A High-Performance Curve25519 and Curve448 Unified Elliptic Curve Cryptography Accelerator

Aniket Banerjee and Utsav Banerjee

Electronic Systems Engineering

Indian Institute of Science, Bengaluru, India

Email: aniketb@iisc.ac.in, utsav@iisc.ac.in

Abstract—In modern critical infrastructure such as power grids, it is crucial to ensure security of data communications between network-connected devices while following strict latency criteria. This necessitates the use of cryptographic hardware accelerators. We propose a high-performance unified elliptic curve cryptography accelerator supporting NIST standard Montgomery curves Curve25519 and Curve448 at 128-bit and 224-bit security levels respectively. Our accelerator implements extensive parallel processing of Karatsuba-style large-integer multiplications, restructures arithmetic operations in the Montgomery Ladder and exploits special mathematical properties of the underlying pseudo-Mersenne and Solinas prime fields for optimized performance. Our design ensures efficient resource sharing across both curve computations and also incorporates several standard side-channel countermeasures. Our ASIC implementation achieves record performance and energy of 10.38 μs / 54.01 μs and 0.72 μJ / 3.73 μJ respectively for Curve25519 / Curve448, which is significantly better than state-of-the-art.

Index Terms—elliptic curve cryptography, ASIC, Curve25519, Curve448, NIST standard, unified hardware accelerator, high performance, side-channel countermeasures.

I. INTRODUCTION

The need for enhanced cybersecurity is more critical than ever in today's interconnected world. As digital communications and transactions become ubiquitous, ensuring information confidentiality, integrity and authenticity has become a paramount concern. Public Key Cryptography (PKC) [1], [2] is pivotal in securing these communications by enabling key establishment, digital signatures and authentication protocols. This is especially important in the Internet of Things (IoT) such as industrial automation, sensors, power grids, smart cities and automotive applications [3]. As these sectors continue to digitize, ensuring robust security while maintaining operational efficiency becomes a significant challenge. For example, digital communications between intelligent electronic devices in modern electrical substations follow the IEC 61850 standard [4], [5] and their security recommendations are provided by the IEC 62351 standard [6], [7]. These protocols demand low latency and high reliability for associated messaging protocols for real-time operation, e.g., 3 ms and 250 μs for GOOSE (Generic Object Oriented Substation Event) and Sampled Value (SV) messages respectively, thus making it extremely challenging to implement strong cryptographic security measures in critical infrastructure [8]. Due to the use of embedded systems, it is also crucial to ensure hardware resilience against side-channel attacks [9].

Cryptographic hardware accelerators are widely used to meet application-specific requirements such as low power, high performance and energy efficiency which are not achievable using software implementations with general purpose micro-processors [10]. Elliptic Curve Cryptography (ECC) is the current standard for PKC algorithms due to small key sizes [11], [12], and the U.S. National Institute of Standards and Technology (NIST) has recently recommended two new elliptic curves Curve25519 and Curve448 [13]. Both Curve25519 and Curve448 are Montgomery curves which stand out for their exceptional performance and security properties, as specified in [14]. Curve25519, introduced by Daniel J. Bernstein in 2006 [15], is renowned for its speed and robustness against side-channel attacks. Curve448, introduced by Mike Hamburg in 2015 [16], offers a higher security level while maintaining efficiency and side-channel resilience. In spite of the advent of new quantum-safe cryptography algorithms, elliptic curves such as Curve25519 and Curve448 continue to play a vital role in enabling post-quantum hybrid key exchange protocols due to the strong confidence in their security [17].

Previous literature has presented various dedicated hardware accelerator designs for Curve25519 and Curve448 implemented in FPGA (field-programmable gate array) and ASIC (application-specific integrated circuit) [18]–[40]. Despite the numerous similarities between Curve25519 and Curve448, a unified hardware architecture supporting ECC with both curves is yet to be explored. In this work, we present the ASIC implementation of a high-performance unified hardware accelerator which can be configured to perform elliptic curve scalar multiplication over both Curve25519 and Curve448. Since finite field arithmetic is the most expensive component of these computations, we propose an efficient modular arithmetic architecture with four 256-bit 2-level Karatsuba multipliers. This allows us to exploit the power of parallel processing for Curve25519, while the same can also be fully re-used for Curve448 by exploiting the special structure and mathematical properties of its underlying prime field. We re-structure and re-arrange the sequence of operations in the Montgomery Ladder to reduce the latency. Our design is constant-time by design and also incorporates the randomized projective coordinate countermeasure against power side-channel attacks. Compared to previous designs, our proposed accelerator excels in terms of both performance and energy-efficiency while supporting a higher security level along with side-channel countermeasures.

II. BACKGROUND

A. Elliptic Curve Cryptography (ECC)

An elliptic curve E over a finite field $\mathbb{K}$ is defined as

$$E : y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6$$

where $a_1, a_2, a_3, a_4, a_5, a_6 \in \mathbb{K}$. There are two major types of elliptic curves defined over finite fields where the characteristic char($\mathbb{K}$) is a very large prime p:

- Short Weierstrass curves consisting of the set of points $E(\mathbb{F}_p) = \{(x, y) \mid y^2 = x^3 + ax + b \pmod{p}\} \cup \mathcal{O}$
- Montgomery curves consisting of the set of points $E(\mathbb{F}_p)$ $= \{(x, y) \mid by^2 = x^3 + ax^2 + x \pmod{p}\} \cup \mathcal{O}$

where $a, b \in \mathbb{F}_p$ are the curve parameters and $\mathcal{O}$ is the distinguished point at infinity.

