Secure-by-Design Methodology using Meet-in-the-Middle Design Flow for Hardware Implementations of ECC-based Passive RFID Tags

Manh-Hiep Dao*†, Vincent Beroulle*, Yann Kieffer*, Xuan-Tu Tran†
*Univ. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France
†VNU Information Technology Institute, Vietnam National University, Hanoi, Vietnam
Corresponding author's email: tutx@vnu.edu.vn

***Abstract*—With the rapid development of needs concerning the secured passive Radio Frequency IDentification (RFID) tag, several works propose the implementation of authentication protocols based on Elliptic Curve Cryptography (ECC). But, there are no systematical approaches that allow considering both the compatibility of security and the implementation cost as part of the requirements and limitations of passive RFID tags. Therefore, the problem of balancing the implementation cost and the security requirements for passive RFID tags is still an open question. In this paper, we present a part of a Security-by-Design methodology that targets passive RFID tags using ECC primitives.**

***Index Terms*—Passive RFID, Side-Channel Attack, Design Methodology, Security by Design, ECC, Meet-in-the-Middle.**

I. Introduction

Passive RFID tags are portable devices utilizing radio frequency to authenticate the identity of the objects. By communicating via a wireless channel, these devices face a variety of vulnerabilities such as wireless and hardware attacks. The wireless threats try to illegally access the system to steal or modify the data communicated via the wireless channel. Hardware attacks, such as Side-Channel Attacks (SCA) and Fault Attacks (FA), exploit the weaknesses of design to reveal the secret key. In order to protect the secret information contained in the passive RFID tag, the device tends to implement a secured authentication protocol using cryptography primitives.

Among the cryptography primitives, the asymmetric encryption algorithms, which are based on the Discrete Logarithm Problem (DLP), are recommended to replace the symmetric ones to avoid the key distribution issue. In the asymmetric family, Elliptic Curve Cryptography (ECC) is more attractive due to the advance of shortened key length. However, compared to other cryptography primitives such as symmetric encryption algorithm (AES [1], PRESENT [2]) or hash function, ECC as well as asymmetric cryptography methods are much more complicated. That leads to higher implementation costs such as physical area, power consumption, and latency. Meanwhile, the design of passive RFID tags is very challenging because of the constraints of implementation. Therefore, looking for a compatible design methodology that balances the protection of security and identity data privacy with minimizing the implementation cost of passive RFID tags is still an open question for researchers.

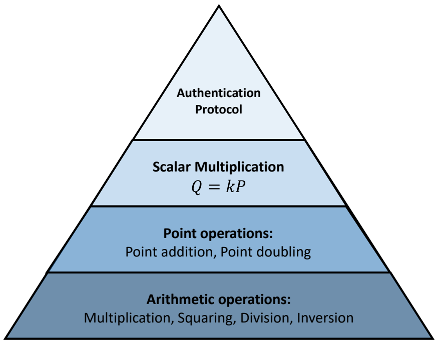

Fig. 1. Concept of Authentication Protocol using ECC primitives.

In the literature, the secured authentication protocols that use cryptography primitives, for passive RFID tags are implemented based on the concept including four primary abstraction levels, as seen in Fig. 1. There are already several design flows that allow taking security into account. The most popular approach used for designing ECC primitives is the top-down design methodology, which is applied in several works [3]–[8] in literature. However, because they lack information on the systematical architecture, these implementations can not prove the compatibility of their design with the design constraints of passive RFID tags.

In addition, there are some works [9], [10] that apply the Bottom-Up Design Methodology. One of the disadvantages of the bottom-up design methodology is that it is time-consuming. When the combined system is too complicated, the simulation and verification become much more complex. Because of this issue, this design methodology is not compatible with designing ECC primitives. In both design methodologies mentioned above, they misconstrue the impact of security as an additional feature of the design. Therefore, it is difficult to obtain an ECC-based authentication protocol balancing implementation cost and security requirements.

