UNIVERSITI TEKNOLOGI MARA

TECHNICAL REPORT

IDENTITY AUTHENTICATION BY USING ZERO-KNOWLEDGE FIAT-SHAMIR PROTOCOL (P10S20)

NURANATASSIA IZLYN BINTI HASSAN (2018207464)

RAFIQAH EZLEEN BINTI RAZALI (2018440762)

MUHAMMAD NUR ARIF BIN NEKMAT (2018402368)

Report submitted in partial fulfillment of the requirement for the degree of
Bachelor of Science (Hons.) Computational Mathematics
Faculty of Computer and Mathematical Sciences

JULY 2020

ACKNOWLEDGEMENT

IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL

Firstly, we are grateful to Allah S.W.T for giving us the opportunity, strength, and patience to complete this final year project successfully. We would like to express our appreciation and gratitude to those who are helping us to complete this research. This is one of the most difficult semesters we have ever had, given that, due to COVID-19; we had to do our semester from home. Through this one-year journey, this work would not have been possible without the support of our supervisor Mr. Md Nizam bin Udin. We are very grateful and thankful to Allah S.W.T for his guidance, patience, and constant encouragement throughout this research. We also would like to thank Dr. Zati Aqmar binti Zaharudin and Dr. Nur Azlina binti Abdul Aziz (Prof. Madya) for their assistance and teaching through MSP660 and MAT530. Thank you for your guidance and advice on writing a better study report.

Finally, we would like to thank all those who helped us either directly or indirectly in order to finish this study. Especially our families and friends that have given us full support and encouragement for us to complete this project. We will strive to implement and share the knowledge acquired in the best and possible manner.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Identity authentications are widely used in technology industries to identify a person's identity. An example of identity authentication is the requirement to enter a username and password when you log in to a website. To prove the identity of a user, a zero-knowledge Fiat-Shamir protocol is used. Fiat-Shamir protocol is one of the well-known methods to verify the user without revealing the secret value that the user holds. Fiat-Shamir protocol is based on the difficulty of finding square root modulo a sufficiently large composite number of *n* and the factorization is unknown. In a previous study, the key used is large. Although the large key will enhance security, it takes very long computations and the costs are expensive. Therefore, the Fiat-Shamir protocol is impractical. On the other side, Elliptic Curve Cryptography (ECC) is one of the powerful cryptography that can be applied in modern technologies since it is more complex and convenient. ECC allows smaller keys compared to non-ECC to provide equivalent security. Our purpose of the study is to overcome the drawback of the Fiat-Shamir protocol by applying ECC in the Fiat-Shamir protocol. To prove the purpose of our study, we modify the Fiat-Shamir Protocol using the ECC method. Then, we develop Graphical User Interface (GUI) for the new protocol. Last but not least, to verify whether the algorithm is valid or not, we proof the secret key, *s* in algorithm.

1. INTRODUCTION

The identity of verifications and authentications are widely used for security in this modern era. In security, authentication is a method of authenticating a person's identity. In addition, authentication is distinct from acceptance, which is the mechanism by which device artefacts are granted individual access depending on their identification. Authentication implies the person who wants to be without revealing a single thing about the individual's access right. To do business safely with that specific individual, the risk of fraud must be minimized and built-up confidence. The authentication protocols are taken for granted in many access control systems. A common example would be the requirement to enter a username and password when you log in to a website. Most of the time, identification schemes are based on zero-knowledge interactive proofs. An interactive identification scheme can be developed using an interactive scheme when it is converted and a hash function. A zero-knowledge protocol is a method by which one prover can prove without revealing the secret value to the verifier. The claimant does not reveal anything that might endanger the confidentiality of the secret.

There is a type of zero-knowledge parallel protocol developed by Amos Fiat, and Adi Shamir known as the Fiat-Shamir Protocol, one of the most well-known protocols. Identification of Fiat-Shamir is constructed based on the difficulty of finding square root modulo a sufficiently large composite number n, the factorization is unknown. The prover processed a secret token and is seeking authentication and must be proved to another entity. The verifier is the one who must authenticate the provider based on the secret token the prover has (Fiat, 2015).

To illustrate zero-knowledge Fiat-Shamir protocol, we use Elliptic Curve Cryptography (ECC). ECC is a form of elliptical algebraic curves over finite fields. ECC is one of primitive public key cryptography that can be used to produce cryptographic keys faster, smaller, and more securely. ECC has been commonly used in industries such as the production of mobile apps since ECC helps to create equal protection with lower processing power and use of battery resources. This study identifies possibly the best-established protocols of Fiat-Shamir regarding the problem of a discrete logarithm and we modify them to the ECC setting.

1.1 Problem Statement

ECC is one of the powerful cryptography that can be applied in modern technologies since it is more complex and convenient. ECC allows smaller keys compared to non-ECC to provide equivalent security. In a previous study, the comparison between ECC and RSA in evolving capacity, providing an attractive and alternative way in cryptographic algorithms. It proves ECC is more convenient than RSA. There were also studies about applying ECC in Zero-Knowledge for smart city IoT deployment.

Fiat-Shamir protocol is the most well-known protocol identification using zero-knowledge proof protocol, which is based on the difficulty of finding square root modulo a sufficiently large composite number of *n* and the factorization is unknown. In a previous study, the key used is large. Although the large key will enhance security, it takes very long computations and the costs are expensive. Therefore, the Fiat-Shamir protocol is impractical.

Our purpose of the study is to overcome the drawback of the Fiat-Shamir protocol by applying ECC in the Fiat-Shamir protocol. By combining both methods, the key used in Fiat-Shamir protocol can be implement in a practical way.

1.2 Objective

1. To modify Fiat-Shamir Protocol by implementing Elliptic Curve Cryptography (ECC).
2. To develop Graphical User Interface (GUI) for the new protocol.

3. To verify whether the algorithm valid or not by proving the secret key, *s* in the algorithm.

1.3 Significance of the Study

The significance of this study is the modification of the Fiat-Shamir protocol by using Elliptic Curve Cryptography (ECC). The reason why Fiat-Shamir protocol and ECC are chosen to be combined is because Fiat-Shamir protocol is widely used in identity authentication and ECC can illustrate the assigned key from the calculation. Identity authentication is where the claimant needs to be verified by the verifier which is why the Fiat-Shamir protocol is the suitable method to be implemented. The modification of the Fiat-Shamir protocol gives more variation using multiple mathematical primitives. In our research, we are focusing on modifying the private key and public key in the Fiat-Shamir protocol. The addition law in ECC is used to determine the key to be illustrated. The public key in the Fiat-Shamir protocol is also being used in ECC. Since both methods can be related, then modifying the Fiat-Shamir protocol by using ECC is efficient. Plus, the efficiency and performance of both methods can be improved.

1.4 Scope and Limitation

In this research, we focused on the Fiat-Shamir Protocols which is a technique for taking interactive proof of knowledge of a certain secret number can be publicly proven without revealing underlying information. The method that has been used in this project is by combining two methods which involve Fiat-Shamir Protocol and Elliptic Curve Cryptography method. The Fiat-Shamir protocols is modified by using ECC method which converts the private and public key to point from ECC by using additive law and multiplicative law. From the point of ECC, we develop a GUI using a Maple software to generate the result to prove whether the claimant knows the secret key or not and meet the objectives of this research.