The fundamental operations in ECC are *point addition* ($R = P + Q$) and *point doubling* ($R = P + P$), where $P, Q, R \in E(\mathbb{F}_p)$. With these operations, the points on the curve $E(\mathbb{F}_p)$ form an abelian group, with $\mathcal{O}$ serving as the identity element, that is, $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E(\mathbb{F}_p)$. The order of this group (number of points in $E(\mathbb{F}_p)$) is denoted by $\#E(\mathbb{F}_p) = n$, and $nP = \mathcal{O}$ for all $P \in E(\mathbb{F}_p)$.

Repeated additions of a point P with itself is called *elliptic curve scalar multiplication* (ECSM). For any scalar k, the scalar multiple kP is computed as

$$\underbrace{kP = P + P + \cdots + P}_{(k\text{-1}) \text{ point additions}}$$

This computation forms the basis of the *elliptic curve discrete logarithm problem* (ECDLP) – determine scalar k given the elliptic curve $E(\mathbb{F}_p)$ of order n, and the points $P, Q \in E(\mathbb{F}_p)$ such that $Q = kP$. For a t-bit prime p, the fastest known algorithms that can solve ECDLP have time complexity $O(2^{t/2})$ [12]. For sufficiently large primes and appropriate curve parameters, it is infeasible for a computationally bounded (non-quantum) adversary to solve ECDLP, and this guarantees the security of ECC and associated public key protocols.

B. Curve25519

Curve25519 is a Montgomery curve defined by the equation

$$y^2 = x^3 + 486662x^2 + x$$

over the prime field $\mathbb{F}_{2^{255}-19}$ [15]. It is optimized for key exchange at the 128-bit security level and has smaller key sizes and faster computations compared to traditional elliptic curves. It is designed to resist side-channel attacks, making it suitable for a wide range of applications including secure communications. Its simplicity and efficiency have led to its widespread adoption in various standards [13], [14].

C. Curve448

Curve448 is a Montgomery curve defined by the equation

$$y^2 = x^3 + 156326x^2 + x$$

over the prime field $\mathbb{F}_{2^{448}-2^{224}-1}$ [16]. It offers a higher 224-bit security level while maintaining similar characteristics

as Curve25519. It is optimized for cryptographic protocols requiring robust security such as key exchange and digital signatures. Its resistance to side-channel attacks makes it suitable for high-security applications [13], [14].

D. Hardware Acceleration of Curve25519 and Curve448

Previous work on hardware implementations of Curve25519 have been mostly based on Zynq 7000 series FPGAs [18], [19], [22], [24]–[27], [31]–[34], [36], [37]. The general approach has been to efficiently utilize the on-chip DSP and BRAM slices to speed up the ECSM operation. Various side-channel countermeasures such as scalar blinding, randomized projective coordinates and memory address scrambling have also been proposed [41]. Most recently, [39] demonstrated a high-performance design consisting of compute groups with processing elements (PEs), massive parallelism and a high degree of pipelining implemented in a Zynq 7000 series FPGA. The first compact low-power ASIC implementation of Curve25519 was presented in [20], and this was subsequently optimized for high-performance in [28], [40]. A similar approach was followed for FPGA-based hardware implementations of Curve448 [21], [23], [29], [35], [38], with ECSM performance significantly slower than Curve25519 due to the increased computational complexity. Various implementation strategies for Curve448 hardware architectures in FPGA, such as light-weight, area-time-efficient and high-performance, were investigated by [30]. Efficient ASIC implementations of Curve448 are yet to be explored. Also, unified hardware architectures for accelerating ECSM over Curve25519 and Curve448 have not yet been demonstrated in state-of-the-art, thus motivating our proposed design in this work.

III. ACCELERATOR ARCHITECTURE

A. ECSM Computation using Montgomery Ladder

Curve25519 and Curve448 are elliptic curves offering 128-bit and 224-bit of security levels respectively. Both curves use the Montgomery form, enabling fast and constant-time elliptic curve scalar multiplication, which is crucial for secure implementation. Supporting both curves in the same hardware accelerator facilitates interoperability and flexibility, allowing systems to choose the appropriate curve based on application-specific security needs. The proposed unified implementation also benefits from sharing hardware resources for the core arithmetic computations, thus reducing overall power and area without compromising performance.

Algorithm 1 shows how an ECSM computation is performed on a Montgomery curve with t-bit scalar $k = (k_{t-1}, k_{t-2}, \cdots, k_2, k_1, k_0)_2$. Note that $t = 255$ and $t = 448$ for Curve25519 and Curve448, respectively. The input and output points P and Q are specified by their x-coordinates x_P and x_Q, respectively. For Curve25519 and Curve448, x_P and x_Q will be elements in their respective prime fields $\mathbb{F}_{2^{255}-19}$ and $\mathbb{F}_{2^{448}-2^{224}-1}$. This algorithm is inherently constant-time, that is, execution time is independent of the secret scalar k. This helps prevent timing and simple power analysis (SPA) attacks. In order to prevent more sophisticated differential power

Algorithm 1 ECSM using the Montgomery Ladder [42]

Require: input point P with x-coordinate x_P and t-bit secret scalar $k = (k_{t-1}, k_{t-2}, \cdots, k_2, k_1, k_0)_2$

