Entropy **2014**, *16*, 5144-5158; doi:10.3390/e16095144

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

Nonlinearities in Elliptic Curve Authentication

Ramzi Alsaedi [1], Nicolae Constantinescu [2] and Vicenţiu Rădulescu [1,3,*]

[1] Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah Campus, P.O. Box 80203, Jeddah 21589, Saudi Arabia; E-Mail: ramzialsaedi@yahoo.co.uk

[2] Department of Informatics, University of Craiova, Street A.I Cuza 13, 200585 Craiova, Romania; E-Mail: nicolaeconstantinescu@yahoo.com

[3] Institute of Mathematics "Simion Stoilow" of the Romanian Academy, P.O. Box 1-764, 014700 Bucharest, Romania

* Author to whom correspondence should be addressed; E-Mail: radulescu@inf.ucv.ro; Tel.:+40-723-049-852; Fax:+40-251-412-673.

Received: 2 September 2014 / Accepted: 19 September 2014 / Published: 25 September 2014

Abstract: In order to construct the border solutions for nonsupersingular elliptic curve equations, some common used models need to be adapted from linear treated cases for use in particular nonlinear cases. There are some approaches that conclude with these solutions. Optimization in this area means finding the majority of points on the elliptic curve and minimizing the time to compute the solution in contrast with the necessary time to compute the inverse solution. We can compute the positive solution of PDE (partial differential equation) like oscillations of $f(s)/s$ around the principal eigenvalue λ_1 of $-\Delta$ in $H_0^1(\Omega)$. Translating mathematics into cryptographic applications will be relevant in everyday life, wherein there are situations in which two parts that communicate need a third part to confirm this process. For example, if two persons want to agree on something they need an impartial person to confirm this agreement, like a notary. This third part does not influence in any way the communication process. It is just a witness to the agreement. We present a system where the communicating parties do not authenticate one another. Each party authenticates itself to a third part who also sends the keys for the encryption/decryption process. Another advantage of such a system is that if someone (sender) wants to transmit messages to more than one person (receivers), he needs only one authentication, unlike the classic systems where he would need to authenticate himself to each receiver. We propose an authentication method based on zero-knowledge and elliptic curves.

Keywords: partial differential equation; elliptic curve; zero knowledge; authentication

---

1. Introduction

The system we propose has three components: two parties that communicate and one party that authenticates them and provides the keys for the cryptosystem used. The most common authentication is based on passwords, which help to verify the identity of a user. This method is not secure enough because the passwords are generated from small dictionaries or they are chosen directly by the users who usually make poor selections. In addition, users frequently forget passwords. In such cases, an authentication system needs two authentication modes. The first mode is the primary one, and the second is the emergency one (it is used only when the primary is not available). The most popular emergency mode used on the Internet when a password is forgotten is the e-mail. The password or the instructions to reset it are sent by e-mail. The first password authentication protocol used on a network proven secure was presented by Halevi and Krawczyk [1]. Their protocol prevents leakage of information and the server's private key can be verified by the user. If the server's key cannot be verified it is recommended to use strong password authentication protocols. Such protocols were proposed by Bellovin and Merritt [2,3], Jablon [4] and Wu [5], among others.

We propose a zero-knowledge authentication using elliptic curves. A zero-knowledge proof is a proof of some statement that reveals nothing else but the veracity of the statement. In order to give a formal definition for a zero-knowledge proof, we will first define the interactive proof system.

Definition 1. *An interactive proof system for a set A is a process between a verifier which executes a probabilistic polynomial-time strategy and a prover, which executes a computationally unbounded strategy satisfying:*

- ***Completeness**: For any $a \in A$, the verifier always accepts the common input a (after interacting with the prover).*
- ***Soundness**: For some polynomial p, for any $x \notin A$ and any potential strategy S, the verifier rejects the common input a with a probability of at least $\frac{1}{p(|a|)}$ (after interacting with S).*

Therefore, a proof is complete if an honest verifier is always convinced of the veracity of a statement from an honest prover, and it is sound if a cheating prover can convince an honest verifier with a very small probability that a false statement is true.