2. BACKGROUND THEORY AND LITERATURE REVIEW

2.1 Literature Review/ Related Research

2.1.1 Zero-knowledge Proof

Zero-knowledge proof is one of the methods in cryptography used to verify one's prover. By using zero-knowledge proof, the prover can be proved by the verifier by giving the prover a challenge that he/she has to complete. The prover needs to prove that he/she knows the secret value without revealing it. One of the reasons why most of the identity authentication uses zero-knowledge proof is because the essence of the zero-knowledge proof is that the prover can be proven by the verifier without revealing any additional information from both parties. There are several authentication protocols and systems regarding the security authentication and verification that can be used to secure the identity of the user.

2.1.2 Fiat-Shamir Protocol Method

In 1986, Amos Fiat and Adi Shamir introduce one of the most-known protocols for acknowledgement using a zero-knowledge proof method. Identification of Fiat-Shamir is constructed based on the difficulty of finding square root modulo a sufficiently large composite number *n*, the factorization is unknown. The prover processed a secret token and is seeking authentication and must be proved to another entity. The verifier is the one who must authenticate the provider based on the secret token the prover has Fiat-Shamir protocol.

During the same year, Desmedt and Quisquater had already clarified the fraudulent use of the Fiat-Shamir protocol. Their leading remark was that the Fiat-Shamir protocol defines confidential information instead of recognizing the individual. However, the Fiat-Shamir methods suffer from modern and well-known old malicious schemes if the physical representation is not accurate or not addressed properly, while its reliability is not based on zero-knowledge, Desmedt et al. (1988).

There's also some discovery between Non-interactive zero-knowledge (NIZK) protocols through the transmission of Fiat-Shamir protocols. The natural approach to developing NIZK protocols is to use the modification of Fiat-Shamir protocols. That prescribes a general way to eliminate interaction from public-coin interactive proof to transform an interactive into such a non-interactive one, Canetti et al. (2019).

According to Bernhard et al. (2011), the weakness of Fiat-Shamir proof is used as a source of attack somewhat greater than most of those previously proposed, they have presented current and unexpected complications of these insufficient data, and have proven that shifting to their strong counterpart enhances Helios to claim the privacy of voting and presents a key presumption on which current Helios verification analyses depend.

Apart from the Fiat-Shamir protocol, Rivest-Shamir-Adleman (RSA) is an example of public key cryptographic algorithms. RSA is the most commonly used public key cryptography cypher (PKC) at this time of year. Fiege-Fiat-Shamir is an encryption scheme based on public key cryptography. In this regard, it is identical to the original authentication protocol of SSH (Secure Shell) that uses RSA. Fiege-Fiat-Shamir therefore does not address key management problems. In particular, the Fiege-Fiat-Shamir protocol requires that the public key is released elsewhere by a trustworthy third party.

Fiege-Fiat-Shamir is a basic authentication protocol that can be used for login procedures. Unlike RSA, it is not possible to use it also for encryption. However, its strength over RSA is that it is much lighter computationally, Feige-Fiat-Shamir computations require only multiplications, while RSA uses power increases, Raffo (2015).

2.1.3 Elliptic Curve Cryptography Method

The rapid growth of technology especially in communication and internet of things (IOT) introduces a few challenges. Securing the security between two communication or more requires cryptographic algorithms and to implement some protocol in the devices. However, the IOT devices are often physically accessible for irresponsible users to get the information that hackers want. Researchers consider both resources constrained and side channels protected by applying Elliptic Curve Cryptography (ECC) Pirotte et al (2019). ECC is more suitable in constrained devices because the key size is shorter compared to the RSA. The study proposed special discrete log-based cryptosystems using additional points on an elliptic curve this point corresponding to point addition law. The addition law requires different equations for a pair of identical points and for a pair of different pointes. The basic way to implement an elliptic curve point multiplication is through conditional branching of point doublings and point additions. This is unfavorable for side-channel analysis attacks.

ECC is one of the methods for new cryptographic applications. ECC is more efficient compared to RSA because ECC requires smaller key sizes and possible faster computation that can lead to lower power consumption as well as better memory and bandwidth savings, Khleborodov (2018). This is the main advantage of ECC, the possibility to use short keys and at the same time provide a high level of security. An important aspect of the approach used in ECC, in correlation with other approaches used in cryptography such as Finite Field Cryptography (FFC) and Integer Factorization Cryptography (IFC), both have the same ability to provide the same security with significantly shorter keys. It reduced CPU performance, power usage and key storage requirements for hardware devices.

Rapid growth of technologies operating with important information from the user or the environment. For example, medical records, financial records and communication interaction that most of the people would like to keep private. Securing security must be guaranteed to enhance the quality and performance of service for the Internet of Things (IoT). The method available to reinforce security mechanisms in physical devices is

limited. The previous study, Lara-Nino et al (2020), lightweight cryptography is a proposed solution for securing the security services under constrained systems. A lightweight cryptography with more focus on asymmetric constructions is required for the IOT.

A recent study by Kohel (2011) of different models for elliptical curves found a method that is more efficient to use in cryptographic applications. The models provide more efficiency in computable algorithms for the group law than the standard Weierstrass model. Example the models induced by a rational torsion structure. The module structure of the space sections of the addition morphisms is analyzed. Then, the explicit dimension formulas for the spaces of sections are determined and apply the formulas to specific models of elliptic curves. Last but not least, we will highlight previous research on the combination of RSA and ECC methods.

2.1.4 RSA using ECC method

Elliptic Curve Digital Signature Algorithm (ECDSA), which is one of the variants of ECC proposed as an alternative to existing public key systems such as Digital Signature Algorithm (DSA) and Rivest Shamir Adleman (RSA), has received a great attention from industry and academia. The important segment for ECDSA 's attractiveness is that there is no defined sub-exponential algorithm to solve an elliptical curve discrete logarithm problem on a properly chosen elliptical curve. Therefore, it takes absolute exponential time to solve even though the best algorithm known to solve the underlying integer factorization for RSA and the discrete logarithm problem in DSA both take sub-exponential time. The key created by the implementation is highly protected and uses less bandwidth due to the limited key size used by the elliptical curves, Khalique (2010).

ECC is like RSA, a kind of public-key cryptosystem. But it differs from RSA from its smoother developing capability and provides cryptographic algorithm researchers with an appealing and alternative way of doing so. The security level which is given by RSA, can be provided even by smaller keys of ECC. Comparison between the two asymmetric cryptographic algorithms such as RSA and ECC, same degree of security, data size, encrypted message size and computational resources. Although ECC provides smaller keys than many other cryptographic algorithms (RSAs). In this research, they used binary methods to compare RSA and ECC, since implementation is the easiest, Amara & Siad (2011). From another article, the researchers presented a new index calculus algorithm using summation polynomials in elliptic curves specified over primary fields to overcome the discrete logarithm, Amadori et al. (2018).

The researcher discovered that the benefit of the elliptic curve groups rather than the multiplicative groups is that arithmetic operations need less execution time, less storage space and less energy consumption, as well as smaller message sizes that are worth less and have better execution chances. In addition, it's been shown that ECC actually performs RSA in computing time, memory requirements and hence energy usage in restricted environments, Chatzigiannakis et al. (2011).

