Hardware Components for Post-Quantum Elliptic Curves Cryptography

Rodrigue Elias[1], Valerii Hlukhov[2], Mohammed Rahma[2], Ivan Zholubak[2]

1. School of Engineering at Lebanese International University, LEBANON, Beirut, 2nd floor - Block F- Beirut Campus, P.O. Box: 146404, Mazraa, e-mail: rodrigue.elias@liu.edu.lb

2. Computers Dept., Lviv Polytechnic National University, UKRAINE, Lviv, 12 Bandera Str., e-mail: glukhov@polynet.lviv.ua

Abstract: **Investigations in the sphere of quantum calculations form up new challenges in the public key cryptography. Currently known public key crypto algorithms may be compromised with the implementation of quantum computers. The workgroups of ETSI and NIST determined the promising trends, within the framework of which there could be obtained acceptable solutions and one of the trends is the use of algorithms that process points of supersingular elliptic curves. Hardware implementations of components for processing points of elliptic curves are well known. The purpose of this publication is to investigate how they can be used to process points of supersingular elliptic curves.**

Keywords: **hardware components, post-quantum, elliptic curves, cryptography.**

I. Introduction

The advent of large-scale quantum computing offers great promise to science and society, but brings with it a significant threat to global information infrastructure. Public-key cryptography - widely used on the internet today – relies upon mathematical problems that are believed to be difficult to solve given the computational power available now and in the medium term.

However, popular cryptographic schemes based on these hard problems – including Elliptic Curve Cryptography – will be easily broken by a quantum computer. This will rapidly accelerate the obsolescence of currently deployed security systems and will have dramatic impacts on any industry where information needs to be kept secure.

Quantum-safe cryptography refers to efforts to identify algorithms that are resistant to attacks by both classical and quantum computers, to keep information assets secure even after a large-scale quantum computer has been built [1].

Supersingular isogeny Diffie–Hellman key exchange (SIDH) is a post-quantum cryptographic algorithm used to establish a secret key between two parties over an otherwise insecure communications channel. It is analogous to the Diffie–Hellman key exchange, but is designed to resist cryptanalytic attack by an adversary in possession of a quantum computer.

II. The Supersingular Isogeny Diffie-Hellman Background

The supersingular isogeny Diffie-Hellman (SIDH) method works with the set of supersingular elliptic curves E over Galois Field $GF(p^2)$. An isogeny of an elliptic curve E is a rational map from E to another elliptic curve E' which is also a group homomorphism. Provided the isogenies are separable, they are determined by the points inside their kernel up to isomorphisms of E'.

The SIDH method works with a prime of the form $p=(w_A)^{e_A}(w_B)^{e_B}(f)\pm 1$ where w_A and w_B are small primes and an elliptic curve E defined by the equation: $y^2=x^3+ax+b$. SIDH builds an isogeny map from a single elliptic curve point which is taken as the generator for the isogeny's kernel. This point is chosen to be a random linear combination to two fixed points chosen to be in the kernel of the isogeny.

The j-invariant of an elliptic curve E is a fixed function of a set of isomorphic curves. It is computed from the parameters that define the curve. For an elliptic curve E defined by the equation: $y^2=x^3+ax+b$ the j-invariant of the curve E is $j(E)=1728\frac{4a^3}{4a^3+27b^2}$.

The security of SIDH is closely related to the problem of finding the isogeny mapping between two supersingular elliptic curves with the same number of points. In [3] it was shown that the security of SIDH will be $O(p^{1/4})$ for classical computers and $O(p^{1/6})$ for quantum computers. This suggests that SIDH with a 768-bit prime (p) will have a 128-bit security level.

In 2014, researchers at the University of Waterloo developed a software implementation of SIDH. They ran their partially optimized code on an x86-64 processor running at 2.4 GHz. For a 768-bit modulus they were able to complete the key exchange computations in 200 milliseconds thus demonstrating that the SIDH is computationally practical [4].

In 2016, researchers from Microsoft posted software for the SIDH which runs in constant time (thus protecting against timing attacks) and is the most efficient implementation to date [5].

In 2017, researchers from Florida Atlantic University developed the first FPGA implementations of SIDH for 83-bit and 124-bit quantum security levels [6].

III. Definition of Isogeny

For supersingular elliptic curves there is well known Vélu algorithm [7] for isogeny [8]: Let E1 and E2 be elliptic curves over the field F. The isogeny E1 E2 over F is a non-

constant rational mapping over F, which is also a group homomorphism

$$(x,y)\longrightarrow\left(\frac{f_1(x,y)}{f_2(x,y)},\frac{g_1(x,y)}{g_2(x,y)}\right),$$ where f_1, g_1, f_2, g_2 are polynomials. For example, F = GF(19) ,

elliptic curve E1: $y^2 = x^3 + x + 1$;

elliptic curve E2: $y^2 = x^3 + 4x + 13$

#E1 = #E2 = 21;

$$(x,y)\longrightarrow\left(\frac{x^3-4x^2-8x-8}{x^2-4x-4},\frac{x^3y-6x^2y-5xy-6y}{x^3-6x^2-7x-8}\right).$$

So, to determine the isogeny, it is necessary to perform the operations of addition, multiplication and inverse element calculation in the Galois field GF(p2).

