Side-Channel Attack using Order 4 Element against Curve25519 on ATmega328P

Yoshinori Uetake, Akihiro Sanada, Takuya Kusaka, and Yasuyuki Nogami
Graduate School of Natural Science and Technology, Okayama University, Japan
{yoshinori.uetake, akihiro.sanada}@s.okayama-u.ac.jp
{kusaka-t, yasuyuki.nogami}@okayama-u.ac.jp

Léo Weissbart
Grenoble INP-Esisar : Advanced Systems and Networks, France
leo.weissbart@grenoble-inp.org

Sylvain Duquesne
Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France
sylvain.duquesne@univ-rennes1.fr

***Abstract*—With the matter of secure communication between devices, and especially for IoT devices, more and more applications need trustful protocols to communicate using public key cryptography. Elliptic curve cryptography is nowadays a very secure and efficient public key cryptography method. One of the most recent and secure curve is Curve25519 and one of its failure is attack on low-order elements during a Diffie-Hellman key exchange. This document demonstrates that an attack using an order 4 point is possible on an embedded system with a simple power analysis, pointing out every IoT using Curve255119 as a cryptographic method, a potential target to side-channel attacks.**

I. Introduction

Elliptic Curve Cryptography (ECC) has been introduced in 1985 by Neal Koblitz [1] and Victor S. Miller [2] as a new cryptographic method that can concurrence e.g. RSA [3] or DSA [4]. Since then, ECC has become a popular method for cryptography because it can offer the same level of security as other cryptographic method by using a shorter public key and faster computations. This asset makes ECC one of the most explored cryptographic mean in todays security issues.

In 2006, Daniel J. Bernstein proposed the Curve25519 [5] as a new secure elliptic curve. The key exchange protocol X25519 is based on Diffie-Hellman key exchange protocol using Curve25519 and have been designed to be efficient, secure and easy to implement. Curve25519 has been adopted by IETF as one of the next generation curve for the widely used cryptography standard on Internet, TLS [6].

More recently, Daniel Genkin, and Luke Valenta and Yuval Yarom have successfully exploited a failure in X25519 with a software based attack using order 4 elements [7]. Contrary to Flush+Reload method presented in the previous paper, this paper is focus on a power consumption SPA, showing simple power analysis is also successful method to extract the secret information.

To confirm order 4 elements can be a threat to X25519, our study use the existing open source library μNaCl developed in [8] to implement scalar multiplication (SCM) over Curve25519 on Arduino UNO and perform a side-channel attack (SCA) to do a simple power analysis (SPA) and retrieve secret information in X25519 using low-order element.
This research proves Curve25519 possess dangerous elements for cryptographic use, and confirms the possibility of SPA exploiting this failure without memory access such as conventional methods.

After explaining the mathematical fundamentals used in ECC, we will explain the plan of the attack and the setup we used to, finally present our results and the conclusion we can draw from them.

II. Fundamentals

A. Elliptic Curve

Let denote the finite field $\mathbb{F}_p$ defined by it characteristic p, a prime number. And let E be the elliptic curve defined over the prime field $\mathbb{F}_p$ in the simplified Weierstrass form [9].

$$E: \ y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_p \tag{1}$$

We say that a point is on the elliptic curve if it is a rational point. Every rational point including the point at infinity $\mathcal{O}$ forms the additive abelian group $E(\mathbb{F}_p)$.

Elliptic curve cryptography using general type of Elliptic curve is not efficient and is also not secure. In fact, computation of ECA and ECD needs inversion which is a very heavy operation on finite fields with a big prime number. The next section will introduce a particular type of elliptic curve that can do calculation without inversion.

B. Montgomery Curve

The Montgomery curves as defined in [10] is defined as follows:

$$E : By^2 = x^3 + Ax^2 + x \tag{2}$$

Where $A, B \in \mathbb{F}_p$ and where $B(A^2 - 4) \neq 0 \pmod p$.
By moving to projective coordinates, a rational point $P = (x, y)$ on Montgomery curve is represented with coordinates $\overline{P} = (X : Z)$ where $x = X/Z$ for $Z \neq 0$.