According to Chung et al. (2007), The elliptic curve discrete logarithm problem is slightly more complex than the integer factorization problem. For the most part, the well-known RSA method must use 1024-bit keys, only then can it obtain computationally rational security; the ECC uses only 160-bit keys. Thus, at the same level of security, the ECC speed is many times faster than the RSA system; it can also save on key storage space. Clearly, whether it is in terms of protection or efficiency, the proposed scheme is superior to the signature scheme of Nyang and Song (2007).

Therefore, in this study we apply ECC in Fiat-Shamir protocols in order to prove that ECC can be apply in Fiat-Shamir protocols. Since both methods are complex algorithms, and both are efficient to be modified for ensuring the identity authentication.

3. METHODOLOGY AND IMPLEMENTATION

3.1 Overview

This section explains the research activities to ensure the research goal is achieved. Apart from that, the process is including all the implementations used and explained. The steps and flows for the methodology phases are shown based on Figure 1 below:

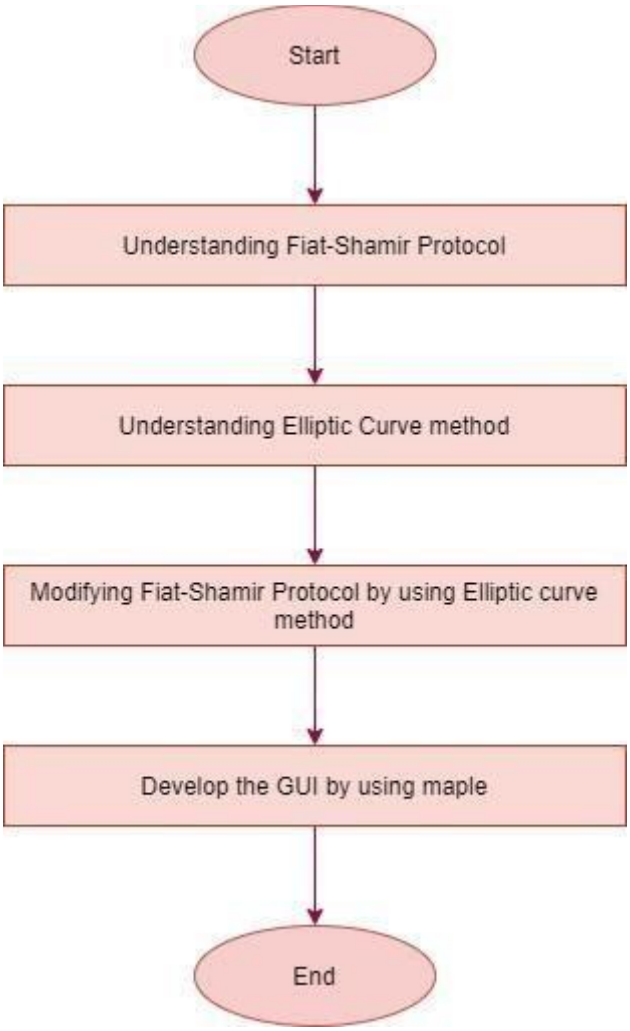

Figure 1: Flowchart of Methodology

The process was implemented step by step as stated at Figure 1 starting with understanding the basics of Fiat-Shamir protocol of zero-knowledge and followed by understanding the elliptic curve method. After understanding both Fiat-Shamir protocol and elliptic curve method, modifying Fiat-Shamir protocol by using elliptic curve method. Then, develop the GUI by using maple software.

3.2 Understanding Fiat-Shamir Protocol Method

Fiat-Shamir protocol enables one to transform an interactive protocol in a single message (non-interactive) protocol between a prover, *P* and a verifier, *V*. Thus, certain evidence for instance knowledge of a certain hidden number can be proved publicly without disclosing the underlying information. The original interactive proof must have the property of being a public number, for instance the random number of the verifier is made available in the proof process for the method to work.

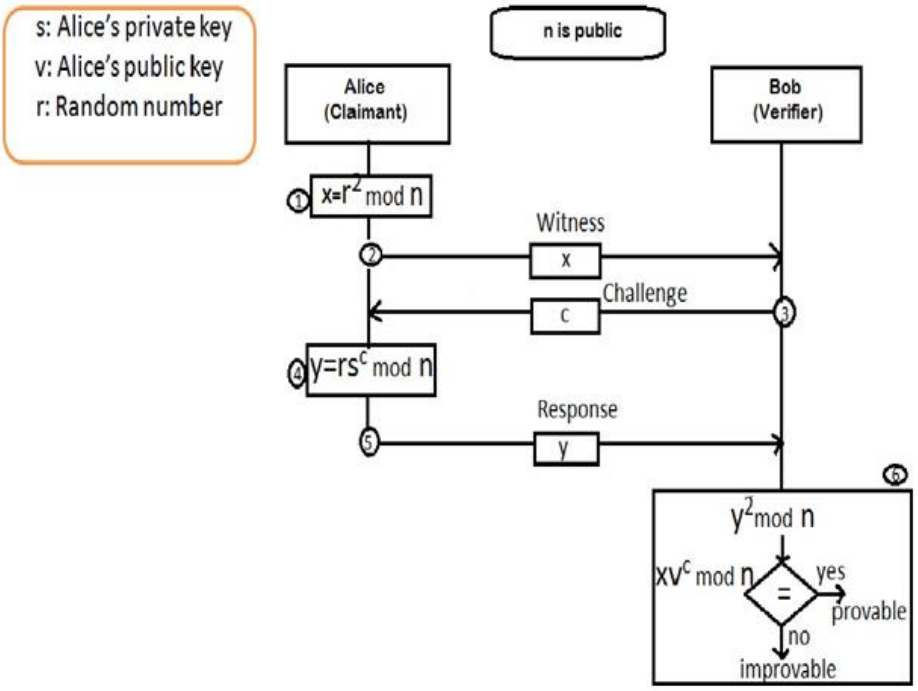

Figure 2: Fiat-Shamir Protocol

Based on Figure 2, we explain how does Fiat-Shamir Protocol works. To find the value of *p* by finding $k * l$ where *k* and *l* are kept secret.

Steps 1 : The claimant selects a random number, *r* from *0* to *p-1*. The claimant evaluates *x* as a witness.

$$x = r^2 \bmod p \quad (1)$$

Steps 2 : The claimant sends the value of *x* to the verifier.

Steps 3 : The verifier then sends a challenge number, *c* to the claimant where the challenge number is 0 and 1.

Steps 4 : From the challenge number, *c*. The claimant calculate the respond, *y*

$$y = rs^c \bmod p \quad (2)$$

where *s* is private key.

Steps 5 : The claimant then sends the value of *y* to the verifier.