Definition 2. *A strategy S is zero-knowledge on the set A if for any feasible strategy B exists a feasible computation C so that the following are computationally indistinguishable:*

- *the output of B after interacting with S on common input $a \in A$*
- *the output of C on input $a \in A$*

From this definition, any information obtained by interacting with S on some input a, can also be obtained from a without interacting with S [6]. In our method, the verifier knows the right answer before communicating with the prover. Therefore, he cannot possibly obtain any new information. This method is called "no-leak" authentication. A formal definition can be obtained from the zero-knowledge definition given above by eliminating "probabilistic polynomial time". This means that whatever

the verifier can compute after communicating with the prover, he could already compute before the communicating process. Like the verifier, a passive adversary cannot obtain new information from the prover.

2. State of Art

2.1. Mathematical Preliminaries

To understand the foundation of the cryptosystem functionality, we have to understand how the secret can be hidden and how it can be revealed ([7] and [8]). This is pure mathematics, and is based on some function operation intractability.

Definition 3. *The Waierstrass mathematical model is the basement:*

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where $a_i \in K$ and K represents the field over which the curve is defined. From this point we have the discriminant:

$$\Delta = d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

with:

$$\begin{aligned} d_2 &= a_1 + 4a_2 \\ d_4 &= 2a_4 + a_1a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \end{aligned}$$

and $\Delta \neq 0$.

If we have $K = F_p$ where $p > 3$ is a prime, Equation (1) can be simplified to:

$$E : y^2 = x^3 + ax + b$$

and the discriminant: $\Delta = -16(4a^3 + 27b^2)$. In case of $K = F_{2^m}$ we have:

$$E : y^2 = x^3 + ax + b$$

and the discriminant: $\Delta = b$. If the curve E is defined over a prime field F_p and we have a point $P(x, y) \in E$ then the inverse of it will be $-P(x, -y)$. If we want to compute $R(x_3, y_3) = P + Q$ where $P(x_1, y_1) \in E$ and $Q(x_2, y_2) \in E$ we have:

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

where λ is given by:

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2}$$

For doubling a point $2P(x_3, y_3)$ we use the formulas:

$$x_3 = \lambda^2 - 2x_1$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

where λ is given by:

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

For the affine coordinates we replace x with x/z and y with y/z, where $z \neq 0$ obtaining the equation:

$$y^2 z = x^3 + axz^2 + bz^3$$

To compute $P(x_1, y_1, z_1) + Q(x_2, y_2, z_2) = R(x_3, y_3, z_3)$ we have:

$\lambda_1 = x_1 z_2^2$

$\lambda_2 = x_2 z_1^2$

$\lambda_3 = \lambda_1 - \lambda_2$

$\lambda_4 = y_1 z_2^3$

$\lambda_5 = y_2 z_1^3$

$\lambda_6 = \lambda_4 - \lambda_5$

$\lambda_7 = \lambda_1 + \lambda_2$

$\lambda_8 = \lambda_4 + \lambda_5$

$z_3 = z_1 z_2 \lambda_3$

$x_3 = \lambda_6^2 - \lambda_7 \lambda_3^2$

$\lambda_9 = \lambda_7 \lambda_3^2 - 2x_3$

$y_3 = (\lambda_9 \lambda_6 - \lambda_8 \lambda_3^3)/2$

For doubling a point $2P(x_3, y_3, z_3)$ we use:

$\lambda_1 = 3x_1^2 + az_1^4$

$z_3 = 2y_1 z_1$

$\lambda_2 = 4x_1 y_1^2$

$x_3 = \lambda_1^2 - 2\lambda_2$

$y_3 = \lambda_1(\lambda_2 - x_3) - 8y_1^4$

If the curve E is defined over a binary field F_{2^m} for a point $P(x, y)$ the inverse will be $-P(x, x + y)$. Addition and doubling are defined in the same way as on the prime curves.