In order to solve the mentioned problem, we propose to use a Security-by-Design methodology based on the classical Meet-in-the-Middle approach for designers. A combination of the recommended design methodology with our proposal

ISBN: ISBNFILL

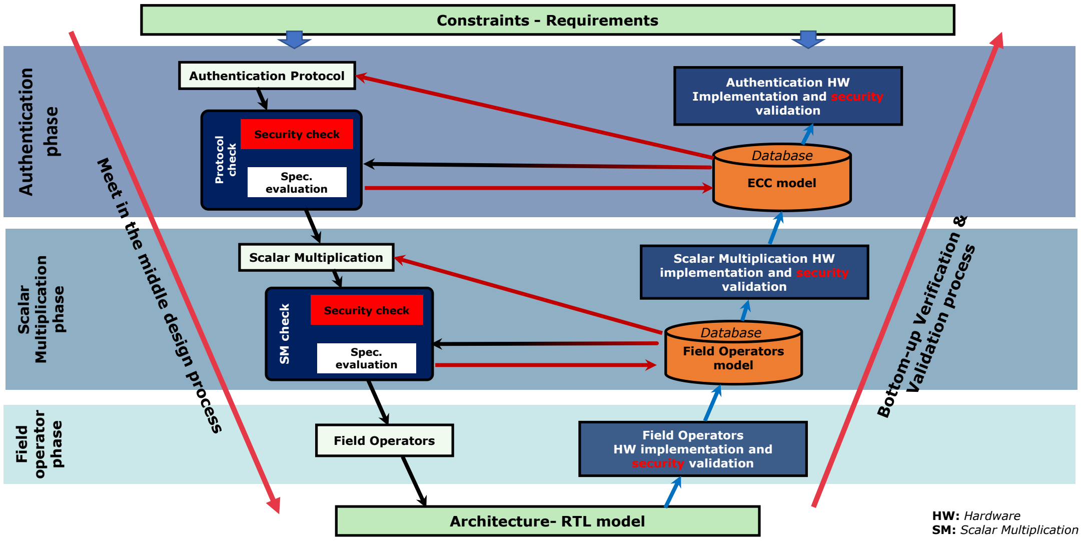
Constraints - Requirements
Authentication phase
Authentication Protocol
Protocol check
Security check
Spec. evaluation
Authentication HW Implementation and security validation
Database
ECC model
Meet in the middle design process
Scalar Multiplication phase
Scalar Multiplication
SM check
Security check
Spec. evaluation
Scalar Multiplication HW implementation and security validation
Database
Field Operators model
Bottom-up Verification & Validation process
Field operator phase
Field Operators
Field Operators HW implementation and security validation
Architecture- RTL model
HW: Hardware
SM: Scalar Multiplication

Fig. 2. Proposed Security-by-Design Methodology using Meet-in-the-Middle.

helps designers to consider both hardware security issues and the implementation cost of authentication protocol using ECC primitive. Consequently, the final authentication protocol based on ECC primitives balances both the security requirements and the design constraints of passive RFID tags. Specifically, our contributions in this paper are:

- A proposed Security-by-Design methodology based on the classical Meet-in-the-Middle approach.
- An estimation of implementation cost and SCA vulnerabilities assessment method for the ECC block based on the knowledge of field operator primitives in the literature.

The organization of this paper is as follows. Section II presents our proposal. Section III indicates an example of the proposed design methodology. In the last section, a conclusion gives a summary of the work and the remaining tasks in the future.

II. Security by Design Methodology

In this section, we present our proposed Security-by-Design Methodology using Meet-in-the-Middle, as described in Fig. 2. This strategy combines the Meet-in-the-Middle Design and the Bottom-Up Evaluation and Validation process. At the beginning of the design process, designers determine the specifications of the target system in terms of implementation cost and security requirements by analyzing choices with the knowledge of the sub-blocks. The final result of this design process is finding a configuration and architecture for the ECC primitive with a compatible authentication protocol.