Ensure: output point $Q = kP$ with x-coordinate x_Q

1: $X_1 \leftarrow x_P$, $X_2 \leftarrow 1$, $X_3 \leftarrow x_P$
2: $Z_1 \leftarrow 1$, $Z_2 \leftarrow 0$, $Z_3 \leftarrow 1$
3: **for** $(i = t - 1; i \geq 0; i = i - 1)$ **do**
4:     **if** $k_i = 1$ **then**
5:         $(X_3, Z_3, X_2, Z_2) \leftarrow \text{LADDER}(X_1, X_3, Z_3, X_2, Z_2)$
6:     **else**
7:         $(X_2, Z_2, X_3, Z_3) \leftarrow \text{LADDER}(X_1, X_2, Z_2, X_3, Z_3)$
8:     **end if**
9: **end for**
10: $Z_2 \leftarrow Z_2^{-1}$
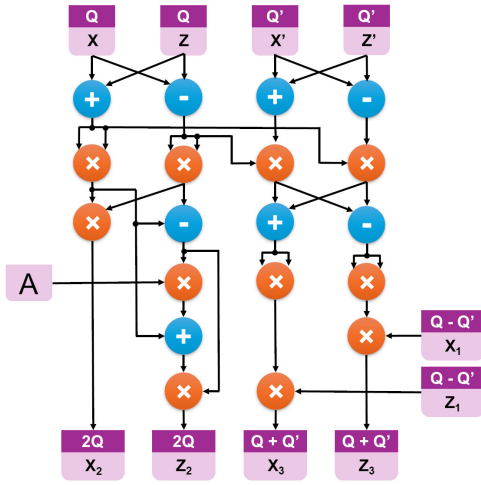11: $x_Q \leftarrow X_2 Z_2$
12: **return** x_Q

Fig. 2. Top-level block diagram of the proposed unified Curve25519 and Curve448 elliptic curve cryptography accelerator.

Fig. 1. Modular arithmetic operations in the Montgomery Ladder.

Fig. 3. Restructuring of arithmetic operations in the LADDER computation.

analysis (DPA) attacks, the randomized projective coordinate technique can be used [41]. This involves first generating a pseudo-random element λ in the underlying field ($\mathbb{F}_{2^{255}-19}$ for Curve25519 and $\mathbb{F}_{2^{448}-2^{224}-1}$ for Curve448). Then, steps 1 and 2 in Algorithm 1 are modified as $X_1 \leftarrow \lambda x_P$, $X_2 \leftarrow \lambda$, $X_3 \leftarrow \lambda x_P$ and $Z_1 \leftarrow \lambda$, $Z_2 \leftarrow 0$, $Z_3 \leftarrow \lambda$ respectively. Rest of the ECSM computation remains unchanged, and the modular division in steps 10 and 11 in Algorithm 1 ensure that the final output is correct irrespective of the value of λ.

The most important step in Algorithm 1 is the LAD-DER(.) function, which is the Montgomery Ladder [42]. Both Curve25519 and Curve448 ECSM computations employ the Montgomery Ladder to perform point double-and-add operations in projective coordinates, and its constituent modular arithmetic operations are shown in Fig. 1. There are 8 modular additions/subtractions and 11 modular multiplications/squarings (including multiplication by the curve constant A, where $A = 121665$ for Curve25519 and $A = 39081$ for Curve448) involved in each LADDER computation.
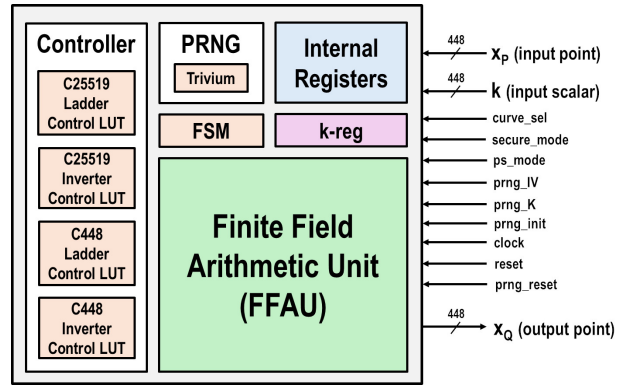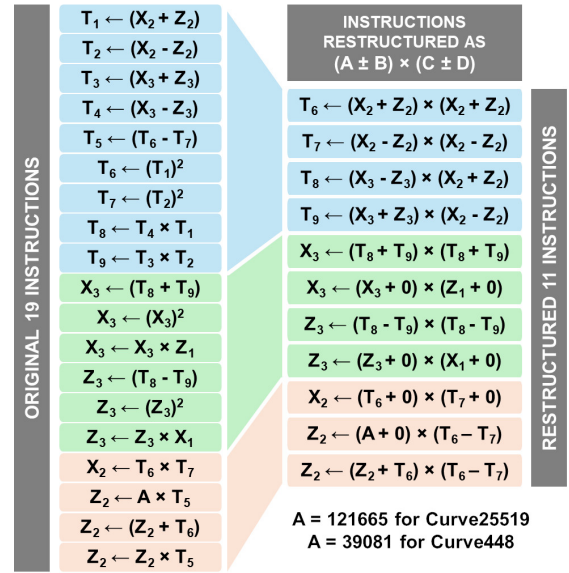
B. Accelerator Building Blocks