To obtain the projective coordinates we proceed as above. The inverse of a point $P(x, y, z)$ is $-P(x, x + y, z)$. To compute $P + Q = R$ we have:

$$\lambda_1 = x_1 z_2^2$$

$$\lambda_2 = x_2 z_1^2$$

$$\lambda_3 = \lambda_1 + \lambda_2$$

$$\lambda_4 = y_1 z_2^3$$

$$\lambda_5 = y_2 z_1^3$$

$$\lambda_6 = \lambda_4 + \lambda_5$$

$$\lambda_7 = z_1 \lambda_3$$

$$\lambda_8 = \lambda_6 x_2 + \lambda_7 y_2$$

$$z_3 = z_2 \lambda_7$$

$$\lambda_9 = \lambda_6 + z_3$$

$$x_3 = a z_3^2 + \lambda_6 \lambda_9 + \lambda_3^2$$

$$y_3 = \lambda_9 x_3 + \lambda_8 \lambda_7^2$$

And for doubling a point $2P$ we have:

$$z_3 = x_1 z_1^2$$

$$x_3 = x_1^4 + b z_1^8$$

$$\lambda = z_3 + x_1^2 + y_1 z_1$$

$$y_3 = x_1^4 z_3 + \lambda x_3$$

2.1.1. Frontier Points on Elliptic Curves

According with [9], from all points which define an elliptic curve, only a part can be used on applications (cryptography), we can found the special points with properties in this way, called frontier points:

(1) $|E(F_p)| = c \cdot l$ where $l > 2^{160}$ a prime and c a positive integer. $|E(F_p)|$ denotes the cardinal of the set of points on E over F_p.

(2) $l \neq p$.

(3) the order of the prime p in the multiplicative group $F_l^{\times}$ of F_l is at least $\lceil 2000/\log_2 p \rceil$.

These three conditions provide a high level of security. There were developed as algorithms for resolving discrete logarithms with running time equal with the square root of the largest prime factor of the group order [10]. These algorithms cannot be applied to a cryprosystem, which respects the first condition. [11] describes the anomalous curve attack. This attack consists in resolving the elliptic curve discrete logarithm problem for curves with the group order equal to the order of the finite field. The method uses Hensel's lemma and has low complexity. The second condition presented above makes this kind of attack impossible. In [12] the authors presented an attack which reduces the discrete logarithm problem in $E(F_p)$ to one in a finite extension field F_p. The third condition depends on the assumption that the DLP in a finite field which has a cardinal 2000-bit long is intractable.

The efficiency of an elliptic curve cryptosystem is based on the arithmetic in F_p. So the efficiency is directly proportional with p. This means that $|E(F_p)|$ must be as small as possible. From the first condition we have $|E(F_p)| = c \cdot l$ where $l > 2^{160}$. So the efficiency depends on the co-factor c. The first condition becomes:

- $|E(F_p)| = c \cdot l$ where $l > 2^{160}$ a prime and $c \leq 4$ a positive integer. $|E(F_p)|$ denotes the cardinal of the set of points on E over F_p.

2.1.2. Nonliniarities on Elliptic Curves

For every elliptic curve cryptosystem we have to declare the domain parameters. We will work with a nonsupersingular elliptic curve E defined over a prime field. The domain parameters will be $(F, p, a_E, b_E, G, n, h)$ where F_p is the prime field, a_E, b_E define the curve $E : y^2 = x^3 + a_E x + b_E$, $G \in E$ is a point of order n (this means that n is the smallest positive number for which $nG = O$), $h = |E(F_p)|/n$ is the co-factor. To meet the above conditions it is recommended for $|E(F_p)|$ to be prime or $|E(F_p)| = h \cdot n$ where n is a large prime and $h \in \{1, 2, 3, 4\}$ [13].