Steps 6 : The verifier then will calculate the value of y^2 and xv^c. Where the public key, *v*

$$v = s^2 \bmod p \quad (3)$$

If y^2 and xv^c has the compatible value, then the claimant is said to be honest because they know the value of *s* or dishonest because they have found the value of *y* in other ways. The reason is that we can easily prove that $y^2 = xv^c$ in modulo *p* arithmetic as shown below:

$$y^2 = (rs^c)^2 = r^2(s^2)^c = xv^c \quad (4)$$

$$y^2 \bmod p = xv^c \bmod p \quad (5)$$

Example 1 Alsaedi et al. (2014), k = 5 and l = 11 then p = 55 is made public. Suppose claimant choose secret s = 14 and computes Equation (3) to $v = 14^2 \ mod \ 55 = 31$. The verifier chooses t = 2, and which number of t will the verifier select.

1. Claimant chose r = 9
2. Claimant sends $x = 9^2 \ mod \ 55 = 26$ from Equation (1) to verifier
3. Verifier sends c = 0 to claimant
4. Claimant sends $y = r$ = 9 to verifier
5. Verifier verifies $y \neq 0$ and $9^2 \ \ mod \ 55 = (26 \cdot 31^0) \ mod \ 55 \equiv 26$
6. Claimant chose r = 15
7. Claimant sends $x = 15^2 \ \ mod \ 55 = 45$ to verifier
8. Verifier sends c = 1 to Claimant
9. Claimant sends Equation (5.2.3) with mod 55 = 45 to verifier.
10. Verifier verifies $y \neq 0$ and $45^2 \ \ mod \ 55 = (15 \cdot 31^1) \ mod \ 55 \equiv 45$

The tenderness of this procedure is due to the fact that the prover has the hidden s and can also measure $y = r$ or $y = rs$, and send it to the verifier. An honest verifier will therefore always complete all t iterations and will agree with probability 1.

3.3 Understanding Elliptic Curve Method

The elliptic curve cryptosystem uses elliptic curves, where certain variables and equation are restricted to finite field elements divided by two groups, prime curves described over Z_p and binary curves constructed over GF (2^n), Jeng & Wang (2006). An ECC equation of the form:

$$E_P(a,b): y^2 = x^3 + a \cdot x + b \; mod \; p \tag{6}$$

Where *a*, *b*, are real numbers and *p* are prime number. The constant *a* and *b* are non-negative integer smaller than the prime number *p* and must satisfy the condition:

$$4a^3 + 27b^2 \; mod \; p \neq 0 \tag{7}$$

For each value of *x*, one needs to determine whether or not it is a quadratic residue. If *x* is the quadratic residue, then there are two values in the Elliptic group. If not, there are no points in Elliptic group.

3.3.1 Additive Law and Multiplicative Law

Additive and multiplicative law are sufficient and necessary to ensure that Equation (7) has no repeated factors, which means that a finite abelian group can be defined based on the set $E_p(a,b)$. An operation over $E_p(a,b)$ called additive law is used. For all $L, Q \in E_p(a,b)$and the rules as follows:

1. $L + O = L \cdot O$, additive identity. Thus $O = -O$.
2. If $L = (x_L, y_L)$ then $L + (x_L - y_L) = O$
3. If $L = (x_L, y_L)$and $Q = (x_Q, y_Q)$with $L \neq Q$ then $R = L + Q = (x_R, y_R)$ as determined by the following rules:
    1. $x_R = \left(m^2 - x_L - x_Q\right) mod \; p$ (8)
    2. $y_R = (m\,(x_L - x_R) - y_L) \; mod \; p$, where, (9)

$$m = \begin{cases} \left(\dfrac{y_Q - y_L}{x_Q - x_L}\right), & if \; L \neq Q \quad (10) \\ \left(\dfrac{3x_L^2 + a}{2y_L}\right), & if \; L = Q \quad (11) \end{cases}$$

(4) Multiplication by an integer is defined by repeated addition.

Example 2: Let $p = 11$ and consider the elliptic curve in Equation (6). In this case, let $a = 1$ and $b = 6$. To check whether the Equation is not repeated using Equation (7),

$$4a^3 + 27b^2 \ mod\ 11 \equiv 8 \ mod\ 11 \neq 0$$

To find the value of *y*, $1 < y < p - 1$:

$y_1 = y^2$	$y_2 = (p - y)^2$	y^2 mod 11
$\mathbf{1^2\ mod\ 11}$	$10^2\ mod\ 11$	1
$\mathbf{2^2\ mod\ 11}$	$9^2\ mod\ 11$	4
$\mathbf{3^2\ mod\ 11}$	$8^2\ mod\ 11$	9
$\mathbf{4^2\ mod\ 11}$	$7^2\ mod\ 11$	5
$\mathbf{5^2\ mod\ 11}$	$6^2\ mod\ 11$	3

Table 1: Example to find the value of *y*

Therefore, the set of quadratic residues $Q_{11} = \{1,3,4,5,9\}$. For $0 \leq x \leq p - 1$, computes in $x^3 + x + 6\ mod\ 11$ and determine if y^2 is in the set of Quadratic residues:

x	$x^3 + x + 6\ mod\ 11$	Points
0	6	-
1	8	-
2	5	(2,4), (2,7)
3	3	(3,5), (3,6)
4	8	-
5	4	(5,2), (5,9)
6	8	-
7	6	(7,2), (7,9)
8	9	(8,3), (8,8)
9	7	-
10	4	(10,2), (10,9)

Table 2: Example to find the value of *x* and Consisting Point

The set $E_{11}(1,6)$, consisting point (2,4), (2,7), (3,5) (3,6), (5,2), (5,9), (7,2), (7,9), (8,3), (8,8), (10,2) and (10,9)

By using Additive Law, Let $L = (3,5)$, $Q = (7,2)$ in $E_{11}(1,6)$, then $L \neq Q$:

$$m = \left(\frac{y_Q - y_L}{x_Q - x_L}\right), \qquad \left(\frac{2-5}{7-3}\right) mod\ 11 = \frac{-3}{4}\ mod\ 11 = 2$$

$$x_R = (2^2 - 3 - 7)\ mod\ 11 = -6\ mod\ 11 = 5$$

$$y_R = (2\ (3 - 5) - 5)\ mod\ 11 = -9\ mod\ 11 = 2$$

$$\therefore L + Q = (5,2)$$

By using Multiplicative Law, suppose point $L = (3,5)$ is add to itself in $E_{11}(1,6)$, then $2L$:

$$m = \left(\frac{3x_L^2 + a}{2y_L}\right), \qquad \left(\frac{3(3)^2 + 1}{2(5)}\right) mod\ 11 = \frac{6}{10}\ mod\ 11 = 5$$

$$x_R = (5^2 - 3 - 3)\ mod\ 11 = 19\ mod\ 11 = 8$$

$$y_R = (5\ (3 - 8) - 5)\ mod\ 11 = -30\ mod\ 11 = 3$$

$$\therefore 2L = (8,3)$$

To find 3L=2L+L, where L= (3,5) and 2L= (8,3), then 3L = (x_3, y_3):

$$m = \left(\frac{y_{2L} - y_L}{x_{2L} - x_L}\right), \qquad \left(\frac{3-5}{8-3}\right) mod\ 11 = \frac{-2}{5}\ mod\ 11 = 4$$

$$x_3 = (4^2 - 3 - 8)\ mod\ 11 = 5\ mod\ 11$$

$$y_3 = (4(3 - 5) - 5)\ mod\ 11 = -13\ mod\ 11 = 9$$

$$\therefore 3L = (5,9)$$

Therefore, point (5,2), (8,3) and (5,9) is required in set $E_{11}(1,6)$.

