the multiplication algorithms.

The finite field multiplier structures the most imperative part in the elliptic curve crypto processor (ECCP). It consumes the most space on the device and furthermore has the longest latency. The execution of the ECCP is influenced most by the multiplier. Finite field multiplication of two components in the field $GF(2^l)$ is characterized as:

$$\gamma(x) = \alpha(x).\beta(x) \bmod f(x)$$

Where $\alpha(x), \beta(x)$ and $\gamma(x)$ are in $GF(2^l)$ and $f(x)$ is the irreducible polynomial that generates the field $GF(2^l)$. Implementing the multiplication requires two steps. First, the polynomial product $\gamma(x)^* = \alpha(x).\beta(x)$ is determined, and then modulo operation is done on $\gamma(x)^*$. This can be given as,

$$\alpha(x) = \sum_{k=0}^{l-1} \alpha_k x^k = \alpha_0 x^0 + \alpha_1 x^1 + \ldots\ldots + \alpha_{l-1} x^{l-1}$$

$$\beta(x) = \sum_{k=0}^{l-1} \beta_k x^k = \beta_0 x^0 + \beta_1 x^1 + \ldots\ldots + \beta_{l-1} x^{l-1}$$

$f(x)$ is represented as,

$$f(x) = x^l + \sum_{k=0}^{l-1} f_k x^k$$

The proposed hybrid multiplier module integrates two variants. The first Multiplier variant is the traditional basic recursive Karatsuba multiplier described in algorithm 5.

Upon investigation as shown in Table 2, up to the multiplicand size 29, this algorithm uses fairly stable number of slice registers and fewer LUTs but LUTs rises phenomenally as multiplicand size rises above 29. Hence up to the multiplicand size 29, this algorithm is most suited.

Second version of the multiplier uses systolic approach, splits word into least significant and most significant words and using the shifting operation generates the product. This systolic multiplier algorithm 6, as indicated in Table 2, is found to be highly LUT efficient after m>29 as compared to Karatsuba Multiplier. In proposed low footprint hybrid multiplier based on HMUL Algorithm 7, the m bit multiplicands are split into two parts. When the number of bits is greater than or equal to the threshold 29, Systolic version is invoked. As the number of bits of the multiplicand falls less than 29, the Karatsuba algorithm is invoked. The proposed hybrid multiplier does the initial recursion using the systolic algorithm while final small sized multiplications are accomplished using the Karatsuba algorithm.

Algorithm : 5

KMUL (Karatsuba Multiplier Algorithm)

Input: A,B are multiplicands of m bits

Output: C of length 2m − 1 bits

/* **Define :** Mx → AxBx */

/* **Define :** M(x,y) → (Ax + Ay)(Bx + By) */

begin

for i = 0 *to* m − 2 **do**

Ci = C_{2m-2-i} = 0

for j = 0 to[i/2] **do**

if i = 2j then

$C_i = C_i + M_j$

$C_{2m-2-i} = C_{2m-2-i} + M_{m-1-j}$

else

$C_i = C_i + M_j + M_{i-j} + M_{(j,i-j)}$

$C_{2m-2-i} = C_{2m-2-i} + M_{m-1-j} + M_{m-1-i+j} + M_{(m-1-j,m-1-i+j)}$

end

end

end

$C_{m-1} = 0$

for j = 0 to [(m − 1)/2] **do**

if m − 1 = 2j **then**

$C_{m-1} = C_{m-1} + M_j$

else

$C_{m-1} = C_{m-1} + M_j + M_{m-1-j} + M_{(j,m-1-j)}$

end

end

end

Algorithm : 6

SMUL (Systolic Multiplier algorithm)

Input: The multiplicands A, B and their length m

Output: C of length 2m − 1 bits

begin

l = [m/2]

A′ = A[m−1….l] + A[l−1….0]

B′ = B[m−1…..l] + B[l−1….0]

C_{p1} =**HMUL** (A[l−1…0],B[l−1…0], l)

C_{p2} = **HMUL** (A′ ,B′ , l)