In the following, the Bottom-Up Evaluation and Validation process is carried out after implementing and assembling the sub-blocks into the system. The aim of this process is to assess the security and validate the design by implementing them on hardware.

A. *Meet-in-the-Middle Design Process*

This process comprises three phases: Authentication, Scalar Multiplication, and Field Operators Design phase, as described in Fig. 2. Due to the page limitation of this paper, we only discuss in more detail the last two phases of the Meet-in-the-Middle Design Process.

1) Authentication Phase: In the Authentication Phase, according to the knowledge of the ECC primitive blocks, which are proposed in the literature, designers consider both the security and implementation cost of various protocols. After choosing a compatible authentication protocol, designers would know the design constraints and security requirements of the ECC primitive at the beginning of the second phase.

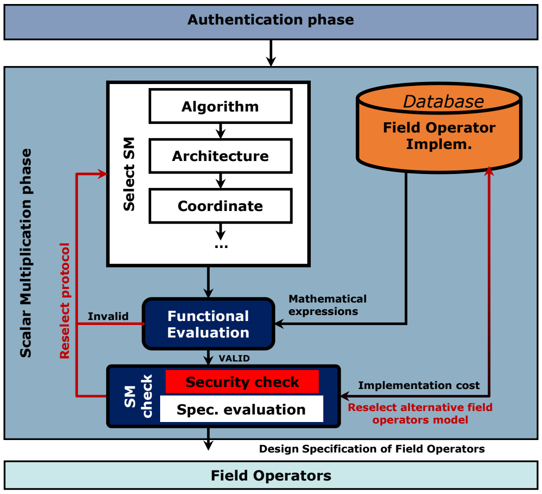

Fig. 3. Scalar Multiplication Design Phase.

2) Scalar Multiplication Phase: In this phase, firstly, based on the determined specification of the ECC primitive, a process of choosing a field, algorithm, architecture, and projective coordinates system would be done, as depicted in Fig. 3.

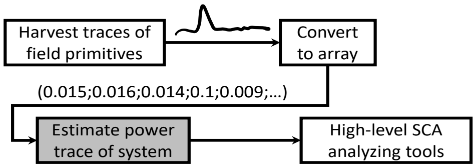
Harvest traces of field primitives
Convert to array
(0.015;0.016;0.014;0.1;0.009;...)
Estimate power trace of system
High-level SCA analyzing tools

Fig. 4. Process of estimating power trace of the ECC primitive.

In the first sub-step, behavioral and abstraction models are extracted from the previous work in the literature. Behavioral models express the mathematical formulas of the field operators, for example, field multiplier, inversion, and squaring. They are used by designers to verify the functionalities of Scalar Multiplication. The abstraction model of field operators indicates the configuration, implementation cost, and power traces with corresponding data. The designers use these abstraction models for evaluating the security and estimating the implementation cost in the Evaluation and Estimation step.

a) Implementation cost estimation of the ECC block: The parameters that we can estimate are the area (A_{total}), power consumption (P_{total}), and maximum duration (T_{total}) of the ECC design. By knowing the detailed configuration of the system, designers could know how many field operators are needed to carry out scalar multiplication. In the following, we note N_i the number of required field operators with the index of $\{1; 2; 3\}$ being field multiplier, field square, and field inverter, respectively. We also note L_{key} the number of bits of the key.

