Implementing ECC for Embedded Systems
Using Hash Table to Achieve Large Key Size

In today’s networked systems, cryptography is widely used for secure communication, validation, authentication and similar other purposes. Various algorithms for this purpose have been proposed based on mathematical irreversible problems. Research in the field of cryptography also has significant progress in the direction of cryptanalysis—the branch that deals with methods in finding weaknesses in presently used algorithms.

There are a lot of standard algorithms proposed and are in use today. Cryptography based on elliptic curves is latest of them all. Since its introduction in 1980, it has been proved strong to tolerate various cryptanalysis attacks. It provides security with comparatively low parameter size as compared to other algorithms. This is a very important aspect as security of encryption algorithms depend on large data size, more than hundred bits at least. The data size is always kept much larger than available processor word sizes to ensure security against hardware based cryptanalytical attacks. This poses the problem of speed especially in resource constrained embedded devices. Therefore not much attention is given to high level programming platforms. Also data are represented in as simple forms as possible. This adds rigidity in software modules. Every time security requirements change, related parameters have to be modified accordingly.

In this research object oriented platform of C++ is used to implement elliptic curve cryptography for embedded systems. The advantages and limitations and their impact on final outcome is studied. It is proposed to use
a hash table to store long data or more specifically keys. It is certainly going to affect the speed therefore various optimization techniques are considered. Various classes are designed to represent binary fields, ECC points and related functions. As the research is targeting embedded platforms, only those actions which are required for field arithmetic and ECC arithmetic are implemented to reduce code size.

In this research binary fields are used to perform field arithmetic which are generally used for hardware based implementations. Basic arithmetic operations are modified accordingly. Field based multiplication is performed using Montgomery multiplication. Field based division is based on extended Euclidean algorithm for GCD.

As division is the most time consuming operation it is not used while implementing ECC related operations. It is possible as ECC arithmetic is done in projective coordinate system which does not require division. Mixed coordinate system is also used as it enhances the speed.

Various optimization techniques are currently available for both field arithmetic and ECC arithmetic. Few of them are studied and implemented. Further enhancement options are also proposed. All the algorithms are modified to accommodate proposed data structure.

The data structure presented has been made to operate on all the functions independent of data size. It is also made capable to operate upon varying length data in dyadic operations. This may be useful in general for a variety of other applications like mathematical and scientific fields.