As is described in [14], starting from an oscillation $\theta(t)\backslash t$ around the principal eigenvalue λ_1 of $-\Delta$ in $H_0^1(\Omega)$ in one dimensional case will generate infinitely many solutions if $\theta(t) > 0$ in $\mathbb{R}$ and

$$\lim_{t\to\infty} inf \frac{2\psi(t)}{t^2} < \lambda_1 < \lim_{t\to+\infty} sup \frac{2\psi(t)}{t^2},$$

where $\psi(t) = \int_0^t \theta(\xi) d\xi$.

These conditions, as is proved in [15] can not be replaced by:

$$\lim_{t\to+\infty} inf \frac{\theta(t)}{t} < \lambda_1 < \lim_{t\to+\infty} sup \frac{\theta(t)}{t}$$

nor by

$$\lim_{t\to+\infty} inf \frac{\theta(t)}{t} = 0 \text{ and } \lim_{t\to+\infty} sup \frac{\theta(t)}{t} = +\infty$$

The results of these conclude in [16]

$$\begin{cases} -\Delta u = \theta(x, u) & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\theta : \overline{\Omega} : \mathbb{R} \to \mathbb{R}$ is a continuous function. In [14] it is stated

$$\psi(x, t) = \int_0^x \theta(x, \xi) d\xi$$

and it is defined the functional $\Phi : H_0^1(\Omega) \cap L^\infty(\Omega) \to \mathbb{R}$ with $\Phi(u) = \frac{1}{2}\int_\Omega |\Lambda u|^2 dx - \int_\Omega \psi(x,u)dx$ as generator of infinitely solutions. From these, the space of chosen criteria for cryptographic points is big enough such that can be considered as space of strong points in cryptography.

2.1.3. Counting the Elliptic Curve's Frontier Points

To know the amount of points belonging to the elliptic curve we have to compute $|E(F_p)|$. In 1985 [17] Schoof presented an algorithm for counting the points on an elliptic curve over a large field F_p. Schoof's algorithm had a polynomial running time and used Hasse's theorem on elliptic curves.

Theorem 1. *Hasse's Theorem If E is an elliptic curve over the finite field F_p then:*

$$|p + 1 - |E(F_p)|| \leq 2\sqrt{p}$$

If we define $t = p + 1 - |E(F_p)|$ we have to compute t *mod* N where $N > 4\sqrt{p}$. Schoof's algorithm computes this using small primes l_i where $\prod l_i = N$. After computing t *mod* l_i we can find t using the Chinese Remainder Theorem. Knowing t we can then compute $|E(F_p)| = p+1-t$. To compute t *mod* l Schoof used the Frobenius endomorphism ϕ and division polynomials.

Theorem 2. *Frobenius endomorphism The Frobenius endomorphism ϕ satisfies the following:*

$$\phi^2 - t\phi + p = 0 \quad where \quad t = p + 1 + |E(F_p)|$$

According to the Theorem 2 we have the equation:

$$\phi^2 P + p_l P = t_l \phi P \quad where \quad P(x,y) \in E(F_p)$$

Here $p_l = p$ *mod* l and $t_l = t$ *mod* l. If we restrict to nontrivial l-torsion points (a tortion subgroup consists of all the elements of an abelian group that have finite order) we obtain:

$$(x^{p^2}, y^{p^2}) + \overline{p}(x,y) = \overline{t}(x^p, y^p) \tag{2}$$

where $\overline{x}$ is an unique integer such that $x = \overline{x}$ *mod* l. The above equation is valid because in a l-tortion subgroup the scalar multiplication has the property $pG = \overline{p}G$. Starting from Equation (2) and applying division polynomials, Schoof's algorithm computes the value of $|E(F_p)|$. The reader can study the algorithm and its improvements made over time in [18].