- ECA with Montgomery Curve

Let $\overline{P} = (X_i : Z_i)$ and $\overline{Q} = (X_j : Z_j)$ be two different rational points. Let also consider the points represented by $\overline{U} = \overline{P} - \overline{Q}$ with coordinates $\overline{U} = (X_{i-j} : Z_{i-j})$. Their

addition $\overline{R} = \overline{P} + \overline{Q}$ have coordinates $\overline{R} = (X_{i+j} : Z_{i+j})$ and are calculated as follows:

$$\begin{aligned} X_{i+j} &= Z_{i-j}[(X_i - Z_i)(X_j + Z_j) + (X_i + Z_i)(X_j - Z_j)]^2 \\ Z_{i+j} &= X_{i-j}[(X_i - Z_i)(X_j + Z_j) - (X_i + Z_i)(X_j - Z_j)]^2 \end{aligned} \quad (3)$$

Algorithm 1 ECA

Input: $\overline{P} = (X_i, Z_i), \overline{Q} = (X_j, Z_j), \overline{U} = (X_{i-j}, Z_{i-j})$
Output: $\overline{R} = (X_{i+j}, Z_{i+j})$

$t_1, t_2, t_3, t_4, t_5, t_6, t_7$
$t_1 = x_i - z_i$
$t_2 = x_j + z_j$
$t_3 = x_i + z_i$
$t_4 = x_j - z_j$
$t_5 = t_1 \cdot t_2$
$t_6 = t_3 \cdot t_4$
$t_7 = t_5 + t_6$
$t_7 = t_7 \cdot t_7$
$X_{i+j} = Z_{i-j} \cdot t_7$
$t_7 = t_5 - t_6$
$t_7 = t_7 \cdot t_7$
$Z_{i+j} = X_{i-j} \cdot t_7$

- ECD with Montgomery Curve

Let $\overline{P} = (X : Z)$ be a rational point. Consider the point addition $\overline{R} = \overline{P} + \overline{P}$, where $\overline{R} = (X_R : Z_R)$ is defined as follows:

$$\begin{aligned} X_R &= (X + Z)^2 (X - Z)^2 \\ T &= (X + Z)^2 - (X - Z)^2 \\ Z_R &= T[(X - Z)^2 + \tfrac{A+2}{4} \cdot T] \end{aligned} \quad (4)$$

Algorithm 2 ECD

Input: $P(X, Z), \alpha = \frac{A+2}{4}$
Output: $2P(X_2, Z_2)$

t_1, t_2, t_3
$t_1 = X + Z$
$t_1 = t_1 \cdot t_1$
$t_2 = X - Z$
$t_2 = t_2 \cdot t_2$
$X_2 = t_1 \cdot t_2$
$t_1 = t_1 - t_2$
$t_3 = \alpha \cdot t_1$
$t_2 = t_3 + t_2$
$Z_2 = t_1 \cdot t_2$

With the Montgomery curves, ECA and ECD are more efficient and do not rely on inversion operations. However, as a consequence of this representation, $\overline{P} - \overline{Q}$ must be known in order to compute $\overline{P} + \overline{Q}$ and by this fact making the computation of y coordinate in this representation unnecessary.

Elliptic Curve Diffie-Hellman (ECDH) based cryptography relies on scalar multiplication (SCM) over the elliptic curve to generate a session key. During this operation the algorithm can potentially be attacked.

C. Montgomery Ladder (ML)

The Montgomery ladder is an algorithm introduced in [10], and is an efficient method to perform SCM for a rational point $\overline{P}$ and a scalar $s = (s_{n-1}, s_{n-2} \dots s_1, s_0)_2$. The ML is an efficient SCM technique and is calculated as follows:

Algorithm 3 SCM with Montgomery Ladder

Input: $\overline{P}, s = (s_{n-1}, s_{n-2} \dots s_1, s_0)_2$
Output: $\overline{T_1} = [s]\overline{P}$

$\overline{T_1} \leftarrow \mathcal{O}$
$\overline{T_2} \leftarrow \overline{P}$
for $i = n - 1$ to 0 do **do**
    if $s_i = 1$ **then**
        $\overline{T_1} \leftarrow \overline{T_1} + \overline{T_2}$
        $\overline{T_2} \leftarrow 2\overline{T_2}$
    else
        $\overline{T_2} \leftarrow \overline{T_1} + \overline{T_2}$
        $\overline{T_1} \leftarrow 2\overline{T_1}$
    end if
end for
return $\overline{T_1}$

D. Curve25519

The curve Curve25519 is a Montgomery curve introduced by Daniel J. Bernstein [5] in 2006. This curve has received a great interest in modern cryptography and is at this day used in hundreds of applications for its efficiency and rapidity. Curve25519 is defined over prime field $\mathbb{F}_q$ of order $q = 2^{255} - 19$ and its equation is defined as follows:

$$E_{25519} : y^2 = x^3 + 486662x^2 + x \quad (5)$$

Curve25519 has characteristic low-order points, (0, 0) on the affine coordinates is a order 2 point and $(1, \pm\sqrt{486664})$ is a order 4 point over $\mathbb{F}_q$.