3.4 Modifying Fiat-Shamir Protocol by using Elliptic Curve Method

The next step after understanding the Fiat-Shamir protocol in 3.2 and understanding the Elliptic curve method in 3.3. This research modified Fiat-Shamir protocol by using an elliptic curve method.

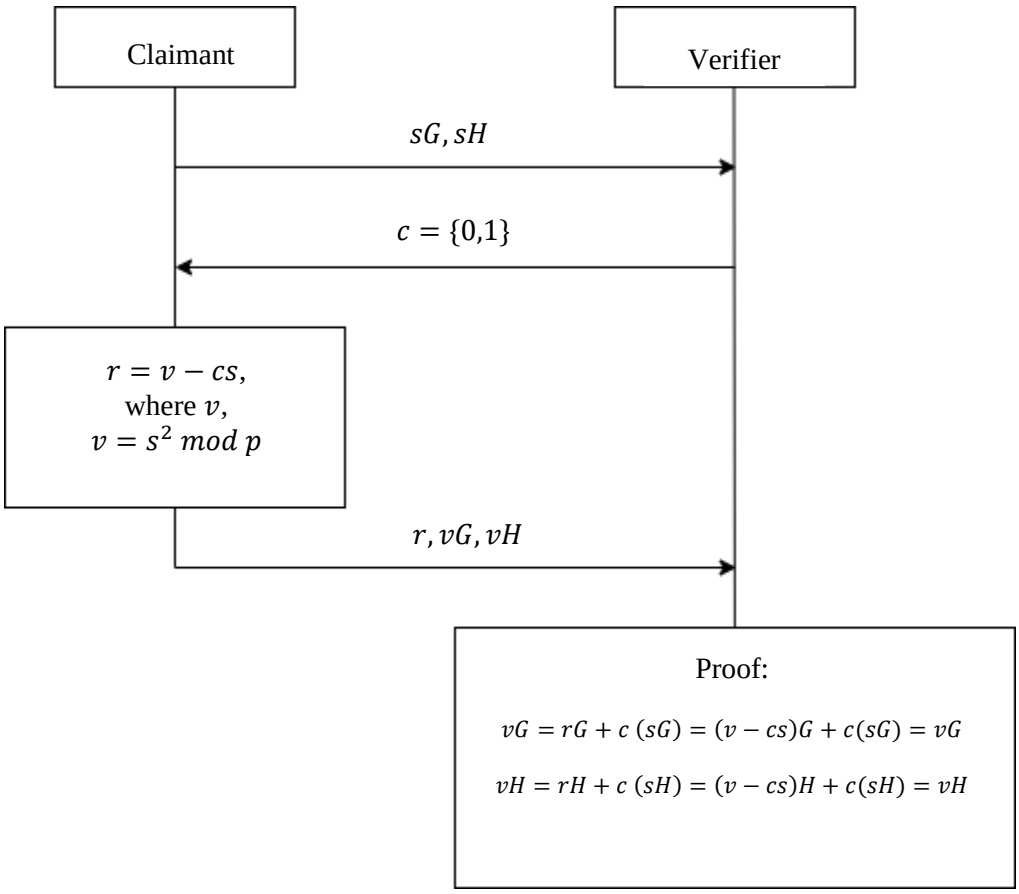

Figure 3: Modifying Fiat-Shamir Protocol by using Elliptic Curve Method

Step 1 : $E_p\ (a, b)$: $y^2 = x^3 + ax + b$, where p is the prime number. Where we set the value of a, b and p.

Step 2 : After find the consisting point from step 1, The verifier and claimant are agreed on two point on $E_p\ (a, b)$ as G and H

$$Point\ G: (x_G, y_G)$$
$$Point\ H: (x_H, y_H)$$

Step 3 : The claimant chooses secret keys, s then computes with point G and H into sG and sH using Addition Law and Multiplicative Law, then sends to the verifier.

Step 4 : The verifier then sends a challenge number, c to claimants

$$c = \{0,1\}$$

Step 5 : The claimant computes r,

$$r = v - cs \tag{12}$$

Where v,

$$v = s^2 mod\ p$$

Step 6 : The claimant computes vG and vH, then sends r, vG and vH to the verifier to check

$$vG(x_G, y_G) = rG(x_G, y_G) + c\big(sG(x_G, y_G)\big) mod\ p \tag{13}$$
$$vH(x_H, y_H) = rH(x_H, y_H) + c\big(sH(x_H, y_H)\big) mod\ p \tag{14}$$

Step 7 : To proof if the claimant knows the s value

$$\begin{aligned} vG(x_G, y_G) &= rG(x_G, y_G) + c\big(sG(x_G, y_G)\big) \\ &= (v - cs)G(x_G, y_G) + c\big(sG(x_G, y_G)\big) = vG(x_G, y_G) \end{aligned}$$

$$\begin{aligned} vH(x_H, y_H) &= rH(x_H, y_H) + c\big(sH(x_H, y_H)\big) \\ &= (v - cs)H(x_H, y_H) + c\big(sH(x_H, y_H)\big) = vH(x_H, y_H) \end{aligned}$$

4. RESULT AND DISCUSSION

4.1 CALCULATIONS OF MODIFIED FIAT-SHAMIR

This section provides some examples of purposed system. To define the systems, we have calculated to find consisting point using ECC.

Step 1: To find the consisting point of ECC.

The chosen elliptic curve in cryptosystem $E_p\ (a, b)$: $y^2 = x^3 + ax + b$, where $a = 4$, $b = 12$ and $p = 13$.

y^2	$(p-y)^2$	$(p-y)^2\ mod\ 13$
$\mathbf{1^2\ mod\ 13}$	$12^2\ mod\ 13$	1
$\mathbf{2^2\ mod\ 13}$	$11^2\ mod\ 13$	4
$\mathbf{3^2\ mod\ 13}$	$10^2\ mod\ 13$	9
$\mathbf{4^2\ mod\ 13}$	$9^2\ mod\ 13$	3
$\mathbf{5^2\ mod\ 13}$	$8^2\ mod\ 13$	12
$\mathbf{6^2\ mod\ 13}$	$7^2\ mod\ 13$	10

Table 3: To find the value of *y*

x	$x^3 + 4x + 12\ mod\ 13$	Points
0	12	(0,5) (0,8)
1	4	(1,2) (1,11)
2	2	-
3	12	(3,5) (3,8)
4	1	(4,1) (4,12)
5	1	(5,1) (5,12)
6	5	-
7	6	-
8	10	(8,6) (8,7)
9	10	(9,6) (9,7)
10	12	(10,5) (10,8)
11	9	(11,3) (11,10)
12	7	-

Table 4: To find the value of *x* and Consisting Point

The consisting point of $E_{13}\ (4,12)$ are:

(0,5), (0,8), (1,2), (1,11), (3,5), (3,8), (4,1), (4,12), (5,1), (5,12), (8,6), (8,7), (9,6), (9,7), (10,5), (10,8), (11,3) and (11,10).

Step 2: Assigning points.