C_{p3} = **HMUL** (A[m−1…l],B[m−1…l], m − l)

return (C_{p3} left shift 2l) + (C_{p1} + C_{p2} + C_{p3}) left shift l + C_{p1}

end

TABLE 2: RESOURCE UTILIZATION SUMMARY BY TWO MULTIPLIERS

			Karatsuba Multiplier			Systolic Multiplier		
Multiplicand sizes	*Slice Register*	*LUTs*	*Slices*	*IOBs*	*Slice Register*	*LUTs*	*Slices*	*IOBs*
2	227	98	124	259	240	118	127	259
4	236	171	168	259	265	153	146	259
8	242	316	237	259	307	248	191	259
16	245	426	292	259	387	482	307	259
29	471	629	412	259	455	562	362	259
32	525	739	473	259	467	580	374	259
64	534	723	469	259	467	597	381	259
128	620	851	576	259	467	599	382	259
256	579	1535	916	259	467	583	374	259

Algorithm : 7

HMUL (Hybrid Multiplier algorithm)

Input: The multiplicands A, B and their length m
Output: C of length 2m − 1 bits
begin
  if m < 29 **then**
    return KMUL(A,B,m)
  else
    return SMUL(A,B,m)
  end
end

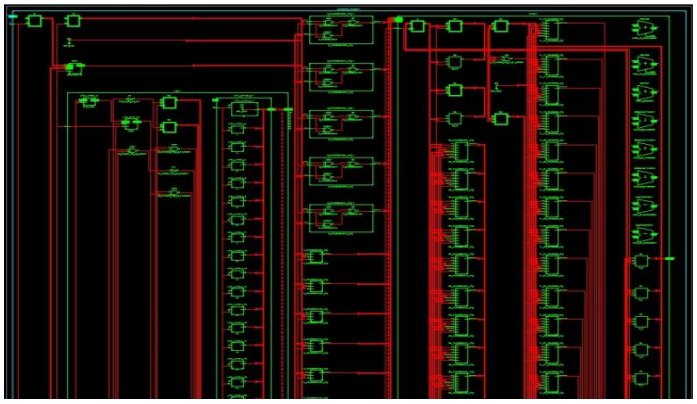

Fig. 3: RTL Schematic of proposed Hybrid Finite Field Multiplier

Thus by intelligently switching between the two multiplication algorithms, proposed low print hybrid multiplier performs multiplication activity with minimum resources and contributes to the reduction in overall footprint.

V. Results And Discussion

In this section, the FPGA implementations of the proposed architecture of FPGA based low footprint Security Chip using proposed ECC algorithm and hybrid multiplier is carried out to investigate area footprint. The proposed architectures are coded in verilog HDL and are synthesized using Xilinx ISE version 14.4 design software and implemented on Xilinx Virtex-4 xc4vlx200ff1513 FPGA. The RTL schematic for the implemented Hybrid Finite Field Multiplier and security chip is shown in Fig. 3 and Fig. 4.