E. Side-channel Attack (SCA)

Side-channel attack [11] is the method of analyzing the physical behavior of a cryptographic module to recover secret information.

A cryptographic module is included on an integrated circuit (IC) and is composed of numerous CMOS gates handling secret information. With the cryptographic processing, a current accompanying the switching of the MOS transistor is generated, causing fluctuations in the power supply voltage and electromagnetic radiation. The number of gates changing state depends on plain text, ciphertext, and secret key.

In other words, by observing unintended physical data (or so-called side-channel information) at the time of cryptographic processing, there is a possibility of leaking the secret information. The consumption level of IC is different when switching from low to high and the opposite. SCA takes advantage of this characteristic to analyze either power consumption or electromagnetic field radiations of the IC. Analyzing power consumption to visually examine its trace is called simple power analysis (SPA).

III. Implementation

A. Computational Environment

This section describes the means used during the experiments to perform SCA on Curve25519 with Arduino UNO. As the values of Curve25519 are contained in the prime field of characteristic $q = 2^{255} - 19$, one value should be interpreted as a 256 bits machine word.

Working with a precision arithmetic library is necessary to store these values (the secret key, base points' coordinates, etc.).

The chosen library to work on Curve25519 is the μNaCl library [8]. This library is designed to perform ECC on Curve25519 for AVR microcontroller. This precision arithmetic library does manage addition, subtraction, multiplication and modular operations with 256 bits variables, with very efficient use of memory.

The three algorithms ECA (Alg. 1), ECD (Alg. 2) and ML (Alg. 3) have been implemented as described above, using μNaCl library for multiprecision arithmetic. To perform a single SCM using ML, we need at least 580Bytes of space in RAM using the μNaCl library. The maximum memory consumption is during the ML when performing an ECA. In fact at that moment, there is,

$$18 \cdot 256 \text{ bits} + 4 \cdot 8 \text{ bits} = 4640 \text{ bits} = 580 \text{ Bytes}$$

of memory used. (256 for alpha, 256 for the prime, 256 for the scalar, $2 \cdot 256$ for the point P and $2 \cdot 256$ for point $[s]P$, $4 \cdot 256$ and $4 \cdot 8$ in temporary values of ML function and $7 \cdot 256$ for temporary value in ECA function.)

TABLE I
ARDUINO UNO SPECIFICATIONS

CPU	ATmega328P
Flash	32K bytes
Memory	2K bytes
Language	C and Arduino functions
Compiler	avr-gcc

B. SCA of Order 4 Point

Scenario of Attack: For the sake of the following explanation we will imagine a scenario of attack in which an attacker will fraudulently introduce a point of order 4 in a ECC method. We will choose EC ElGamal cryptography (as introduced in [1]), where the attacker replace the ciphertext $(\overline{C_1}, \overline{C_2})$ with $(\overline{P}, \overline{C_2})$ during decrypting phase of the algorithm. As a consequence, calculation of SCM will append involving point $\overline{P}$ of order 4 and endangered the secret key involved.

In this paper, order 4 point is used for SCA to estimate secret key. As it is known, there is a subgroup of order 4 since Montgomery curves' group order is divisible by 4 [10]. This doesn't mean that the curve always has an order 4 point. There is not order 4 point when the subgroup is isomorphic with $\mathbb{Z}_2 \times \mathbb{Z}_2$ however Curve25519 has it. Then, let this order 4 point $\overline{P}$ define as a chosen-ciphertext. $\overline{P} = (X = \beta : Z = \beta), \beta \neq 0$ when use projective coordinates. The result of doubling $\overline{P}$ is order 2 point $\overline{2P}$, represented by $\overline{2P} = (X = 0 : Z \neq 0)$. $\overline{3P} = \overline{P} + \overline{2P}$ is same with $\overline{P}$ on the projective coordinates because the y-coordinate of $\overline{3P}$ is just y-axis opposition of $\overline{P}$ and there is no need to consider about y-coordinate. In addition, point at infinity $\mathcal{O}$ is defined $\mathcal{O} = (X \neq 0 : Z = 0)$. The relations between these points are as follows:

$$\mathcal{O} + \overline{P} = \overline{P},\ \mathcal{O} + \overline{2P} = \overline{2P}$$
$$\overline{P} + \overline{P} = \overline{2P},\ \overline{P} + \overline{2P} = \overline{P},\ \overline{2P} + \overline{2P} = \mathcal{O}$$

In other words, during SCM with ML, the outcome of every operation is within those 3 rational points.