Another algorithm based on Hasse's theorem was developed by D.Shanks [19]. The algorithm is named Baby Steps-Giant Steps and computes a number $m \in (p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p})$ such that $mG = O$ where G is a random point from the curve $E : y^2 = x^3 + ax + b$. The algorithm is described below:

(1) Compute $s \approx \sqrt[4]{p}$

(2) Compute $G, 2G \ldots sG$

(3) Compute $Q = (2s + 1)P$ and $R = (p + 1)P$

(4) Compute $R, R \pm Q, R \pm 2Q, \ldots R \pm tQ$ where $t = \left[\frac{2\sqrt{p}}{2s+1}\right]$

The first three steps are known as baby steps while computing $R, R \pm Q \ldots, R \pm tQ$ is the giant step. From Hasse's theorem we know that $R + iQ \;\; i = 0, \pm 1, \pm 2, \ldots, \pm t$ is equal with one from the points computed in second step. For this i we have:

$$R + iQ = jG \quad j \in \{0, \pm 1, \pm 2, \ldots, \pm t\}$$

The number m will be $m = p + 1 + (2s + 1)i - j$ which represents the cardinal of the elliptic curve points set. Variations, improvements and enhancements on this algorithm can be studied in [20]. A very important zero-knowledge protocol, which represents the basis for the most popular zero-knowledge protocols, is the Fiat-Shamir Identification Protocol. Important protocols derived from it are Feige-Fiat-Shamir [21] and Guillou-Quisquater. We chose it because it is the simplest protocol which illustrates the most important properties of the modern sophisticated schemes. This protocol is used in cryptography for authenticating a certain person. Suppose Alice has a secret Se known only by her. She will prove her identity to Bob by proving that she possesses Se, of course, without revealing the secret. Because the secret is not revealed to the verifier, no adversary can find it from the prover response. A trusted part is needed for this protocol which generates two secret prime numbers p and q, and computes the public value $n = pq$. The steps that follow this operation are repeated t times, each time using independent random numbers. If the verifier has repeated the steps t times then he accepts.

The algorithm is described below (see Algorithm 1) and the repeating steps begin with the fifth one. The first two steps are executed by the third trusted part, while the steps three and four are executed by the prover only one time each. The number t is chosen by the verifier, if the verifier is easy to convince, t can be smaller. A detailed explanation on this algorithm can be found in [22].

Algorithm 1 Fiat-Shamir Identification Protocol.

1: p and q are generated
2: $n = pq$ is made public
3: the prover selects Se co-prime to n such that $1 \leq Se \leq n - 1$
4: the prover computes $v = Se^2 \; mod \; n$ which is his public key
5: the prover chooses r such that $1 \leq r \leq n - 1$
6: the prover computes $x = r^2 \; mod \; n$ and sends it to the verifier
7: the verifier chooses a bit $e \in \{0, 1\}$ and sends it to the prover
8: **if** e=0 **then**
9: the prover computes $y = r$
10: **else**
11: the prover computes $y = rs \; mod \; n$
12: **end if**
13: the prover sends y to the verifier
14: the verifier rejects if $y = 0$ or $y^2 \neq x * v^e \; (mod \; n)$

For example $p = 5$ and $q = 11$ then $n = 55$ is made public. Suppose Alice (prover) chooses her secret $Se = 14$ and computes $v = 14^2 \; mod \; 55 = 31$. Bob is an easy to convince verifier and chose $t = 2$.