The verifier and claimant are agreed on two point on E_{13} $(4,12)$ as *G* and *H*

$$Point\ G: (x_G, y_G) = (8,6)$$
$$Point\ H: (x_H, y_H) = (4,1)$$

Step 3: Evaluate secret key, *s* with points assigned.

The claimant chooses secret keys, *s* then computes with point *G* and *H* and sends to the verifier. The claimant selects a secret key between 1 and *p*-1. Where, *s = 11*

$11_{10} = 1011_2 = 8_{10} + 2_{10} + 1_{10}$

To find the value of sG:
$G = (8,6)$
$2G = (x_2, y_2)$

$$m_2 = \left(\frac{3x_G^2 + a}{2y_G}\right), \qquad \left(\frac{3(8)^2 + 4}{2(6)}\right) mod\ 13 = \frac{49}{3}\ mod\ 13 \equiv 12$$

$$x_2 = (12^2 - 8 - 8)\ mod\ 13 = 128\ mod\ 13 \equiv 11$$

$$y_2 = (12\ (8 - 11) - 6)\ mod\ 13 = -42\ mod\ 13 \equiv 10$$

$2G = (11,10)$, with the same doubling process, we will find $4G = (1,11), 8G = (10,8)\ldots$

$3G = (x_3, y_3) = 2G + G;$

$$m_3 = \left(\frac{y_2 - y_G}{x_2 - x_G}\right), \qquad \left(\frac{10 - 6}{11 - 8}\right) mod\ 13 = \frac{4}{3}\ mod\ 13 \equiv 10$$

$$x_3 = (10^2 - 8 - 11)\ mod\ 13 = 81\ mod\ 13 \equiv 3$$

$$y_3 = (10\ (8 - 3) - 6)\ mod\ 13 = 44\ mod\ 13 \equiv 5$$

$$3G = (3,5)$$

$11G = 8G + 3G;$

$$m_{11} = \left(\frac{y_8 - y_3}{x_8 - x_3}\right), \qquad \left(\frac{8-5}{10-3}\right) mod\ 13 = \frac{3}{7}\ mod\ 13 \equiv 6$$

$$x_{11} = (6^2 - 3 - 10)\ mod\ 13 = 23\ mod\ 13 \equiv 10$$

$$y_{11} = (6\ (3 - 10) - 5)\ mod\ 13 = -47\ mod\ 13 \equiv 5$$

$\therefore \mathbf{11}\boldsymbol{G} = (\mathbf{10}, \mathbf{5})$

To find the value of sH:

$H = (4,1)$
$2H = (x_2, y_2)$

$$m_2 = \left(\frac{3x_H^2 + a}{2y_H}\right), \qquad \left(\frac{3(4)^2 + 4}{2(1)}\right) mod\ 13 = \frac{52}{2}\ mod\ 13 \equiv 0$$

$$x_2 = (0^2 - 4 - 4)\ mod\ 13 = -8\ mod\ 13 \equiv 5$$

$$y_2 = (0\ (4 - 5) - 1)\ mod\ 13 = -1\ mod\ 13 \equiv 12$$

$2H = (5,12)$,with the same doubling process, we will find $4H = (0,5), 8H = (9,7)$…

$3H = (x_3, y_3) = 2H + H$;

$$m_3 = \left(\frac{y_2 - y_H}{x_2 - x_H}\right), \qquad \left(\frac{12-1}{5-4}\right) mod\ 13 = \frac{11}{1}\ mod\ 13 \equiv 11$$

$$x_3 = (11^2 - 4 - 5)\ mod\ 13 = 112\ mod\ 13 \equiv 8$$

$$y_3 = (11\ (4 - 8) - 1)\ mod\ 13 = -45\ mod\ 13 \equiv 7$$

$3H = (8,7)$

$11H = 8H + 3H$;

$$m_{11} = \left(\frac{y_8 - y_3}{x_8 - x_3}\right), \qquad \left(\frac{7-7}{9-8}\right) mod\ 13 = \frac{0}{1}\ mod\ 13 \equiv 0$$

$x_{11} = (0^2 - 8 - 9)\ mod\ 13 = -17\ mod\ 13 \equiv 9$ $y_{11} = (0\ (8 - 9) - 7)\ mod\ 13 = -7\ mod\ 13 \equiv 6$ $\therefore \mathbf{11H} = (\mathbf{9}, \mathbf{6})$

Step 4: Choosing challenge number, c.

The verifier then sends a challenge number, *c* to claimants where $c = \{0,1\}$.

Step 5: Find the value of claimant random number, v and public random number, r.

Before finding a random number, *r*. The claimant random number, *v*.

$\boldsymbol{v = s^2\ mod\ p}$	$v = 11^2\ mod\ 13$ $= 121\ mod\ 13$ $= 4\ mod\ 13$
$\boldsymbol{r = v - cs}$	$c = 0,$ $r = 4 - (0)(11)$ $\equiv 4\ mod\ 13$ $c = 1,$ $r = 4 - (1)(11)$ $\equiv -7\ mod\ 13$ $\equiv 6\ mod\ 13$

Step 6: Evaluate vG and vH.

The claimant computes vG and vH, then sends *r*, vG and vH to the verifier to check

$\boldsymbol{vG = rG + c(sG)\ mod\ 13}$	$c = 0, r = 4,$ $\equiv 4(8,6) + (0)(10,5)\ mod\ 13$ $\equiv (1,11) + 0\ mod\ 13$ $\equiv (1,11)$ $c = 1, r = 6,$ $\equiv 6(8,6) + (1)(10,5)\ mod\ 13$ $\equiv (4,1) + (10,5) mod\ 13$ $\equiv$ (11,3) mod 13

$\boldsymbol{vH = rH + c(sH)\ mod\ 13}$	$c = 0, r = 4,$ $\equiv 4(4,1) + (0)(9,6)\ mod\ 13$ $\equiv (0,5) + 0\ mod\ 13$ $\equiv (0,5)$ $c = 1, r = 6,$ $\equiv 6(4,1) + (1)(9,6)\ mod\ 13$ $\equiv (11,3) + (9,6) mod\ 13$ $\equiv (5,1) mod\ 13$

Step 7: To verify algorithm by proving the secret key, s.

To proof if the claimant knows the *s* value.

$\boldsymbol{vG = rG + c(sG) = (v - cs)G + c(sG) = vG}$
$c = 0,$ $4(8,6) + (0)(10,5) = \big(4 - (0)(11)\big)(8,6) + (0)(10,5) = 4(8,6)$ $(1,11) = (1,11) = (1,11)$ $c = 1,$ $6(8,6) + (1)(10,5) = \big(4 - (1)(11)\big)(8,6) + (1)(10,5) = 6(8,6)$ $(11,3) = (11,3) = (11,3)$
$\boldsymbol{vH = rH + c(sH) = (v - cs)H + c(sH) = vH}$
$c = 0,$ $4(4,1) + (0)(9,6) = \big(4 - (0)(11)\big)(4,1) + (0)(9,6) = 4(4,1)$ $(0,5) = (0,5) = (0,5)$ $c = 1,$ $6(4,1) + (1)(9,6) = \big(4 - (1)(11)\big)(4,1) + (1)(9,6) = 6(4,1)$ $(5,1) = (5,1) = (5,1)$

By substituting the value of *a = 4*, *b = 12* and *p = 13* in elliptic curve,$E_p(a, b)$: $y^2 = x^3 + ax + b$, we get $E_{13}(4,12)$. Then, the consisting point in $E_{13}(4,12)$ are obtained. By using consisting point in $E_{13}(4,12)$, the claimant and verifier are agreed as point G: $(x_G, y_G) = (8,6)$ and point H: $(x_H, y_H) = (4,1)$. The user input the secret key, $s = 11$ to computes with point G and H by using additive law and multiplicative law. The verifier then sends a challenge number, *c* to claimants. The claimant computes r and then

generate v. The value of vG and vH are proved by the condition given. So, the verifier verifies the claimant and it is proved that the claimant is true.