Depending on the architecture of the ECC system, designers know about the layout of sub-blocks and the interconnection of the target system. Therefore, we can estimate the area of the ECC block by using Eq. 1. We note n_i the number of sub-blocks implementing field operators, A_i, as the area of each sub-block carrying out the field operator i.

$$A_{total} = \sum_{i=1}^{3} n_i \cdot A_i \tag{1}$$

Besides, the maximum duration and power consumption of the ECC system are estimated via Eq.2 if the field operators compute in parallel. We note (P_i, T_i) the power consumption and delay of each sub-block carrying out the field operator, respectively i.

$$\begin{aligned} P_{total} &= \sum_{i=1}^{3} n_i \cdot P_i \\ T_{total} &= L_{key} \cdot \max\{(\frac{N_i}{n_i} \cdot T_i) : i = (1, 2, 3)\} \end{aligned} \tag{2}$$

If these sub-blocks work sequentially, the power and maximum duration of the ECC system are estimated via Eq.3.

$$\begin{aligned} P_{total} &= \max\{P_i : i = (1, 2, 3)\} \\ T_{total} &= L_{key} \cdot \sum_{i=1}^{3} \frac{N_i}{n_i} \cdot T_i \end{aligned} \tag{3}$$

b) Power traces estimation: In more detail, our proposal shows an approach to assess approximated power traces of scalar multiplication by the reference traces of primitive operators. At the beginning of this step, we collect the power traces of the field operators, as described in Fig.4 together with corresponding data. These traces could be harvested from the silicon device or estimated by using the power simulator tools. After collecting the traces, they are converted into arrays $P_i = \{p_{i0}, p_1, \cdots, p_{ik}\}$. We note p_{ij} the simultaneous power at the moment $t = j$ of operator i, and k is the length of the trace. In the next step, these arrays are used for estimating the trace of the ECC system.

Depending on the architecture of the ECC primitives, these arrays could be concatenated or accumulated to form the power trace of the system. In the case, two operators, which have power traces P_1, P_2, compute in parallel, the total power trace is the accumulation of P_1 and P_2 as Eq.4:

$$P_{total} = \sum_{i=0}^{k} (P_{1i} + P_{2i}) \tag{4}$$

On the opposite, if two operators work sequentially, the total power trace is the concatenation of two sub-traces as Eq.5. Consequently, the length of the total power trace is longer.

$$P_{total} = \{P_1|P2\} = \{p_{10}, p_{11}, \cdots, p_{1k}, p_{20}, p_{21}, \cdots, p_{2k}\} \tag{5}$$

At the end of this step, the power trace of the system will be assessed by the high-level pre-silicon SCA evaluation tools such as CASCADE [11], or ChipWhisperer [12].

3) Field Operators Phase: In the last phase of the top-down design process, there is a specification of field operators such as field multiplier, square, and perhaps divider as the result of the scalar multiplication phase. The final result of this phase is implementing a full architecture of field operators before interconnecting them to obtain the system of the ECC primitive design. At the end of this phase, designers implement the chosen field operators in the RTL (Register-Transfer Level) model, and then, begin the bottom-up evaluation and validation process.

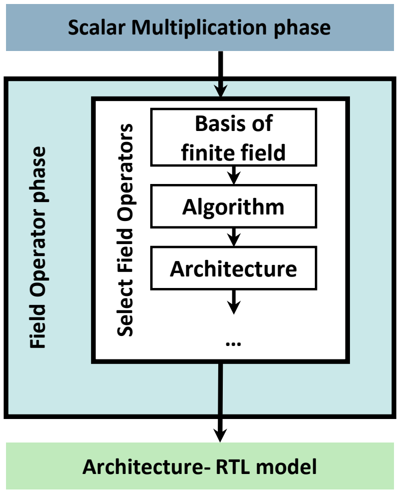

Fig. 5. Field Operators Design Phase.

B. Bottom-up Verification and Validation Process

The inputs of the bottom-up evaluation process are the final RTL model of the full architecture of the ECC-based authentication protocol with the reference models of primitive operators. In each bottom-up evaluation phase, there are two steps: security evaluation and estimation of the cost of input. There are 3 main evaluation phases as discussed below.

1) Field Operator Evaluation phase: The first evaluation phase is Field Operator, as illustrated in Fig. 6. After implementing the RTL model of field operators, by using the EDA (Electronic Design Automation) synthesis tools, designers verify their functionalities. In addition, both estimating the implementation cost and collecting the real power traces of field operators are also carried out on hardware platforms.