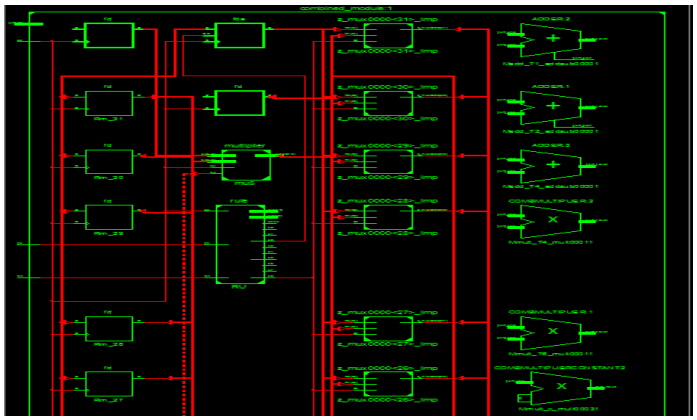

Fig. 4: RTL Schematic of Low Footprint Security Chip

TABLE 3: RESOURCE UTILIZATION SUMMARY OF LOW FOOTPRINT SECURITY CHIP

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Total Number Slice Registers	383	178,176	1%
Number used as Flip Flops	381		
Number used as Latches	2		
Number of 4 input LUTs	412	178,176	1%
Number of occupied Slices	308	89,088	1%
Number of Slices containing only related logic	308	308	100%
Number of Slices containing unrelated logic	0	308	0%
Total Number of 4 input LUTs	456	178,176	1%
Number used as logic	378		
Number used as a route-thru	44		
Number used as Shift registers	34		
Number of bonded IOBs	255	960	26%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Number of DSP48s	7	96	7%
Average Fanout of Non-Clock Nets	1.79		

TABLE 4: AREA COMPARISION RESULTS FOR ECC BASED SECURITY CHIP USING DIFFERENT MULTIPLIERS

ECC Algorithm	*m*	Target Device	Slices	LUTs	IOBs
[18]	193	Virtex4	466	932	932
Karatsuba	233	Virtex4	406	488	247
Systolic	233	Virtex4	373	536	259
Hybrid	233	Virtex4	308	456	255

A. Area Report

The device utilization summary tabulated in Table 3 depicts that proposed hardware implementation of the ECC based security chip utilizes 308 out of 89,088 available slices in the target device and 456 out of 178,176 available LUTs. Among the 383 of 178,176 Slice registers present, 381 act as Flipflops and two of them is used as latch. 7 out of 96 DSP-48 blocks are utilized by implemented architecture.

The comparision of Security chip using different multiplier types is carried out in Table 4. Proposed low footprint architecture of Security chip using Hybrid multiplier utilizes about 36.23 % reduced slices ,51.07% reduced LUTs and 72.63% IOBs. when compared to the similar work in [18].

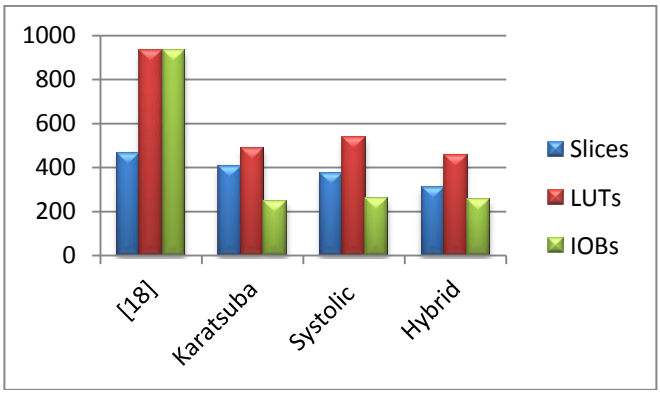

Fig. 5: Resource Utilization by ECC based Security chip using different Multipliers

B. Power Report

Power consumed by low footprint hybrid Multiplier is tabulated in Table 5 and Table 6. The clock frequency is varied between 100 MHz to 1800 MHz and the corresponding power consumed is plotted in Fig. 6 and Fig. 7.

TABLE 5: POWER REPORT FOR PROPOSED HYBRID FINITE FIELD MULTIPLIER (M< 29)

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.922	0.111	0.812
Vccaux	2.500	0.150	0.000	0.150
Vcco25	2.500	0.003	0.000	0.003
		Total	Dynamic	Quiescent
Supply Power (W)		1.488	0.133	1.355

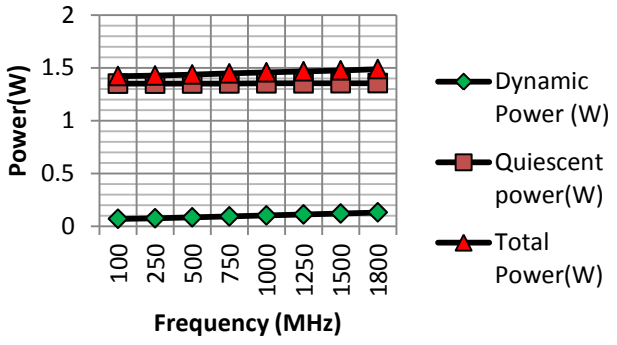

Fig.6: Power consumption with increase in clock frequency (m < 29)

