Elliptic-Curve Digital-Signature System Based on Radix-2^w Arithmetic

F. Nait-Abdesselam[1, 2], A.K. Oudjida[2]
[1]Institute of Electrical and Electronic Engineering,
Boumerdes University
[2]Center for Development of Advanced Technologies
Algiers, Algeria
fnait@cdta.dz

A.Khouas[1], A. Bourahla[1], L. Nebhi[1]
[1]Institute of Electrical and Electronic Engineering,
Boumerdes University
Algiers, Algeria
ab.khouas@univ-boumerdes.dz

Abstract— **The Elliptic Curve Digital Signature Algorithm (ECDSA) is a widely recognized cryptographic tool that plays a crucial role in verifying the authenticity and preserving the integrity of digital messages. What distinguishes ECDSA from its peers is its exceptional ability to provide robust security with smaller key sizes, making it highly efficient, particularly in resource-constrained applications. Based on the flexibility of the Radix-2^w arithmetic, we propose new simple methods to ECDSA implementation handling efficiently the intricacies of elliptic curve point multiplication (PM) and double point multiplication (DPM) operations. The implementation incorporates a Radix-2^w method for PM and a simultaneous Radix-2^w method for DPM. Both methods are streamlined one-pass recode/evaluation processes. These methods reduce the number of point additions (ADDs) to a near-optimal limit in a sublinear computational time without increasing the number of point doublings (DBLs). They also have the advantage to require a minimal number of point-precomputations. In addition, they show high resilience against side-channel attacks. In comparison to the standard binary method used in PM and DPM, the new Radix-2^w methods shows substantial savings in computational time.**

Keywords— Elliptic curve digital signature algorithm (ECDSA), Double point multiplication (DPM), Point multiplication (PM), Radix-2^w arithmetic, Side-Channel Attacks.

I. INTODUCTION

In the context of modern day digital communication, cryptography is concerned with the conception of protocols adaptable to computational-power of existing machines. However, the growing memory and speed capacities of machines entail the continual creation and optimization of cryptographic algorithms satisfying the security, data protection, and information integrity communication requirements. To handle such an aspect of security, a number of cryptographic protocols were developed. The National Institute of Standards and Technology (NIST) approved the Elliptic Curve Digital Signature Algorithm (ECDSA) as a secure digital signature technique [1]. ECDSA ensures robust security with shorter key lengths, making it well-suited for resource-constrained environments, all the while delivering swift computations [2].

Digital signature cryptographic schemes rely on the use of keys in their operations. They are used in the computations of both signature generation and verification, and their size positively correlates with the level of security provided. Leveraging the mathematical intricacies of the elliptic curve discrete logarithm problem (ECDLP), EC effectively minimize key size while upholding a high level of security. This, in turn, translates to expedited operations [3]. Optimizations for speed, memory, energy, etc., go beyond just minimizing key sizes, but extend to enhancing the computational aspects of the signature algorithms [4][5][6].

EC point multiplication (PM) used in the ECDSA is considered the most costly operation of the protocol, making it therefore a target for optimization. It involves the multiplication of a point P belonging to an EC by a scalar k ($k \times P$). The addition of two or more PMs being commonly required in cryptographic algorithms, Double Point Multiplication (DPM) and Multiple Point Multiplication, gave rise to further optimizations in EC Multiplication [7][8][9]. The basic EC point operations are the point addition $P+Q$ (ADD) and point doubling $P+P$ (DBL) used to evaluate the multiplications PM and DPM. A more efficient technique used to reduce the overall cost of the multiplication in terms of ADDs and DBLs is the recoding of the scalar k [10]. A well-used method for computing PM that employs the recoding of k is the windowed non-Adjacent form (w-NAF) [11]. However, this method achieves ADD cost reduction at the expense of a substantial manipulation of extra intermediate memory variables. The Radix-2^w method for PM successfully addresses the limitations of the w-NAF by employing a left-to-right recoding of k, aligning seamlessly with the inherent nature of ECs. This not only eradicates the necessity for intermediate variables but also exe-cutes recoding and evaluation on-the-fly. In addition, it reduces the cost of ADDs to a near-optimal limit in a sub-linear computational time without increasing the number of DBLs [12].

The primary objective of this work is to implement the ECDSA using the Radix-2^w method for EC PM and DPM on PC platform using C language. The use of the Radix-2^w arithmetic for recoding the EC PM and DPM is central to this work since our aim is to provide its proof of concept. Since ECDSA does not target the confidentiality aspect of security, it is implemented along with an encryption protocol to complement it. Due to its adaptability to elliptic curves, ElGamal Encryption algorithm is used [13].

979-8-3503-4935-1/24/$31.00 ©2024 IEEE