4.2 Graphical User Interface (GUI) By Parts

This section shows our GUI by using maple software. This section explains every part of GUI that we had implement.

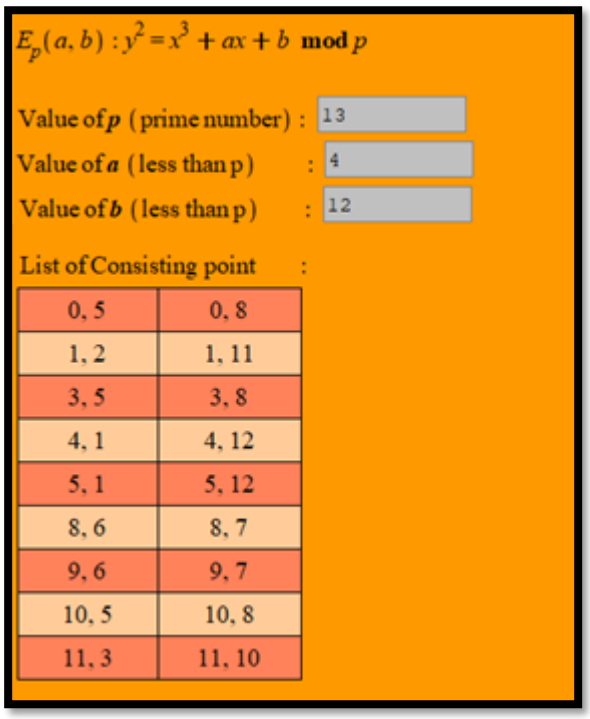

Figure 4: Explanation to get consisting point

In figure 4, step 1 is applied which is to find the variable in elliptic curve method such as value of p, a and b. Then, manually calculate the equation to get the list consisting point.

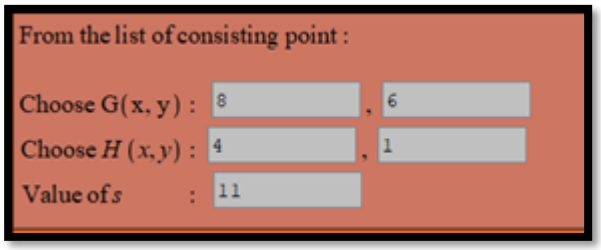

Figure 5: Verifier and claimant are agreed on two points

Choose point G and H from the list of consisting point. Step 2 in the modifying fiat-shamir protocol. Then, the user input the secret key, s.

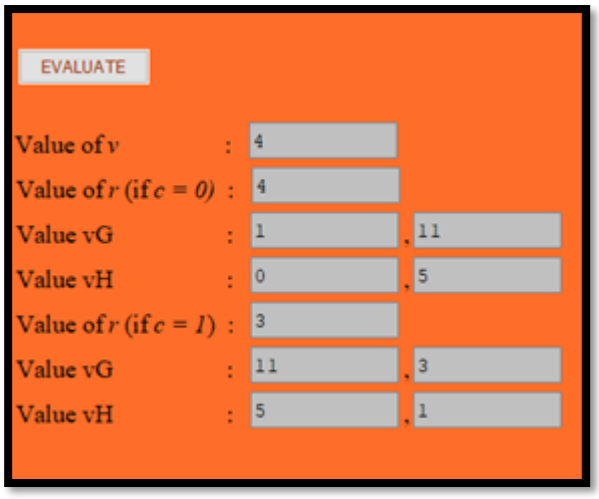

Figure 6: Proving the secret key, s

In figure 6, evaluate the point G and H by applying step 3 until step 7. To proof that claimant knows the secret key, s. The value vG and vH must consist in the list of consisting point.

5. CONCLUSION AND RECOMMENDATIONS

Overall overview of the two methods in cryptosystems which is Fiat-Shamir Protocol and Elliptic Curve Cryptography (ECC) is implemented. As we can conclude based on our studies that has been carried out, modifying Fiat-Shamir protocol with ECC are proved. There are a few conditions that has to be meet in Fiat-Shamir and ECC to be combine such as the main constraint of the systems are private key and public key. The conditional points addition and points doubling schemes of ECC are used in modified equations to proof the consisting point that we retrieved in the early steps.

Based on the original Fiat-Shamir protocol, although the large key will enhance security, but the key used is impractical to use on its own as it takes very long computations and the costs are expensive.

There are many benefits of using ECC includes a smaller size of key for security enhancement, the possibility to implement without cryptosystem processor, and the faster execution in some cases when using a cryptosystem processor.

In consequences, the benefits of ECC helps to cover the drawbacks of the Fiat-Shamir protocol. The key used in Fiat-Shamir protocol can be change to smaller size without changing the complexity of the systems. Furthermore, the computations of Fiat-Shamir protocol will reduce to a shorter time and the costs can be cut-off.

For future works, there is plenty of recommendation that can be suggested to improve the modified cryptosystem. Firstly, improving the algorithm by using a large value for the public key and private key since we only prove the method by using a small value of the public key and private key. Next, it is recommended for future studies, carryout the plaintext by using the hash function for the encrypting process. Lastly, we recommend for better development process for GUI build using different languages such as JAVA, Python, and others. Due to time limitations, there are minor flaws that can be improved for this modified method to be able to work perfectly. We hoped that in the future our studies can be extended to build a secure and safe modified Fiat-Shamir Protocol cryptosystem.

6. REFERENCES

Alsaedi, R., Constantinescu, N., Campus, J., & Arabia, S. (2014). *Nonlinearities in Elliptic Curve Authentication*. 5144–5158. https://doi.org/10.3390/e16095144

Amadori, A., Pintore, F., & Sala, M. (2018). On the discrete logarithm problem for prime-field elliptic curves. *Finite Fields and Their Applications*, *51*, 168–182. https://doi.org/10.1016/j.ffa.2018.01.009

Amara, M., & Siad, A. (2011). Elliptic Curve Cryptography and its applications. *7th International Workshop on Systems, Signal Processing and Their Applications, WoSSPA 2011*, 247–250. https://doi.org/10.1109/WOSSPA.2011.5931464

Bernhard, D., Pereira, O., & Warinschi, B. (n.d.). *How not to Prove Yourself : Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios*. 1–26.