In Fig. 1, each state represents the pair of value $[\overline{T_1}, \overline{T_2}]$ (Alg. 3), and it shows relation of a current state and the next state. The transitions between states are divided into two cases $\mathrm{K_a}$ and $\mathrm{K_b}$ also give information on the key value(whereas the key bit is 1 or 0.) For example, $\mathrm{K_a} : 0$ means the case is $\mathrm{K_a}$ and the key bit is 0. These transitions are defined as follows:

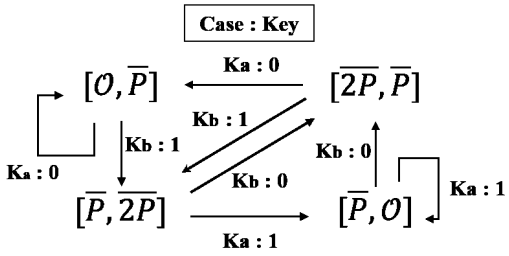


Fig. 1. Flow of SCM

In each case, Eq. (4) becomes

- Case : $\mathrm{K_a}$

$$\begin{aligned} X_R &= (0+\theta)^2 \cdot (0-\theta)^2 = \theta^2 \cdot \theta^2 = \theta \\ T &= (\theta+0)^2 - (\theta-0)^2 = 0 \\ Z_R &= 0 \cdot [(\theta-0)^2 + \tfrac{A+2}{4} \cdot 0] = 0 \end{aligned} \tag{6}$$

- Case : $\mathrm{K_b}$

$$\begin{aligned} X_R &= (\theta+\theta)^2 \cdot (\theta-\theta)^2 = \theta^2 \cdot 0 = 0 \\ T &= (\theta+\theta)^2 - (\theta-\theta)^2 = \theta \\ Z_R &= \theta \cdot [(\theta-\theta)^2 + \tfrac{A+2}{4} \cdot \theta] = \theta \end{aligned} \tag{7}$$

with θ a big number that generates rational point for convenience.

The value 0 is used for the X_R calculation of Eq. (7), on the other hand, Eq. (6) doesn't use. Moreover, Z_R calculation of Eq. (6) has 0 value because T is 0 and Eq. (7) calculates the huge number. For each calculation, we can spot a multiplication being a non-zero multiplication or a zero multiplication depending on the bits of the secret key. Since the power consumption of calculation using 0 is considered to be smaller than others, the secret key can be retrieved using the method described in Fig. 1. This is the method we decided to introduce to recover secret information using order 4 point by SPA.

IV. Experimental Results

The setup for this experiment is: an Arduino UNO flashed with ML algorithm and an oscilloscope (Agilent Technologies DSOS104A). The attack is performed as explained in Section III-B. The base point used to compute SCM is order of 4: $\overline{P}$ (see Apendix.)

This point is intentionally chosen with big coordinates because if the base point has small coordinate, the first loops of ML give too low power consumption differences between zero and non-zero multiplication, and it is not possible to retrieve all the secret key coordinates.

The secret key s (see Apendix) is a 256 bits value initialized randomly.

To make the lecture of the trace easier, a signal is raised with an analog pin during the interesting multiplication(in this paper focus on Z_R calculations of Eq. (6) and Eq. (7)). As the complete SCM is about 7 seconds long, we will focus on the first eight bits calculation, but the same result is visible for all the bits of the secret key. The results of the attack are represented in Fig. 2a.

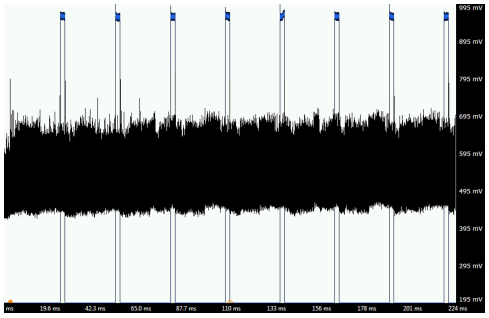
As measurement was noisy, we have run 80 times with the same encryption, and average the traces to obtain the trace in Fig. 2b, making difference between zero and non-zero multiplications more visible for human eyes.

