6.1 Conclusion

Cryptography has now become a well developed branch of computer science. Earlier it dealt with secret communication. Now cryptography is used on computer networks to achieve the target of confidentiality, reliability authentication, protection of privacy etc. Depending on number of keys, it is divided into private key cryptography and public key cryptography. In private key cryptography, same keys are used to encrypt and decrypt data and key has to be kept secret. In public key cryptography two keys are used one for encryption and one for decryption. Both keys depend on each other. Various private and public key algorithms available are AES, DES, RSA etc. cryptanalysis is performed to identify any potential weakness in available algorithms. Today, cryptography has its importance in various fields like mobile banking; secure business communications, security of government and military data and a lot of
others. The issues related to cryptography are standardization, protection of privacy vs. national security, lawful access etc.

Various fields of mathematics related to today’s cryptography are information theory, number theory, prime numbers generation and verification related algorithms prime field arithmetic etc. security of today’s cryptographic algorithms relies greatly on the properties of mathematical concepts used. Therefore it is necessary to have knowledge of the background mathematics used in today’s cryptography.

Importance of Elliptic Curves in Cryptography was independently proposed by Neal Koblitz and Victor Miller in 1985. Since then, Elliptic curve cryptography or ECC has evolved as a vast field for public key cryptography (PKC) systems. It is capable of providing same level of security with smaller key sizes as compared to other public key algorithms therefore it is mainly used for embedded platforms which are resource constrained systems.

Various research directions for elliptic curve cryptography on embedded systems are both software based and hardware based. A hardware implementation is basically focused on the processor design issues. To design any processor we have to consider flow of data into its input, output registers and various ALU units, called processor data path. Also flow of control information in the form of op-code and related micro-operations has to be considered called control logic. It is implemented in different ways like, extending instruction set of the processor or enhancing design of multiplier inside ALU. Following are the generalized design considerations proposed in various papers on elliptic curve cryptography for embedded systems;

- Enhancing the speed of multiplication.
Providing suitable options for very large data size to be implemented on data path of limited data size processor.

Providing suitable options to implement various non standard number formats, used in cryptographic applications.

Modifying the instruction set of domain specific processors for ECC operations

Performing ECC on server, multi-core systems, media processors etc.

Software implementations of ECC have the advantage of interoperability; we can enhance security with several security options while switching between different ECC schemes. Therefore ECC implementation using software is also seriously considered by the researchers.

Research in this category of ECC is mainly related to present various combinations of speed optimized algorithms that will reduce the requirement of separate crypto co processor. Also different mathematical techniques are considered to enhance the speed and security of ECC.

For this implementation binary fields are selected to perform field based arithmetic. They are finite fields of the form GF(2^m), where m is a prime number. This means these are prime numbers having prime number of bits. Various fields proposed are generally trinomials or pentanomilas which means these are bit-strings representing prime numbers, having prime length but, they contain either three or five ones within them only.

Prime field based addition/subtraction, modular multiplication and division are implemented using various optimization algorithms. Projective coordinate system is used to further enhance the speed.
6.2 New Findings

A hash table based data structure is proposed to store long keys of elliptic curve cryptography. Object oriented platform of C++ is used to implement this data structure for various ECC algorithms. Keys are divided into 64 bits blocks and stored as individual table entries. The base class defined is `longint` which is generalized class. It is designed to convert any given block of data into a table with sequential table entries of 64 bits each. It is capable of doing arithmetic operations on the given data independent of their size. It is also capable of performing arithmetic operations on varying bit length data. Apart from elliptic curve cryptography this class can be used to implement other cryptographic algorithms. This class can also be used to implement other mathematical algorithms in a data size independent environment.

Field based division by two is implemented as a binary shift right operation. Generally this is implemented in hardware platforms but it will result in fixed data size. Current implementation removes this restriction as right shifting is performed independent of data size. It right shifts the given number by \( c \) bits to reflect division by \( 2^c \).

Binary field based arithmetic operations are implemented in a class `prime_fld` which is derived from `longint` class. As per the property of binary fields, addition and subtraction are implemented as bitwise XOR operation in a single function `void add(prime_fld num)` the implementation has two unique features.

Modular multiplication is performed using Montgomery’s Algorithm. The algorithm is modified so as to operate upon proposed hash table structure.
Modular division is implemented as per the hardware based algorithm proposed by C. Shantz. This algorithm is based on Extended Euclidean Algorithm for GCD.

Elliptic Curve algorithm to verify that given points \( P(x,y) \) is on the curve is implemented which, again works independent of data size.

Point add and point double formulae are implemented based on projective coordinates as elliptic curve arithmetic in projective coordinate requires less operations. Point add is implemented as `void point_add(el_curv_pt q)` It performs addition of elliptic curve points \( P \) and \( Q \) as \( P=P+Q \). Thus no return value is generated. Point double is implemented as `void point_double()` It takes no value and returns none. It simply applies point double formula to the calling object. Point add works upon elliptic curve group law for mixed coordinates and point double works upon elliptic curve group law for projective coordinates.

Speed optimization is achieved by simple loop unrolling concept. Using this, about 21 to 59 percent speed enhancement is achieved.

### 6.3 Future Works

In future, first preference will be given to enhance the speed of current implementation. Various methods considered will be as follows

1. Speed critical functions or parts of the functions which are critical to speed will be implemented as assembly language modules.

2. As basic data type is one byte, larger byte sized data will be considered. It also means that an analysis has to be done how such change affects the performance on embedded platforms.
3. Basic array size will also be modified so that less null bytes are remaining at the MSB locations for all the accepted binary field based numbers.

4. Various software optimization techniques available for ECC like using Montgomery Multiplication for ECC based multiplication, precomputation schemes available for various large numbers will be considered.

5. The presented implementation will be tested on various optimized hardware platforms which are specially designed for ECC.

Due to restricted cryptographic parameters current implementation is highly specific. It works only for ECC on binary fields. Following improvements will be considered in future;

1. First of all an effort will be made to implement prime fields. This means normal addition and subtraction despite of XOR has to be implemented individually along with carry propagation.

2. All the algorithms used in this implementation are targeted to generate elliptic curve keys. An effort will be made in future to implement other elliptic curve cryptography based algorithms like elliptic curve based authentication, digital signature generation etc.

3. In future it will also be tried to analyze if the hash table structure proposed here also works for other cryptographic techniques like RSA.

Finally, following generalizations will be considered:

1. It will be tried to replace all the functions with operator overloading concept of object oriented platforms.
2. Other object oriented platforms will be considered for proposed data structure.

3. It will be tried to generalize the basic classes so that they can serve for mathematical and scientific research oriented works also.

Finally an effort will be made to implement presented work for general purpose computers.