IV. Estimation of the Software Time Complexity of Operations in the Galois Fields

In works [11] and [12], the evaluation of the ability of data protection means to counteract attacks by hackers was carried out. The definition of Galois field in which hackers work hardest was the purpose of the study. It was assumed that hackers use software methods.

One of the computer system hacking methods is the brute-force method [8], in which the general-purpose computer selects all sorts of keys or passwords until one of them fits. The same operations over Galois fields elements are performed both during the execution of the hack program and in the hardware crypto processors. For general purpose computers, it is possible to estimate the time of execution of the main operation, multiplication of the Galois fields elements, for extended fields with different characteristics, but with approximately the same order. The basis for such a check you can take the field GF(2^{999}). The calculations you can make using the Maple 2017 package [9]. The relative times of execution of such number of multiplications with respect to the time of execution of the same number of operations in the binary field GF(2^{999}) are shown Table 1 and in the Fig. 1 where prime $p\approx2^{999}$ is characteristic of prime GF(p).

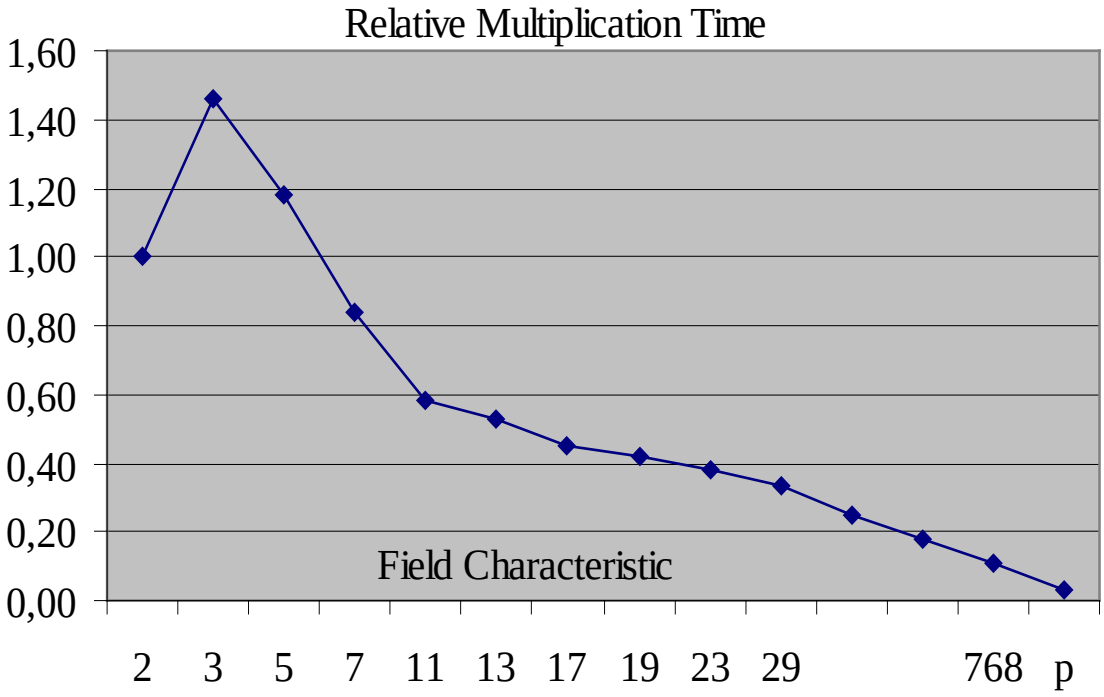

Fig. 1. Relative time complexity of software multiplying in a different fields.

The relative time complexity was determined by the ratio of the multiplication time in the field GF(d^m) to the multiplication time in the field GF(2^{999}).

As can be seen from the Table 1, software multiplication of triple extended field elements has the longest execution time. It provides hardware cryptoprocessors based on such fields additional protection against hacking. Software-implemented operations on simple field elements are executed in the fastest way, that indicates the inappropriateness of cryptographic processors based on such fields. Multiplication in binary fields has one of the highest time complexity, it is third after multiplication complexities in fields with characteristics 3 and 5. Therefore, the following study will focus on binary fields.

Software multiplication in fields with characteristic 768 which is used in [3] is performed for 10 times faster than in binary fields. Therefore, this system can be hacked using classic computers faster than the system that uses binary fields.