Canetti, R., Chen, Y., Holmgren, J., Lombardi, A., Rothblum, G. N., Rothblum, R. D., & Wichs, D. (2019). Fiat-Shamir: From practice to theory. *Proceedings of the Annual ACM Symposium on Theory of Computing*, 1082–1090. https://doi.org/10.1145/3313276.3316380

Canetti, R., Lombardi, A., & Wichs, D. (2019). Fiat-Shamir: From Practice to Theory, Part II NIZK and Correlation Intractability from Circular-Secure FHE. *Stoc 2019*. https://eprint.iacr.org/2018/1248.pdf

Chatzigiannakis, I., Pyrgelis, A., Spirakis, P. G., & Stamatiou, Y. C. (2011). *Elliptic Curve Based Zero Knowledge Proofs and Their Applicability on Resource Constrained Devices*. https://doi.org/10.1109/MASS.2011.77

Chatzigiannakis, I., Vitaletti, A., & Pyrgelis, A. (2016). A privacy-preserving smart parking system using an IoT elliptic curve based security platform. *Computer Communications*, *89–90*, 165–177. https://doi.org/10.1016/j.comcom.2016.03.014

Chung, Y. F., Huang, K. H., Lai, F., & Chen, T. S. (2007). ID-based digital signature scheme on the elliptic curve cryptosystem. *Computer Standards and Interfaces*, *29*(6), 601–604. https://doi.org/10.1016/j.csi.2007.01.004

Desmedt, Y., Goutier, C., & Bengio, S. (1988). Special uses and abuses of the fiat-shamir passport protocol. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *293 LNCS*, 21–39. https://doi.org/10.1007/3-540-48184-2_3

Fiat - Shamir protocol. (2020). *December 2013*, 6911. http://cryptowiki.net/index.php?title=Fiat_-_Shamir_protocol

Fiat, A. (2015). *Zero-knowledge proofs of identity of Identity*. *June 1988*. https://doi.org/10.1007/BF02351717

Khalique, A. (2010). *Implementation of Elliptic Curve Digital Signature Algorithm*. *2*(2), 21–27.

Khleborodov, D. (2018). Fast elliptic curve point multiplication based on window Non-Adjacent Form method. *Applied Mathematics and Computation*, *334*, 41–59. https://doi.org/10.1016/j.amc.2018.03.112

Koblitz, B. N. (1987). *Elliptic Curve Cryptosystems*. *4*(177), 203–209.

Kohel, D. (2011). Addition law structure of elliptic curves. *Journal of Number Theory*, *131*(5), 894–919. https://doi.org/10.1016/j.jnt.2010.12.001

Lara-Nino, C. A., Diaz-Perez, A., & Morales-Sandoval, M. (2020). Lightweight elliptic

curve cryptography accelerator for internet of things applications. *Ad Hoc Networks*, *103*, 102159. https://doi.org/10.1016/j.adhoc.2020.102159

Pirotte, N., Vliegen, J., Batina, L., & Mentens, N. (2019). Balancing elliptic curve coprocessors from bottom to top. *Microprocessors and Microsystems*, *71*, 102866. https://doi.org/10.1016/j.micpro.2019.102866

Raffo, D. (2015). *Traineeship report Digital Certificates and the Feige-Fiat-Shamir zero-knowledge Daniele Raffo Supervisor : François Morain*. *July 2002*.

Smart, N. P. (1999). The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, *12*(3), 193–196. https://doi.org/10.1007/s001459900052

7. APPENDICES

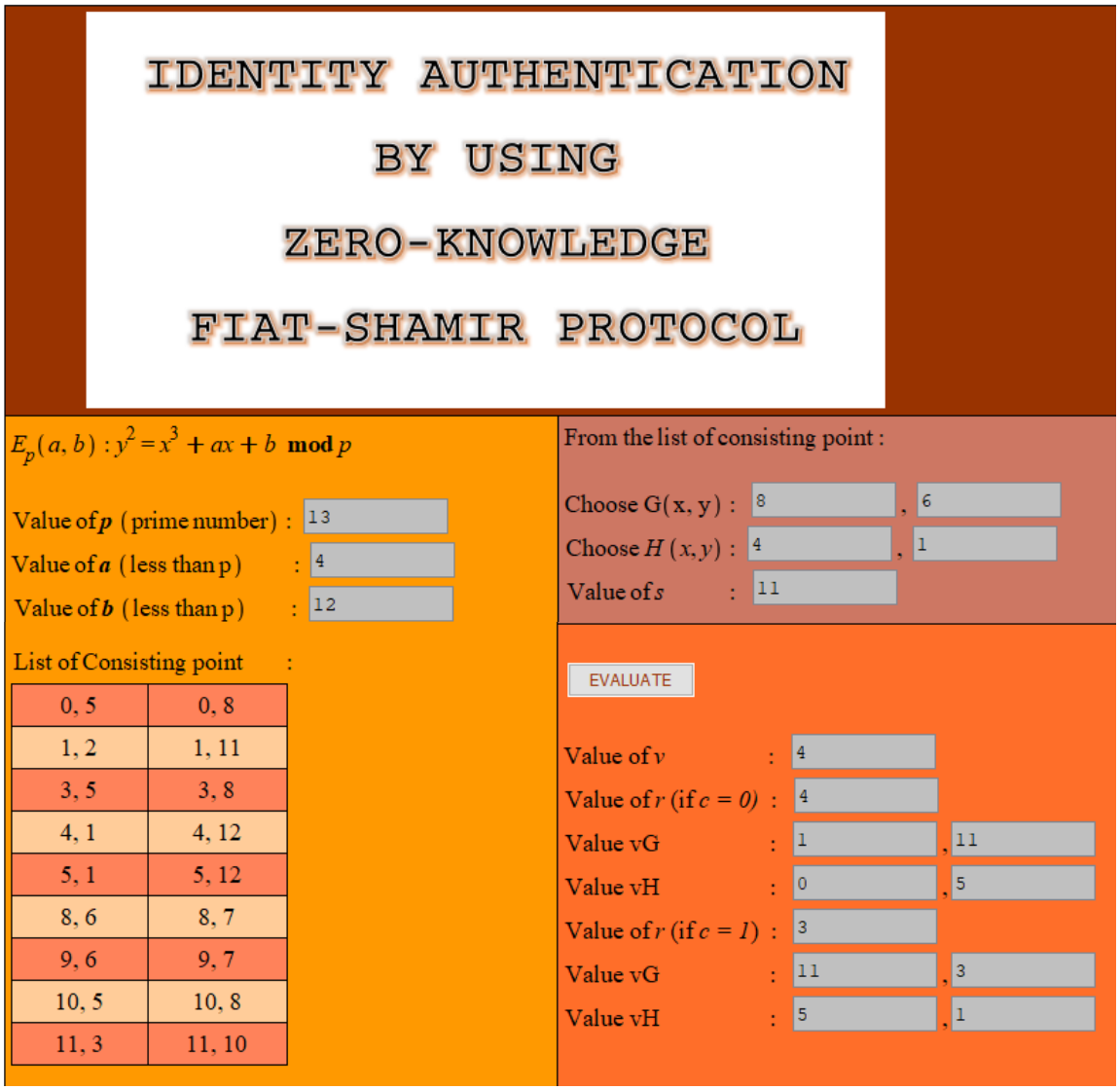

Figure 7: GUI using maple software