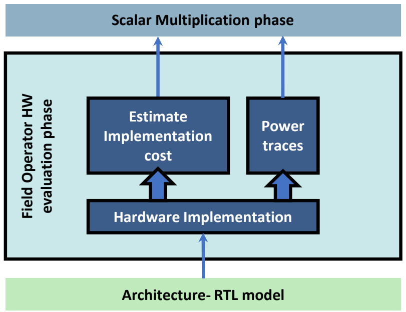
Scalar Multiplication phase
Field Operator HW evaluation phase
Estimate Implementation cost
Power traces
Hardware Implementation
Architecture- RTL model

Fig. 6. Field Operators Evaluation Phase.

2) Scalar Multiplication Evaluation Phase: After evaluating and validating the field operators, these sub-blocks are assembled into the ECC system carrying out the scalar multiplication. The input of this evaluation phase is the hardware implementation of the ECC system. At the beginning of the Scalar Multiplication Evaluation Phase, as demonstrated in Fig.7, the power traces of scalar multiplication are provided to the post-silicon evaluation tool for SCA assessment with the corresponding input data. If the evaluation is failed, designers have to go back to re-select the algorithm of Scalar Multiplication or countermeasures for ECC.

After the successful SCA evaluation of hardware, designers also measure the implementation costs of the ECC primitives. If the implemented ECC primitive is overpriced compared to the reference model, designers go back to the beginning and choose another architecture and algorithm for the Scalar Multiplication block. Conversely, they update the new optimized ECC primitives on the database and embed the ECC primitive to the chosen authentication protocol before continuing to implement the final design of the ECC-based authentication protocol.

III. Example of Meet-in-the-Middle Design Process

In this section, an example of our proposed Secure-by-Design Methodology is presented. Specifically, the process of selecting a design for an ECC-based authentication protocol implementation with design constraints and security requirements is described. The final design specification satisfies all the design constraints and security requirements of the passive RFID tag, which uses ASIC as an implementation platform.

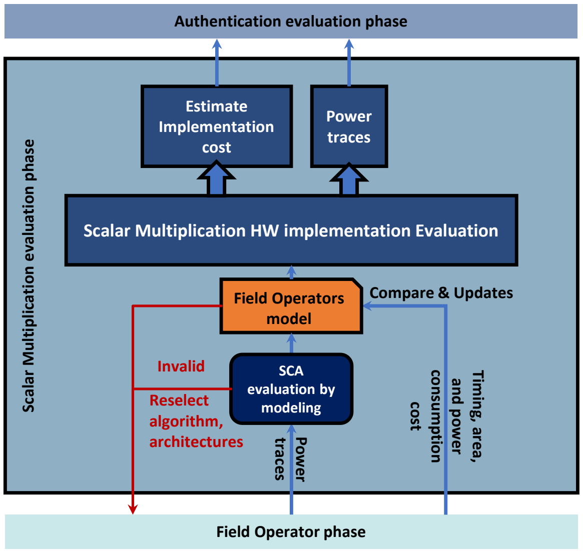
Authentication evaluation phase
Scalar Multiplication evaluation phase
Estimate Implementation cost
Power traces
Scalar Multiplication HW implementation Evaluation
Field Operators model
Compare & Updates
Invalid
SCA evaluation by modeling
Reselect algorithm, architectures
Power traces
Timing, area, and power consumption cost
Field Operator phase

Fig. 7. Scalar Multiplication Evaluation Phase.

The assumption constraints of the ECC-based authentication protocol are listed below:

- Implementation costs:
  - Maximum duration of one tag-server authentication: T_{auth} = 20 ms
  - Maximum available power: P_{max} = 240 μW
  - Implementation area: as low as possible
- Security properties:
  - Secure against Simple SCA and Differential SCA.

The maximum timing of communication refers to the standard ISO/IEC-14443, meanwhile, the maximum power consumption is the peak harvesting at $-3dBm$ incident power by rectifier antenna, according to the proposal in [13].