(1) Alice chose $r = 9$

(2) Alice sends $x = 9^2 \ mod \ 55 = 26$ to Bob

(3) Bob sends $e = 0$ to Alice

(4) Alice sends $y = r = 9$ to Bob

(5) Bob verifies $y \neq 0$ and $9^2 \ mod \ 55 = (26 * 31^0) \ mod \ 55 \Leftrightarrow 19 = 19$

(6) Alice chose $r = 15$

(7) Alice sends $x = 15^2 \ mod \ 55 = 5$ to Bob

(8) Bob sends $e = 1$ to Alice

(9) Alice sends $y = rs \ mod \ 55 = 45$ to Bob

(10) Bob verifies $y \neq 0$ and $45^2 \ mod \ 55 = (5 * 31^1) \ mod \ 55 \Leftrightarrow 45 = 45$

The completeness of this protocol is provided by the fact that the prover possessing the secret Se can also compute $y = r$ or $y = rs$ and send it to the verifier. Therefore, an honest verifier will always complete all t iterations and accept with the probability 1. To demonstrate the soundness we suppose the prover does not possess the secret Se. Therefore, on a given round he cannot compute $y = r$ or $y = rs$. Thus, the probability of rejection will be $\frac{1}{2}$ in each round. The zero-knowledge is provided by the fact that the only values made public in one round are x and y. A (x, y) pair can be simulated by choosing a random y and then computing $x = y^2$ or $x = \frac{y^2}{v}$. We can observe that such pairs are computationally indistinguishable from the ones computed in the protocol.

A "no-leak" zero-knowledge authentication was presented in [23]. Alice's (the prover) private key consists of:

(1) a subset $S_0 \subset S$ where S is an universal set

(2) an efficient test to verify if an element from S does not belong to S_0

(3) a method for distinguishing the subset S_0 to some S'_0

while the public key is the pair of sets S'_0, S_1 such that $S'_0 \cap S_1 = O$. The algorithm has three steps:

Algorithm 2 No-leak Authentication Protocol.

1: Bob sends $(x'_1, x'_2, \ldots x'_{2m})$ to Alice, where $x'_i \ \forall i$ is a random element from S'_0 or S_1, and exactly m elements belong to S'_0 and m to S_1.

2: Alice uses her private test to check whether for element x_i corresponding to x'_i does not belong to S_0, $x_i \notin S_0$. If the test fails, she supposes that $x_i \in S_0$ which means that $x'_i \in S'_0$. She counts how many $x_i \notin S_0$. If the number she obtains is not exactly m then the authentication failed. If she obtains m, she sends to Bob a string with "0" in places corresponding to $x'_i \in S'_0$ and 1 for $x'_i \notin S'_0$.

3: Bob compares Alice's result with the right value. If they are equal he accepts the authentication.

To prevent guessing the answer these three steps can be repeated a number of times like in the Fiat-Shamir scheme. The author emphasized that if $m = 20$ then the probability of guessing the answer in a round of three steps is less than $\frac{1}{10^6}$.

The authors also presented two particularized methods: a subset sum and polynomials. We will describe only the one based on polynomial equations. In this case Alice's private key is:

(1) a polynomial $h(x_1, x_2, \ldots, x_k)$ over Z

(2) a large prime p

(3) a constant $c \in Z$

while the public key consists of:

(1) a polynomial $f(x_1, x_2, \ldots, x_k) = (h(x_1, x_2, \ldots, x_k))^2 - c(mod\, p)$

(2) a random polynomial $g(x_1, x_2, \ldots, x_k)$ over Z which has the same monomials as f and the coefficients with the same magnitude as the ones of f.

We observe that for any $(x_1, x_2, \ldots, x_k) \in Z$ exists a $v \in Z$ such that $f(x_1, x_2, \ldots, x_k) + c = u^2(mod\, p)$. The following algorithm describes the steps for a single element:

Algorithm 3 No-leak Polynomial Authentication Protocol.

1: Bob chose random integers $(x_1, x_2, \ldots x_k)$ and plugs them with the same probability into either f or g. Bob sends the result, noted $b(x_1, x_2, \ldots x_k)$ to Alice.
2: Alice computes $a = b(x_1, x_2, \ldots x_k) + c(mod\, p)$. She verifies if a is a square modulo p. If not she sends "1" to Bob because $b(x_1, x_2, \ldots x_k) \neq f(x_1, x_2, \ldots x_k)$. If it is a square she sends "0" assuming that $b(x_1, x_2, \ldots x_k) = f(x_1, x_2, \ldots x_k)$.
3: Bob compares Alice's result with the right value. If they are equal he accepts the authentication.