The top-level block diagram of our proposed hardware accelerator is shown in Fig. 2. The most important component of the accelerator is the Finite Field Arithmetic Unit (FFAU). A 448-bit k-reg register is used to store the secret scalar k. A unified controller module is used to send appropriate control signals and instructions to the main data-path, and they work in tandem with a finite state machine (FSM). 12×448-bit internal registers are used to store the inputs, outputs and temporary values generated during ECSM computation. The most significant 193 bits of all these registers are clock-gated for power savings when performing ECSM over Curve25519, while all 448 bits are utilized for Curve448. A pseudo-random number generator (PRNG) module, containing a hardware instantiation of the light-weight Trivium stream cipher [43], is used to generate the λ values for DPA countermeasures as discussed earlier. The PRNG is clock-gated for power savings when DPA countermeasures are disabled.
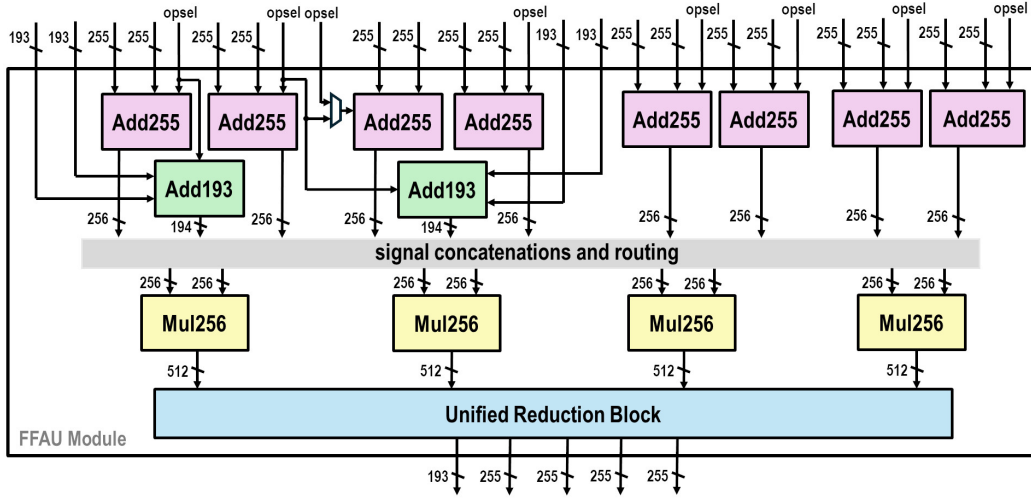
Fig. 4. Detailed architecture of the finite field arithmetic unit (FFAU) module.

C. Unified Controller

Inspired by the instruction mapping technique from [39], we restructure the 19 modular arithmetic operations in the LADDER computation from Fig. 1 as 11 steps in the form of $(A \pm B) \times (C \pm D)$. This restructuring is shown in Fig. 3. Control signals and register addresses for these steps in the LADDER computation for both curves are stored as instructions in lookup tables (LUTs), as shown in Fig. 2. For ECSM computation, 255 and 448 iterations of the LADDER are performed for Curve25519 and Curve448, respectively. The controller also contains similar LUTs for the modular inversion computations which will be discussed later.

D. Finite Field Arithmetic Unit (FFAU)

All the modular arithmetic operations in our accelerator are executed in the FFAU module shown in Fig. 4. It is capable of computing four $(A \pm B) \times (C \pm D)$ operations simultaneously, where A, B, C, D are 255-bit inputs. The FFAU has eight *opsel* (operation select) input lines corresponding to the four operations to select whether additions or subtractions need to be performed. The FFAU contains four 256-bit multipliers *Mul256* and eight 255-bit adder / subtractor modules *Add255*. It also contains two 193-bit adder / subtractor modules *Add193* to together compute 448-bit addition / subtraction. Clearly, the FFAU can perform four instructions together for Curve25519, thus completing a LADDER in just 3 clock cycles. When working with Curve448, the fact that its prime modulus is a Solinas trinomial prime $(= \phi^2 - \phi - 1)$ with the golden ratio $\phi = 2^{224}$ can be exploited. This allows the product of $A = (a_1\phi + a_0) \in \mathbb{F}_{\phi^2 - \phi - 1}$ and $B = (b_1\phi + b_0) \in \mathbb{F}_{\phi^2 - \phi - 1}$ to be calculated efficiently as: $C = A \times B \pmod{\phi^2 - \phi - 1} = (a_1\phi + a_0) \times (b_1\phi + b_0) \pmod{\phi^2 - \phi - 1} = (a_1b_1 + a_0b_0) + (a_1b_0 + a_0b_1 + a_0b_0)\phi \pmod{\phi^2 - \phi - 1}$. Here, a_0, a_1, b_0, b_1 being 224-bit quantities, it is possible to perform this computation using the four 256-bit multipliers. Therefore, the FFAU can perform only one instruction at a

time for Curve448. Consequently, 10 clock cycles are required to complete a LADDER (11 clock cycles with DPA countermeasure enabled). The multiplier outputs are finally processed by the *Unified Reduction Block* which employs fast reduction algorithms to compute the final result. Details of the FFAU sub-module implementations are described as follows:

1) 256-bit Multiplier: The Mul256 modules are implemented as 256-bit 2-level Karatsuba multipliers [44]. Using Karatsuba's algorithm, the product of two $2b$-bit unsigned integers $X = x_1 2^b + x_0$ and $Y = y_1 2^b + y_0$ can be calculated as $Z = XY = x_1y_1 2^{2b} + (x_0y_1 + x_1y_0)2^b + x_0y_0 = x_1y_1 2^{2b} + [(x_0 + x_1)(y_0 + y_1) - (x_0y_0 + x_1y_1)]2^b + x_0y_0$, that is, three instead of four b-bit $\times$ b-bit multiplications at the cost of some extra additions / subtractions. This reduces the complexity of n-bit large integer multiplication from $O(n^2)$ to $O(n^{\log_2(3)})$, or approximately $O(n^{1.585})$. Fig. 5 shows the block diagram for this construction. The Mul_b blocks compute x_0y_0 and x_1y_1, while the Add_b blocks compute $x_0 + x_1$ and $y_0 + y_1$. Then, the final result is computed by performing one more multiplication followed by appropriate additions, subtractions and shifting. These are efficiently handled by
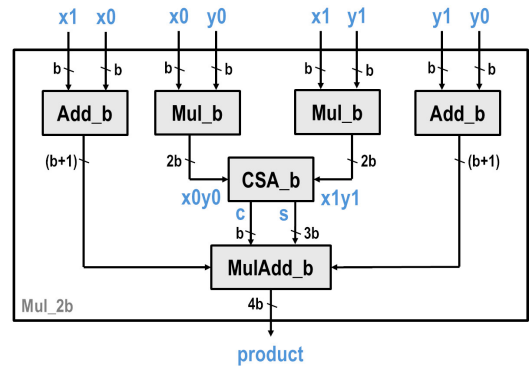

Fig. 5. Block diagram of $2b$-bit $\times$ $2b$-bit Karatsuba multiplier ($b = 128$ and $b = 64$ for Mul256 and Mul128 respectively).

the 3:2 CSA_b Compressor and the MulAdd_b modules. Similarly, the multiplier inside the MulAdd_b module can also be further decomposed into smaller units. In our FFAU, the Mul256 module is implemented following this approach with $b = 128$, while the Mul128 module inside it is again implemented similarly with $b = 64$, thus creating 2 levels of Karatsuba multiplication.

2) Unified Reduction Block: The prime $2^{255} - 19$ in Curve25519 is a pseudo-Mersenne prime which allows fast reduction of each 512-bit product using shifts and additions / subtractions [28]. The prime $2^{448} - 2^{224} - 1$ in Curve448 is a Solinas prime which also allows fast reduction of the 896-bit product using shifts and additions / subtractions [38]. The reduction unit can perform 4 reductions modulo $2^{255} - 19$ at once for Curve25519, simultaneously reducing all 4 of the 512-bit products computed by the 4 multipliers in the FFAU. The adders required for this pseudo-Mersenne prime reduction are re-used for reducing the 896-bit product modulo $2^{448} - 2^{224} - 1$ for Curve448.

3) Modular Inversion: Fermat's Little Theorem [2] has been employed to perform modular inversion at the end of the ECSM computation for both Curve25519 and Curve448. This requires iteratively computing 265 and 462 modular multiplications, respectively, for Curve25519 and Curve448. These two-operand multiplications are executed in the FFAU by computing $A \times C$ as $(A+0) \times (C+0)$. The corresponding control signals and register addresses are also stored in LUTs similar to the LADDER computations.

E. Side-Channel Countermeasures

Side-channel attacks [9] exploit physical leakages such as timing, power consumption and electromagnetic emissions to extract secret information from software and hardware implementations of cryptographic algorithms. Ensuring side-channel resilience is crucial for maintaining the security and integrity of cryptographic operations. Elliptic curves like Curve25519 and Curve448 are designed to have constant-time ECSM computation independent of the secret scalar, thus reducing the risk of timing attacks [42]. This automatically prevents SPA attacks as well. In order to further enhance side-channel resilience by preventing DPA attacks, randomized projective coordinates [41] are used to represent elliptic curve points during the ECSM computation. The randomization process involves transforming a point (X, Y, Z) in projective coordinates to $(\lambda X, \lambda Y, \lambda Z)$, where λ is a pseudo-random non-zero scalar. This helps to obscure the correlation between physical measurements and the internal arithmetic operations.

In our accelerator, the scalar λ is generated using a PRNG and it is multiplied with the input coordinates in the FFAU before the ECSM computations begin. The pseudo-random scalar is 255-bit for Curve25519 and 448-bit for Curve448. This additional security feature has negligible impact on performance as it requires only few additional clock cycles.

The PRNG contains a light-weight Trivium stream cipher [43] which can generate 64 cryptographically secure pseudo-random bits per clock cycle. These 64-bit outputs are concate-nated over multiple cycles to obtain the 255-bit and 448-bit wide pseudo-random scalars. The Trivium core operates with a 288-bit internal state initialized using an 80-bit key and an 80-bit initialization vector (IV).

IV. IMPLEMENTATION RESULTS

We design our accelerator using Verilog HDL and verify its functionality with Cadence Incisive v15.20-s086. We implement the accelerator in a commercial 28nm ASIC technology and obtain post-synthesis simulation results with Cadence Genus v21.18-s082_1 and Cadence Joules v21.18-s002_1. Our synthesized ASIC implementation operates at a maximum frequency of 100 MHz, occupies 1096 kGE (gate equivalent) area and consumes around 69 mW power at 0.9 V supply voltage under typical operating conditions. The FFAU consumes 93% of the area and 97% of total power in the accelerator. The registers consumes 6% of the area and 2% of total power, while the remaining 1% is due to the PRNG and control logic. Area and power breakdown of the FFAU in terms of multipliers (4 × Mul256), adders (8 × Add255 and 2 × Add193), modular reduction and other control logic is shown in Fig. 6. Clearly, the 256-bit multipliers account for majority of the area and power consumption within the FFAU. The critical path of the accelerator also lies in the complex modular arithmetic circuitry (multiplications, additions / subtractions and modular reduction) present inside the FFAU.