A. Authentication Phase

We follow the analysis of Gabsi [14] and choose Zhao's protocol [15]. Gabsi showed that this protocol is secure against the Differential SCA. In Zhao's protocol, the tag performs five scalar multiplications. Thus, the maximum duration for one scalar multiplication is $4ms$ $(= 20ms/5)$. Regarding the security requirements, since the protocol is already secure against differential SCA, the scalar multiplication implementation will only have to be secure against Simple SCA.

B. Scalar Multiplication Phase

After determining the target specifications of scalar multiplication, we start the Scalar Multiplication phase, as declared in Fig. 3.

1) Select field: Wenger et al. [20] recommended choosing the Binary Field $GF(2^m)$. In the hardware implementation, Wenger showed that this field requires less than Prime Field

TABLE I
IMPLEMENTATION COST OF DIFFERENT BINARY ELLIPTIC CURVES IN PROJECTIVE COORDINATE.

Curve	**Coordinate System**	**Cost of Point Multiplication**	**Complete**
Binary Generic Curve [16]	Mixed	$6M + 4S$	×
Binary Edward Curve [17]	Mixed	$6M + 4S$	✓
Binary Edward Curve [18]	Affine	$I + 11M + 4S$	✓
Binary Edward Curve [18]	Projective	$16M + S$	✓
Generalized Hessian Curve [19]	Mixed	$9D + 4S$	✓

* I, M, S denote the field inverter, multiplier, and square, respectively.

$GF(p)$ in the context of implementation cost. In addition, we follow standard FIPS 186-4 and choose the 163-bit length of the key for the passive RFID tag.

2) Select algorithm: During this step, we continue to select the configuration of the curve and the algorithm of scalar multiplication. Firstly, there are several curves that can be implemented in the $GF(2^m)$. Table I lists different curves in $GF(2^m)$. Based on Table I, we chose the Binary Edward Curve in the mixed coordinate to implement the point multiplication. This curve has a completeness property that makes it robust against the Simple SCA and also Differential Side-Channel Attacks [21]. In addition, this curve requires $6M+4S$ less than other choices.

TABLE II
EXAMPLE OF THE INITIALIZATION PHASE: FIELD OPERATORS.

	Montgomery multiplier	Montgomery square
Area (kGates)	3.3	0.6
Power (μW)	63	51
Latency (μs)	8.5	8.5

3) Choosing the reference field operators: Before evaluating and estimating the implementation cost of the scalar multiplication block, we choose the reference field operators. As the analysis above, our design only needs field multipliers and field squares. Therefore, we take the implementation cost of these operators which is presented by Deschamps et al. [22]. The synthesis results that we obtained with the ASIC technology NangateOpencore $45\,\mathrm{nm}$ with the maximum clock frequency of $20\,\mathrm{MHz}$ are shown in Table II.

4) Select architecture: In this section, we only consider 3 different architectures with different levels of parallelism of the multipliers and the squares. Arch. a) comprises 3 sub-blocks of field multiplier and 2 sub-blocks of field square. Arch. b) and c) includes 2 and 1 sub-blocks of field multiplier and field square, respectively.

5) Implementation cost estimation: In this step, the implementation costs of the three previous architectures are computed via Equations 1-3. The results of parallel computing in field multiplier and field square are given in Table III.

TABLE III
IMPLEMENTATION COST ESTIMATION FOR SCALAR MULTIPLICATION USING PARALLEL COMPUTING.