TABLE 6: POWER REPORT FOR PROPOSED HYBRID FINITE FIELD MULTIPLIER (M ≥ 29)

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.842	0.037	0.805
Vccaux	2.500	0.150	0.000	0.150
Vcco25	2.500	0.003	0.000	0.003
		Total	Dynamic	Quiescent
Supply Power (W)		1.392	0.044	1.347

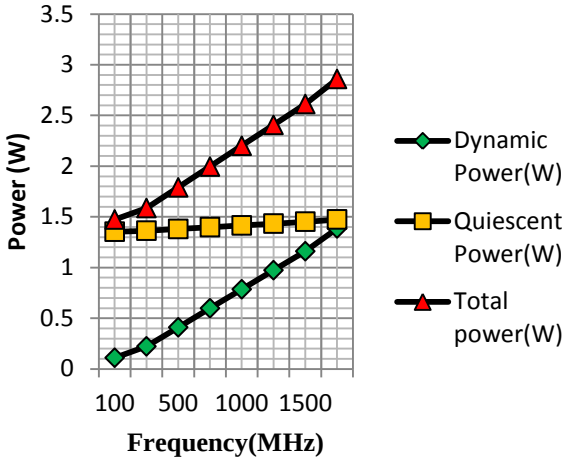

Fig.7: Power consumption with increase in clock frequency (m ≥ 29)

VI. CONCLUSION

In this work, FPGA architecture of low footprint Security chip was designed and implemented using Xilinx ISE software. Area efficiency was achieved by revising and simplifying the double point multiplication ECC algorithm and further reduced by using area efficient hybrid multiplier. The experimental results when compared with the results reported in similar work showed that proposed architecture for ECC processor in security chip utilizes about 36.23 % reduced slices, 51.07% reduced LUTs and 72.63% IOBs. Hence proposed hardware implementation of chip occupies and utilizes very small percentage of FPGA area.

REFERENCES

[1]. T. Wollinger, J. Guajardo, and C. Paar, "Security on FPGA: state of the art implementations and attacks,"ACM Transactions in Embedded Computing Systems, vol.3, pp.53–59, 2004.

[2]. K. Ravikumar, A.Udhayakumar, "A Detailed Study of Elliptic Curve Cryptography Algorithm and Its Performance," International Journal of Engineering Sciences & Research Technology, pp:2960-2964, October, 2013.

[3]. Amir Moradi, Alessandro Barenghi, Christof Paar and Timo Kasper, "On the Vulnerability of FPGA Bitstream Encryption against Power Analysis Attacks," Proceedings of the 18th ACM conference on Computer and communications, pp: 111-124, October 2011.

[4]. Anupama T, Dr. M. B. Manjunath, "Fpga implementation of elliptic curve crypto processor over gf(2^{163}) : A Review," International Journal of Science, Engineering and Technology Research (IJSETR) , Volume 3, Issue 5, May 2014.

[5]. A. Kaleel Rahuman and G. Athisha, "Reconfigurable Architecture for Elliptic Curve Cryptography Using FPGA," Mathematical Problems in Engineering, 2013.

[6]. A.Kaleel Rahuman, Dr. G.Athisha, "Reconfigurable Architecture for Elliptic Curve Cryptography," Proceedings of the International Conference on Communication and Computational Intelligence, pp.461-466, December- 2010.

[7]. Jyu-Yuan Lai and Chih-Tsun Huang Energy-Adaptive Dual-Field Processor for High-Performance Elliptic Curve Cryptographic Applications, IEEE Transactions onVery Large Scale Integration (VLSI) Systems,Vol.19 ,No.8,pp:1512 - 1517, Aug. 2011.

[8]. Sonia Sophia Joseph, S Prakash, "Area Efficient Implementation Of GF(2^m) Multipliers For Finite Fields," International Journal of Electrical, Electronics and Data Communication, Volume-2, No.5, May-2014.