For Curve25519, each ECSM computation takes 1,032 and 1,038 clock cycles respectively without and with randomized projective coordinate DPA countermeasures. This corresponds to ECSM latency of 10.32 μs and 10.38 μs respectively. The corresponding energy consumption per ECSM operation are 0.71 μJ and 0.72 μJ respectively. For Curve448, each ECSM computation takes 4,944 and 5,401 clock cycles respectively without and with randomized projective coordinate DPA countermeasures. This corresponds to ECSM latency of 49.44 μs and 54.01 μs respectively. The corresponding energy consumption per ECSM operation are 3.41 μJ and 3.73 μJ respectively. The ECSM computation times achieved by our design for both curves are well within the requirements of latency-critical industrial communication protocols such as IEC 61850 [8]. Also, the DPA countermeasure, that is, randomization of projective coordinates, has almost no impact on overall performance and energy-efficiency.
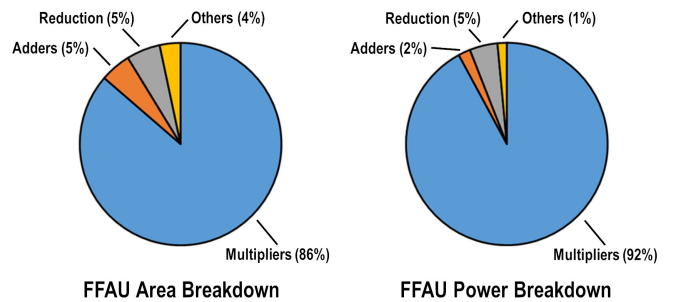

Fig. 6. Area and power breakdown of the FFAU.

TABLE I

COMPARISON WITH STATE-OF-THE-ART HIGH-PERFORMANCE ECSM HARDWARE ACCELERATORS

Design	Implementation Platform	Supported Curve(s)	Voltage (V)	Freq. (MHz)	Area	Power (mW)	ECSM Latency	ECSM Energy	SPA CM [1]	DPA CM [1]
This Work	28nm ASIC [2]	Curve25519	0.9	100	1096 kGE	69	10.38 μs	0.72 μJ	Yes	Yes
		Curve448					54.01 μs	3.73 μJ		
[28]	45nm ASIC [2]	Curve25519	1.1	102	541 kGE	-	52 μs	-	Yes	Yes
[40]	180nm ASIC [3]	Curve25519	1.8	102	377 kGE	627	8.54 ms	5.35 mJ	Yes	-
[31]	Zynq 7020 FPGA	Curve25519	-	60	6,183 Logic Slices + 81 DSPs + 0.5 BRAMs	-	103 μs	-	Yes	Yes
[39]	Zynq 7000 Series FPGA	Curve25519	-	204	5,403 Logic Slices + 128 DSPs + 24 BRAMs	-	14 μs	-	Yes	-
[23]	Zynq 7020 FPGA	Curve448	-	335	1,648 Logic Slices + 35 DSPs + 14 BRAMs	-	1.41 ms	-	Yes	Yes
[30]	Zynq 7020 FPGA	Curve448	-	95	4,424 Logic Slices + 81 DSPs	-	1.4 ms	-	Yes	Yes
[38]	Virtex-7 FPGA	Curve448	-	245	7,666 Logic Slices + 88 DSPs	-	200 μs	-	Yes	Yes
[45]	65nm ASIC [4]	FourQ	1.2	250	1400 kGE	394	10.1 μs	3.98 μJ	Yes	-
[46]	Zynq 7020 FPGA	FourQ	-	190	1,691 Logic Slices + 27 DSPs + 10 BRAMs	-	157 μs	-	Yes	-
[47]	45nm ASIC [2]	NIST P-256	1.1	295	1034 kGE	-	37 μs	-	Yes	-
[48]	65nm ASIC [4]	Any	1.2	105	1.92 mm^2	43	325 μs	13.9 μJ	Yes	-
[49]	65nm ASIC [3]	Any	1.2	105	2490 kGE	178	60 μs	10.7 μJ	Yes	-

For all previous work, the fastest implementations with side-channel (SPA and/or DPA) countermeasures are considered for fair comparison.

[1] CM: Countermeasures   [2] post-synthesis simulation results   [3] post-layout simulation results   [4] post-silicon measurement results

Table I compares our design with previous work on high-performance ECSM hardware accelerators implemented in FPGA and ASIC. Our proposed accelerator not only supports two curves at different security levels in the same hardware but also incorporates SPA and DPA countermeasures. Our design achieves better performance and lower energy consumption compared to previous Curve25519 and Curve448 accelerators [23], [28], [30], [31], [38]–[40]. Compared to previous work on ECC accelerators for other curves at 128-bit security level such as FourQ and NIST P-256 [45]–[49], our design achieves better or similar performance and energy consumption while supporting a higher security level curve as well as stronger side-channel countermeasures. Compared to previous work on low-power ASIC implementations of ECC hardware accelerators [20], [50], our design has lower energy consumption but much larger area due to the high performance requirement.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have presented a high-performance unified hardware accelerator for elliptic curve scalar multiplication (ECSM) over NIST standard Montgomery curves Curve25519 and Curve448. We implement an efficient finite field arithmetic unit (FFAU) with four 256-bit 2-level Karatsuba multipliers to enable parallel processing of arithmetic operations. We restructure the sequence of operations in the Montgomery Ladder for faster computation and store the corresponding instructions in lookup tables for efficient control. Our proposed design strategy facilitates the concurrent execution of up to four 255-bit arithmetic operations during Curve25519 ECSM computation. The same circuitry is re-used for the execution of one 448-bit arithmetic operations during Curve448 ECSM computa-