TABLE 1. THE RELATIVE TIME COMPLEXITY OF MULTIPLICATION IN EXTENDED FIELD OF DIFFERENT CHARACTERISTICS

Field Characteristic	Relative Time
2	1,00
3	1,46
5	1,18
7	0,84
11	0,59
13	0,53
17	0,45
19	0,42
23	0,38
29	0,33
…	…
768	0,11
p	0,03

V. ESTIMATION OF THE HARDWARE TIME COMPLEXITY OF OPERATIONS IN THE GALOIS FIELDS

Implemented in modern FPGA hardware multipliers for extended Galois field GF (d^m) with approximately the same number of elements $d^m \approx 2^n$ were estimated in [13] in terms of their time complexity to determine the fields in which the multiplier will have the least time complexity. Relative to GF(2^n) results of estimation is shown in Fig. 2.

VI. ESTIMATION OF THE SOFTWARE-HARDWARE TIME COMPLEXITY OF OPERATIONS IN THE GALOIS FIELDS

We will assume that the hardware implementation is used by users of data protection tools, and software is used by hackers. Then we can introduce a generalized time complexity index. We will calculate this indicator as the ratio of software time complexity to hardware time complexity for the same fields. When the value of this indicator is greater, then hackers will have more problems, so data protection will be better.

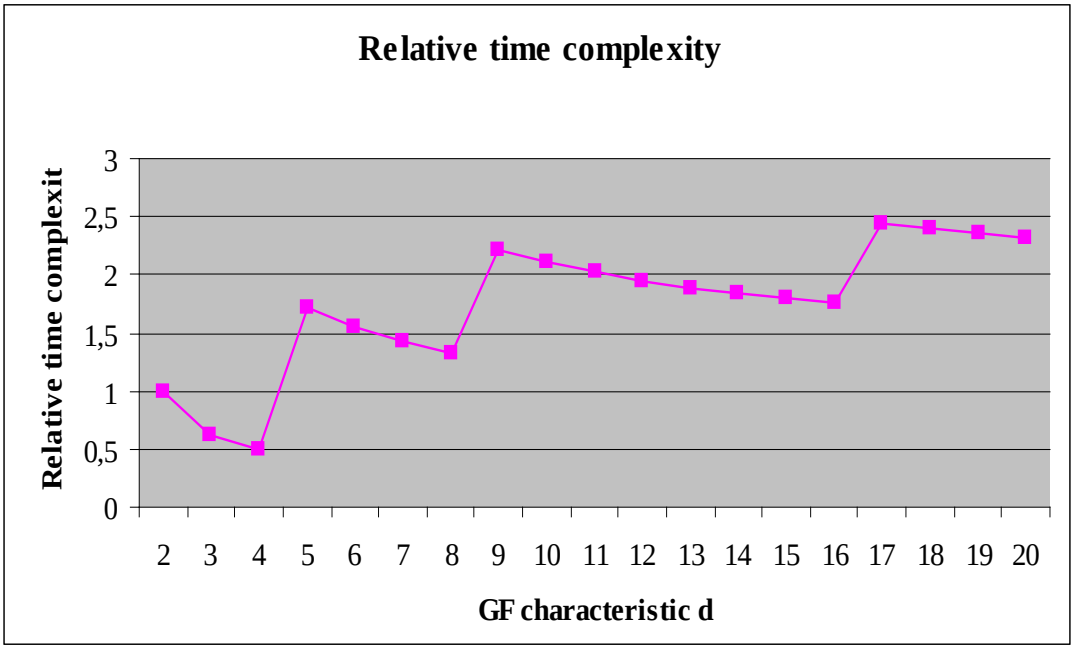

Fig. 2. Relative time complexity of hardware multipliers in different fields.

Results of hardware-software complexity estimation is shown in Fig. 3.

As can be seen from Fig. 3, the trial and binary Galois fields provide the best data protection. Fields with large characteristics provide weaker protection. The use of hardware tools for working in such fields has less effect than for working in binary and trial fields. As result the use of hardware tools to work in the field with the characteristic 768 [3] also has less effect than for working in binary and ternary fields.

The use of isogenies of supersingular elliptic curves is oriented toward usage of Galois field with big characteristics, so they are focused on software implementation.