Archi.	Area (kGates)	Power (μW)	Latency (ms)
Arch. a)	11.1	291	2.7
Arch. b)	7.8	228	4.13
Arch. c)	3.9	114	8.07

In Table III, we choose Arch.b) for implementing the scalar multiplication as it satisfies the maximum available power and requires an acceptable area. Although its latency is much close to our expectation ($4ms$), it will be improved by optimizing the field operators. The specification of the maximum duration for optimizing field operators in the next phase is determined by Eq. 6.

$$\frac{4}{2 \times 163} \approx 8.18(\mu s) \tag{6}$$

6) Power traces Estimation: Because using 163-bit of key length, there are 163 loop iterations in one scalar multiplication. In each loop, based on the chosen architecture, 2 sub-blocks of field multiplier and 2 sub-blocks of field square are parallel computing. Therefore, based on Eq. 4, the total power traces of scalar multiplication is the accumulation of power traces of sub-blocks.

In the third step, we also estimate the power trace of each field operator by the power estimator tool in 8.5(μs), which is the duration of an operation. Power traces of 1^{st} and 2^{nd} field multiplier sub-blocks are indicated as the orange dot line and green dash line in Fig. 8. Red dot-dash and purple lines in Fig. 8 illustrate the power traces of 1^{st} and 2^{nd} field square sub-blocks. The total power trace is described as the blue line in Fig. 8. This trace is assessed by the high-level pre-silicon SCA tools.

At the end of this design phase, by performing 6 steps of choosing, evaluating, and validating, we find the target configuration of scalar multiplication as below:

- Field: Binary field $GF(2^{163})$
- Curve: Binary Edward Curve in Mixed Coordinate
- Architecture: Using 2 sub-blocks of field multiplier and 2 sub-blocks of field square
- Parallel computing in field multiplier and field square

Besides, we also determine the specification of the field operators. They need to optimize to achieve 8.18 (μs) of latency. Furthermore, in this step, we also estimate the power trace of the system for assessment by pre-silicon SCA evaluation tools.

IV. CONCLUSION AND FUTURE WORK

In this paper, we present a part of the Secure-by-Design methodology using the Meet-in-the-Middle approach that enables the designer to obtain an ECC primitive block that balances the security level and the implementation costs.

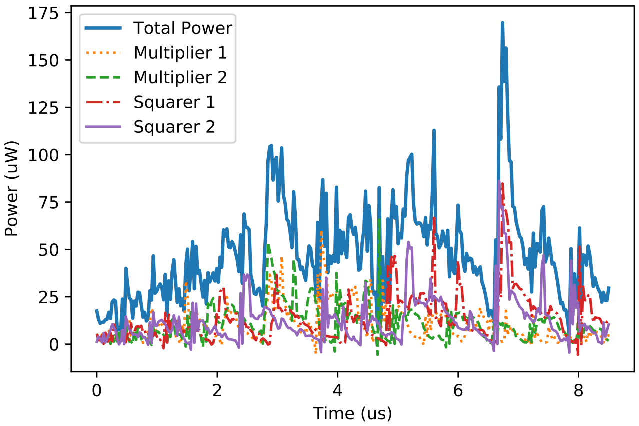

Fig. 8. Example of Power Traces Estimation.

Furthermore, our proposed estimation method enables an approximation of the implementation costs and also the security level of the ECC primitive block based on the knowledge of previous field operator primitives in the literature. In the future, we will include the selection and evaluation of the authentication protocols in the Secure-by-Design Methodology.

Acknowledgment

This work is supported by the French National Research Agency in the framework of the ”Investissements d’avenir” program (ANR-15-IDEX-02) and Vietnam National University, Hanoi under project QG-ECC-RFID, code QG.22.70. The first author is partly funded by Vingroup JSC and supported by the Master, Ph.D. Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, code VINIF.2021.TS.166

References

[1] M.-H. Dao, V.-P. Hoang, V.-L. Dao, and X.-T. Tran, “An energy efficient aes encryption core for hardware security implementation in iot systems,” in *2018 ATC*. IEEE, 2018, pp. 301–304.

[2] A. Bogdanov *et al.*, “Present: An ultra-lightweight block cipher,” in *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings 9*. Springer, 2007, pp. 450–466.