tion. We implement a unified modular reduction block which enables fast reduction using special mathematical properties of the pseudo-Mersenne and Solinas primes in Curve25519 and Curve448 respectively. Our implementation is constant-time by design and the Montgomery Ladder ensures inherent resilience against SPA attacks. Using a Trivium-based PRNG, we also incorporate the randomized projective coordinate countermeasure to prevent DPA attacks with negligible impact on performance. Our ASIC implementation achieves record performance and energy of 10.38 μs / 54.01 μs and 0.72 μJ / 3.73 μJ respectively for Curve25519 / Curve448. This is significantly better than state-of-the-art, which makes our design particularly attractive for latency-critical applications. Our proposed architecture will benefit electronic systems which need to support elliptic curve cryptography at different security levels based on the requirements of the target applications and can easily switch between Curve25519 and Curve448 to achieve security-versus-efficiency trade-offs.

As future work, our proposed hardware architecture can be extended to incorporate additional side-channel countermeasures. With minor modifications to its controller, the design can also be extended to support Montgomery Ladder ECSM computation with other curves, underscoring its versatility.

REFERENCES

[1] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners.* Springer Science & Business Media, 2009.

[2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 2018.

[3] S. L. Keoh *et al.*, "Securing the Internet of Things: A Standardization Perspective," *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 265–275, May 2014.

[4] International Electrotechnical Commission, "IEC 61850: Communication Networks and systems for Power Utility Automation," 2024.

[5] R. Mackiewicz, "Overview of IEC 61850 and Benefits," in *IEEE Power Engineering Society General Meeting*, 2006.

[6] International Electrotechnical Commission, "IEC 62351: Power Systems Management and Associated Information Exchange - Data and Communications Security," 2024.

[7] S. M. S. Hussain *et al.*, "A Review of IEC 62351 Security Mechanisms for IEC 61850 Message Exchanges," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5643–5654, 2020.

[8] F. Hohlbaum *et al.*, "Cyber Security Practical Considerations for Implementing IEC 62351," in *PAC World Conference*, 2010.

[9] D. D. Hwang *et al.*, "Securing Embedded Systems," *IEEE Security & Privacy*, vol. 4, no. 2, pp. 40–49, Mar. 2006.

[10] U. Banerjee, "Efficient Algorithms, Protocols and Hardware Architectures for Next-Generation Cryptography in Embedded Systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2021.

[11] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[12] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography.* Springer Science & Business Media, 2006.

[13] L. Chen, D. Moody, A. Regenscheid, A. Robinson, and K. Randall, "Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters," NIST SP 800-186, Feb. 2023.

[14] A. Langley, M. Hamburg, and S. Turner, "Elliptic Curves for Security," IETF RFC 7748, 2016.

[15] D. J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records," in *International Conference on Theory and Practice in Public-Key Cryptography (PKC)*, 2006, pp. 207–228.

[16] M. Hamburg, "Ed448-Goldilocks, A New Elliptic Curve," IACR Cryptology ePrint Archive, 2015.

[17] A. A. Giron, R. Custodio, and F. Rodriguez-Henriquez, "Post-Quantum Hybrid Key Exchange: A Systematic Mapping Study," *Journal of Cryptographic Engineering*, vol. 13, no. 1, pp. 71–88, 2023.

[18] P. Sasdrich and T. Guneysu, "Efficient Elliptic-Curve Cryptography using Curve25519 on Reconfigurable Devices," in *International Symposium on Reconfigurable Computing (ARC)*, 2014, pp. 25–36.

[19] ——, "Implementing Curve25519 for Side-Channel-Protected Elliptic Curve Cryptography," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 9, no. 1, pp. 1–15, 2015.

[20] M. Hutter, J. Schilling, P. Schwabe, and W. Wieser, "NaCl's crypto_box in Hardware," in *IACR International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2015, pp. 81–101.

[21] P. Sasdrich and T. Guneysu, "Closing the Gap in RFC 7748: Implementing Curve448 in Hardware," Cryptology ePrint Archive, 2016.

[22] P. Koppermann, F. De Santis, J. Heyszl, and G. Sigl, "X25519 Hardware Implementation for Low-Latency Applications," in *Euromicro Conference on Digital System Design (DSD)*, 2016, pp. 99–106.

[23] P. Sasdrich and T. Guneysu, "Cryptography for Next Generation TLS: Implementing the RFC 7748 Elliptic Curve448 Cryptosystem in Hardware," in *Design Automation Conference (DAC)*, 2017, pp. 1–6.

[24] P. Koppermann, F. De Santis, J. Heyszl, and G. Sigl, "Low-Latency X25519 Hardware Implementation: Breaking the 100 Microseconds Barrier," *Microprocessors and Microsystems*, vol. 52, pp. 491–497, 2017.

[25] P. Sasdrich and T. Guneysu, "Exploring RFC 7748 for Hardware Implementation: Curve25519 and Curve448 with Side-Channel Protection," *Journal of Hardware and Systems Security*, vol. 2, pp. 297–313, 2018.