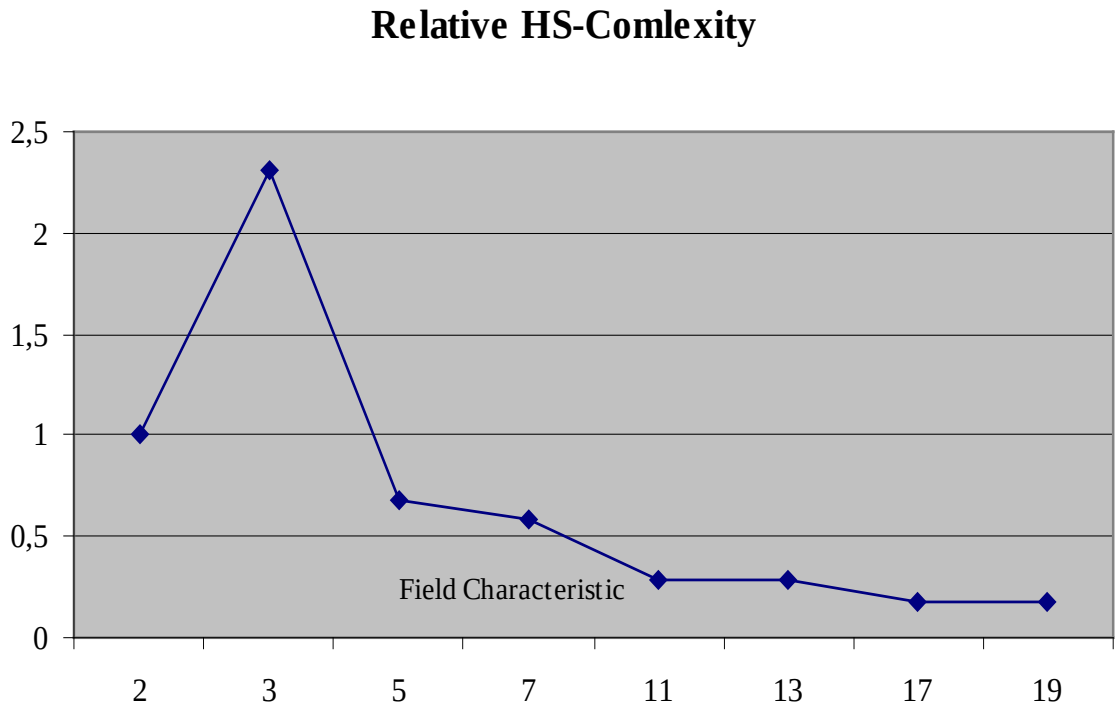

Fig. 3. Relative hardware-software time complexity of multipliers in different fields.

V. Conclusion

The use of isogenies of supersingular elliptic curves is oriented toward software implementation of the method. Hardware implementation of this method will not provide such a reduction in time complexity and increase the degree of data protection as it provides in methods oriented on binary fields.

References

[1] Quantum-safe cryptography. http://www.etsi.org/technologies-clusters/technologies/quantum-safe-cryptography

[2] Supersingular isogeny key exchange. http: https://en.wikipedia.org/wiki/Supersingular_isogeny_key_exchange

[3] De Feo, Luca; Jao, Plut. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies" (PDF). PQCrypto 2011. Springer. Retrieved4 May 2014.

[4] Fishbein, Dieter (30 April 2014). "Machine-Level Software Optimization of Cryptographic Protocols". University of Waterloo Library - Electronic Theses. University of Waterloo. Retrieved 21 June 2014.

[5] Costello, Craig; Longa, Patrick; Naehrig, Michael (2016-01-01). "Efficient algorithms for supersingular isogeny Diffie-Hellman"

[6] Koziel, Brian; Kermani, Mehran; Azarderakhsh, Reza (2016-11-07). "Fast Hardware Architectures for Supersingular Isogeny Diffie-Hellman Key Exchange on FPGA"

[7] Jean Vélu. Isogénies entre courbes elliptiques. Comptes Rendus de l'Académie des Sciences de Paris, 273:238–241, 1971.

[8] SIKE protocol and its stability to classical and quantum attacks. https://www.ruscrypto.ru/resource/archive/rc2018/files/02_Taraskin.pdf

[9] Password cracking. https://en.wikipedia.org/wiki/Password_cracking.

[10] Maple User Manual. Copyright © Maplesoft, a division of Waterloo Maple Inc. 2017.

[11] Hlukhov, V., Zholubak, I., Kostyk, A., Rahma M. (2017), Galois Fields Elements Processing Units for Cryptographic Data Protection in Cyber-Physical Systems. Advances in Cyber-Physical Systems. Volume II, Number 2, 2017. © Lviv Polytechnic National University, pp. 9-18, in press.

[12] Rodrigue Elias, Valerii Hlukhov, Mohammed Rahma, Ivan Zholubak· FPGA cores for fast multiplicative inverse calculation in Galois Fields. 9th International IEEE Conference Dependable Systems, Services and Technologies DESSERT'2018 UKRAINE, KYIV, MAY 24-27, 2018, in press.

[13] R. Elias, M. Rahma, V. Hlukhov. Multipliers for Galois fields Time Complexity. ELECTROTECHNIC AND COMPUTER SYSTEMS. Science and Technical. Odessa, No. 22 (98) 2016. Pp. 323-327 (In Ukrainian)