During the marked multiplication, we can see for the two first times, the power consumption is low and on the next loop, we can see a higher power consumption. The two first invocation of multiplication is zero multiplication and so the energy needed to set the final value to zero is low. In the third invocation of the function, we have a big number with another big number multiplication meaning the number of CMOS to switch active is big and also the energy needed is greater than a zero multiplication.

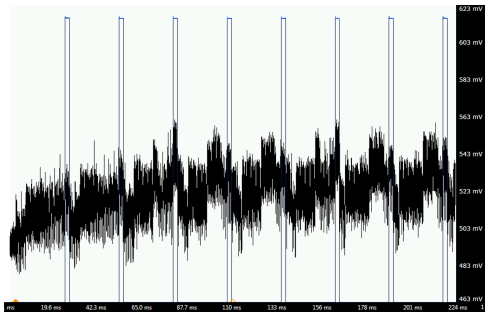
From above results, we can recover the secret key value using Fig. 1. Initially, state is $[\mathcal{O}, \overline{P}]$ and low power consumption (case $\mathrm{K_a}$). Therefore, the key is 0 and next state is $[\mathcal{O}, \overline{P}]$. The second loop is also same pattern. Then, third loop is higher power consumption (case $\mathrm{K_b}$), next state is $[\overline{P}, \overline{2P}]$ with key value 1. We can say 4th key value is 1 in the same way as previous processes. These values are completely same as the secret key s.

In our study, we have shown that attacking Curve25519 with Montgomery implementation using order 4 point attack is possible with an SPA on Arduino UNO. In other words, the most popular board for embedded applications is not secure even with a high secure elliptic curve. Also even if X25519 is highly secure key exchange agreement, it is possible to attack it, (and even with cheap equipment and low-level method) making the method inefficient when using low-order points of Curve25519.

Possible amelioration of the method: To make this attack efficient when used with an SPA, it is possible to train a machine to recognize zero and non-zero multiplication on the power trace. Adding artificial intelligence (AI) would make the attack more exploitable and trustful than the human eye analysis.



(a) Single measure



(b) Averaging over 80 runs

Fig. 2. Power trace of ML implementation on Curve25519 using order 4 point as base point.

This study did not explore this possibility, but using AI for SPA seems to be an efficient improvement for this attack.

V. Conclusion

In this work, we demonstrate a SCA against ML implementation of Curve25519 using the branchless formula of ECA and ECD. This attack has focused on Curve25519 vulnerability for order 4 elements. We find out that recover the secret key used in X25519 is possible when implemented on Arduino UNO by analyzing the power trace of ML algorithm.

VI. Acknowledgment

This work is partially supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of Ministry of Internal Affairs and Communications, Japan, and by French projects ANR-16-CE39-0012 "SafeTLS" and ANR-11-LABX-0020-01 "Centre Henri Lebesgue."

APENDIX

$$s = 25767245217854073605550538472794843835681661831502188636038000083796091605992$$

$$P = (105339071702488710654371683573220568294491524162238083684354190540340853667967, 105339071702488710654371683573220568294491524162238083684354190540340853667967)$$

REFERENCES

[1] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[2] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[3] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[4] David W Kravitz. Digital signature algorithm, July 27 1993. US Patent 5,231,668.

[5] Daniel J Bernstein. Curve25519: new Diffie-Hellman speed records. In *International Workshop on Public Key Cryptography*, pages 207–228. Springer, 2006.

[6] Kenny P. Formal request from TLS WG to CFRG for new elliptic curves, 2014.

[7] Daniel Genkin, Luke Valenta, and Yuval Yarom. May the fourth be with you: A microarchitectural side channel attack on several real-world applications of Curve25519. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 845–858, New York, NY, USA, 2017. ACM.

[8] Michael Düll, Björn Haase, Gesine Hinterwälder, Michael Hutter, Christof Paar, Ana Helena Sánchez, and Peter Schwabe. High-speed curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. *Designs, Codes and Cryptography*, 77(2-3):493–514, 2015.

[9] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic curves in cryptography*, volume 265. Cambridge university press, 1999.

[10] Peter L Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987.

[11] Eric Peeters. Advanced DPA theory and practice, 2013.