[3] R. Salarifard, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, “A low-latency and low-complexity point-multiplication in ecc,” *IEEE TCAS-I: Regular Papers*, vol. 65, 2018.

[4] T. Shahroodi, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, “Low-latency double point multiplication architecture using differential addition chain over gf(2m),” *IEEE TCAS-I: Regular Papers*, vol. 66, 2019.

[5] R. Salarifard and S. Bayat-Sarmadi, “An efficient low-latency point-multiplication over curve25519,” *IEEE TCAS-I: Regular Papers*, vol. 66, 2019.

[6] P. Choi, M. K. Lee, J. H. Kim, and D. K. Kim, “Low-complexity elliptic curve cryptography processor based on configurable partial modular reduction over nist prime fields,” *IEEE TCAS-II: Express Briefs*, vol. 65, 2018.

[7] S. R. Pillutla and L. Boppana, “Low-complexity bit-serial sequential polynomial basis finite field gf(2m) montgomery multipliers,” *Microprocessors and Microsystems*, vol. 84, 2021.

[8] Q. Shao *et al.*, “Low complexity implementation of unified systolic multipliers for nist pentanomials and trinomials over *GF*(2^m),” *IEEE TCAS-I: Regular Papers*, vol. 65, no. 8, pp. 2455–2465, 2018.

[9] N. Pirotte, J. Vliegen, L. Batina, and N. Mentens, “Balancing elliptic curve coprocessors from bottom to top,” *Microprocessors and Microsystems*, vol. 71, p. 102866, 2019.

[10] J. Lutz and M. Anwarul Hasan, “High performance elliptic curve cryptographic co-processor,” in *Wireless Network Security*. Springer, 2007, pp. 3–42.

[11] D. Sijacic, J. Balasch, B. Yang, S. Ghosh, and I. Verbauwhede, “Towards efficient and automated side channel evaluations at design time,” *Kalpa Publications in Computing*, vol. 7, pp. 16–31, 2018.

[12] C. O’flynn and Z. D. Chen, “Chipwhisperer: An open-source platform for hardware embedded security research,” in *COSADE*. Springer, 2014, pp. 243–260.

[13] P. Xu, D. Flandre, and D. Bol, “Analysis, modeling, and design of a 2.45-ghz rf energy harvester for swipt iot smart sensors,” *IEEE JSSC*, vol. 54, 2019.

[14] S. Gabsi, V. Beroulle, Y. Kieffer, H. M. Dao, Y. Kortli, and B. Hamdi, “Survey: Vulnerability analysis of low-cost ecc-based rfid protocols against wireless and side-channel attacks,” *Sensors*, vol. 21, no. 17, p. 5824, 2021.

[15] Z. Zhao, “A secure rfid authentication protocol for healthcare environments using elliptic curve cryptosystem,” *Journal of medical systems*, vol. 38, pp. 1–7, 2014.

[16] J. López and R. Dahab, “Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation,” in *CHES*. Springer, 1999, pp. 316–327.

[17] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, “Low-resource and fast binary edwards curves cryptography,” in *INDOCRYPT*. Springer, 2015, pp. 347–369.

[18] K. H. Kim, C. O. Lee, and C. Negre, “Binary edwards curves revisited,” in *INDOCRYPT*. Springer, 2014, pp. 393–408.

[19] R. R. Farashahi and M. Joye, “Efficient arithmetic on hessian curves,” in *PKC*. Springer, 2010, pp. 243–260.

[20] E. Wenger and J. Grossschadl, “An 8-bit avr-based elliptic curve cryptographic risc processor for the internet of things,” in *MICRO*. IEEE, 2012, pp. 39–46.

[21] H. M. Edwards, “A normal form for elliptic curves,” *Bulletin of the American Mathematical Society*, vol. 44, 2007.

[22] J. P. Deschamps, G. D. Sutter, and E. Cantó, “Guide to fpga implementation of arithmetic functions,” *Lecture Notes in Electrical Engineering*, vol. 149 LNEE, 2012.