[26] F. Turan and I. Verbauwhede, "Compact and Flexible FPGA Implementation of Ed25519 and X25519," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 3, pp. 1–21, 2019.

[27] M. A. Mehrabi and C. Doche, "Low-Cost, Low-Power FPGA Implementation of ED25519 and CURVE25519 Point Multiplication," *Information*, vol. 10, no. 9, p. 285, 2019.

[28] R. Salarifard and S. Bayat-Sarmadi, "An Efficient Low-Latency Point-Multiplication over Curve25519," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 10, pp. 3854–3862, 2019.

[29] Y. A. Shah, K. Javeed, M. I. Shehzad, and S. Azmat, "LUT-Based High-Speed Point Multiplier for Goldilocks-Curve448," *IET Computers & Digital Techniques*, vol. 14, no. 4, pp. 149–157, 2020.

[30] M. B. Niasar *et al.*, "Optimized Architectures for Elliptic Curve Cryptography over Curve448," Cryptology ePrint Archive, 2020.

[31] ——, "Fast, Small, and Area-Time Efficient Architectures for Key-Exchange on Curve25519," in *IEEE Symposium on Computer Arithmetic (ARITH)*, 2020, pp. 72–79.

[32] H.-J. Yang and K.-W. Shin, "A Hardware Implementation of Point Scalar Multiplication on Edwards25519 Curve," in *International Conference on Electronics, Information, and Communication (ICEIC)*, 2021, pp. 1–3.

[33] M. Bisheh-Niasar *et al.*, "Cryptographic Accelerators for Digital Signature based on Ed25519," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 7, pp. 1297–1305, 2021.

[34] B. Yu *et al.*, "High-Performance Hardware Architecture Design and Implementation of Ed25519 Algorithm," *Journal of Electronics and Information Technology*, vol. 43, no. 7, pp. 1821–1827, 2021.

[35] M. Bisheh-Niasar *et al.*, "Area-Time Efficient Hardware Architecture for Signature based on Ed448," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 6, pp. 2942–2946, 2021.

[36] S. Mondal and S. Patkar, "Hardware-Software Hybrid Implementation of Non-Deterministic ECC over Curve-25519 for Resource Constrained Devices," in *Asian Conference on Innovation in Technology (ASIANCON)*, 2021, pp. 1–8.

[37] B. Kieu-Do-Nguyen, C. Pham-Quoc, N.-T. Tran, C.-K. Pham, and T.-T. Hoang, "Low-Cost Area-Efficient FPGA-Based Multi-Functional ECDSA/EdDSA," *Cryptography*, vol. 6, no. 2, p. 25, 2022.

[38] A. M. Awaludin, J. Park, R. W. Wardhani, and H. Kim, "A High-Performance ECC Processor over Curve448 based on a Novel Variant of the Karatsuba Formula for Asymmetric Digit Multiplier," *IEEE Access*, vol. 10, pp. 67 470–67 481, 2022.

[39] G. Wu, Q. He, J. Jiang, Z. Zhang, X. Long, Y. Zhao, and Y. Zou, "A High-Performance Hardware Architecture for ECC Point Multiplication over Curve25519," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2022, pp. 1–9.

[40] B. K. Do-Nguyen, C. Pham-Quoc, N.-T. Tran, C.-K. Pham, and T.-T. Hoang, "Multi-Functional Resource-Constrained Elliptic Curve Cryptographic Processor," *IEEE Access*, vol. 11, pp. 4879–4894, 2023.

[41] J. Fan *et al.*, "State-of-the-Art of Secure ECC Implementations: A Survey on Known Side-Channel Attacks and Countermeasures," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, Jun. 2010, pp. 76–87.

[42] M. Joye and S.-M. Yen, "The Montgomery Powering Ladder," in *IACR International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2002, pp. 291–302.

[43] C. De Canniere and B. Preneel, "TRIVIUM Specifications," eSTREAM, ECRYPT Stream Cipher Project, 2006.

[44] A. A. Karatsuba and Y. P. Ofman, "Multiplication of Many-Digital Numbers by Automatic Computers," *Doklady Akademii Nauk*, vol. 145, no. 2, pp. 293–294, 1962.

[45] H. Awano and M. Ikeda, "FourQ on ASIC: Breaking Speed Records for Elliptic Curve Scalar Multiplication," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1733–1738.

[46] K. Jarvinen *et al.*, "FourQ on FPGA: New Hardware Speed Records for Elliptic Curve Cryptography over Large Prime Characteristic Fields," in *International Conference on Cryptographic Hardware and Embedded Systems (CHES)*, 2016, pp. 517–537.

[47] M. Knezevic *et al.*, "Low-Latency ECDSA Signature Verification – A Road Toward Safer Traffic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 11, pp. 3257–3267, 2016.

[48] M. Tamura and M. Ikeda, "1.68 μJ/Signature-Generation 256-bit ECDSA over GF(p) Signature Generator for IoT Devices," in *2016 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, 2016, pp. 341–344.

[49] ——, "Montgomery Multiplier Design for ECDSA Signature Generation Processor," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 99, no. 12, pp. 2444–2452, 2016.

[50] U. Banerjee, A. Wright, C. Juvekar, M. Waller, Arvind, and A. P. Chandrakasan, "An Energy-Efficient Reconfigurable DTLS Cryptographic Engine for Securing Internet-of-Things Applications," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 8, pp. 2339–2352, Aug. 2019.