[9]. Reza Azarderakhsh and Koray Karabina,"A new double point multiplication algorithm and its application to binary elliptic curves with endomorphisms", IEEE Transactions on Computers, No.99, May 2013.

[10]. Vladimir Tujillo-Olaya and Jaime Velasco-Medina, "Hardware architectures for elliptic curve cryptoprocessors using polynomial and gaussian normal basis over GF(2^{233})", Transaction on computer science XI, Vol.6480, pp: 79-103,2010.

[11]. Chiou-Yng Lee and Pramod Kumar Meher, "Speeding up subquadratic finite field multiplier over GF(2m) generated by trinomials using toeplitz matrix-vector with inner product formula", International conference on genetic and evolutionary computing, pp: 232-236, Sep.2011

[12]. Jen-Wei Lee, Szu-Chi Chung, Hsie-Chia Chang, Chen-Yi Lee "Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture", IEEE Transactions on very large scale integration (vlsi) systems, Vol. 22, NO. 1, pp: 49-61, Jan. 2014.

[13]. Jyu-Yuan Lai and Chih-Tsun Huang, "Energy-adaptive dual-field processor for high-performance elliptic curve cryptographic applications", IEEE Transactions on very large scale integration (vlsi) systems, Vol. 19, NO. 8, pp: 1512-1517, Aug. 2011

[14]. Hossein Mahdizadeh and Massoud Masoumi, ""Novel architecture for efficient FPGA implementation of elliptic curve cryptographic processor over GF(2^{163})", IEEE Transactions on very large scale integration (vlsi) systems, Vol. 21, NO. 12, pp: 2330- 2333, Dec.2013

[15]. Ashkan Hosseinzadeh Namin, Huapeng Wu,and Majid Ahmadi, "A word-level finite field multiplier using normal basis", IEEE Transactions on computers, Vol. 60, No. 6, pp: 890- 895, Jun. 2011

[16]. Johann Grobschadl, "Full-custom VLSI design of a unified multiplier for elliptic curve cryptography on RFID tags", International conference on information security and cryptography, Vol.6151, pp: 366-382, 2010.

[17]. Sunil Devidas Bobade and Dr. Vijay R.Mankar," Low footprint Hybrid Finite field multiplier for Embedded cryptography", International Journal of Computer Science and Information Security(IJCSIS), Pennsylvania, USA, ISSN: 1947-5500**,** Vol. 13, No. 3, pp: 28- 32,March 2015.

[18]. A. Kaleel Rahuman and G. Athisha, "Reconfigurable Architecture for Elliptic Curve CryptographyUsing FPGA," Hindawi Publishing Corporation Mathematical Problems in Engineering, 2013.

Sunil D. Bobade obtained his Engineering Graduate Degree in Electronics and Telecommunication Engineering from VYWS College of Engineering, Amravati (India) in 1994, Post Graduate Degree in Electronics Engineering from S.G.B.Amravati University, Amravati (India) in 2002.

He has been in the field of teaching since last 19 years and is presently, working as an Assistant Professor in Department of Information Technology, Datta Meghe College of Engineering, Navi Mumbai. He is also working as a research scholar in S.G.B.Amravati University and is working on development of area efficient algorithms for protection of memory of embedded systems.

Prof. Dr. Vijay R. Mankar received his B.E. degree in Electronics and Power Engineering from Government College of Engineering, Amravati (India) in 1986, M.Tech. in Electronics Engineering from erstwhile Visvesvaraya Regional College of Engineering, Nagpur (India) in 1990 and Ph.D. from S.G.B. Amravati University, Amravati (India) in 2009.

He has been in the field of teaching since last 23 years and is presently, working as a Deputy Secretary, MSBTE, Pune Region. He has served as Head of Department and the Professor, with Department of Electronics Engineering, Govt. Polytechnic, Amravati. He has been active in the research as well. His research interests are in Neural Networks, Design of Embedded system, Data security. He has published over twenty research papers in journals and conferences of national and international repute.