For this particular method the authors also present some suggestions for the parameters and the keys:

(1) $3 \leq k \leq 5$

(2) $p = 2^t$ where t is a security parameter

(3) $2 \leq degree(h) \leq 3$

(4) the magnitude of f's coefficients at least $p/2$

(5) the integers $x_1, x_2, \ldots, x_k$ are generated uniformly randomly from the interval $[1, 2^{t/k}]$

3. Our Method

We propose a zero-knowledge authentication based on elliptic curves and on the algorithms described in the previous section. For the use of elliptic curves we have to declare the domain parameters. For a nonsupersingular elliptic curve E defined over a prime field the domain parameters will be $(F, p, a_E, b_E, G, n, h)$ where F_p is the prime field, a_E, b_E define the curve $E : y^2 = x^3 + a_E x + b_E$, $G \in E$ is a point of order n (this means that n is the smallest positive number for which $nG = O$), $h = |E(F_p)|/n$ is the co-factor. To meet the above conditions it is recommended for $|E(F_p)|$ to be prime or $|E(F_p)| = h \cdot n$ where n is a large prime and $h \in \{1, 2, 3, 4\}$ [13]. Not all these parameters are used in a zero-knowledge authentication but they are all used in an elliptic curve cryptosystem. Therefore, defining these parameters provides one less step in the encryption/decryption process which the two communicating parties will use after authentication.

The generalized method uses an universal set S of elliptic curves' points. S_0 represents the points from a specific elliptic curve E. S_0' are elements corresponding to the points from S_0, while S_1 is a set of points which do not belong to the elliptic curve E. The private key and the public one remain the same with the above specifications. The Algorithm 2 becomes:

Algorithm 4 No-leak Elliptic Curve Authentication Protocol.

1: Bob sends $(X_1', X_2', \dots X_{2m}')$ to Alice, where $X_i'\ \forall i$ is a random element from S_0' or S_1, and exactly m elements belong to S_0' and m to S_1.

2: Alice uses her private test to check whether for point X_i corresponding to X_i' does not belong to S_0, $X_i \notin S_0$. If the test fails, she supposes that $X_i \in S_0$ which means that $X_i' \in S_0'$. She counts how many $X_i \notin S_0$. If the number she obtains is not exactly m then the authentication failed. If she obtains m, she sends to Bob a string with "0" in places corresponding to $X_i' \in S_0'$ and 1 for $X_i' \notin S_0'$.

3: Bob compares Alice's result with the right value. If they are equal he accepts the authentication.

This algorithm represents the generalized method for elliptic curves. We also present a particularized method which replaces the polynomials from the Algorithm 3 with elliptic curve points. Here Alice's keys change:

- the private key contains:
  - (1) a tuple $(x_1 P, x_2 P, \dots x_k P)$ where $P \in E$ and x_i are random scalars
  - (2) a random point Q (replacing the constant c)
- the public key contains:
  - (1) a tuple $(x_1 M, x_2 M, \dots x_k M) = 2(x_1 P, x_2 P \dots x_k P) - Q$ where $M \in E$
  - (2) a random tuple $(x_1 N, x_2 N, \dots, x_k N)$ where $N \in E$

Using these keys the algorithm becomes:

Algorithm 5 No-leak Elliptic Curve Authentication Protocol.

1: Bob chose random integers $(x_1, x_2, \dots x_k)$ and plugs them with the same probability into either $(x_1M, x_2M, \dots x_kM)$ or $(x_1N, x_2N, \dots, x_kN)$. Bob sends the result, noted $(x_1R, x_2R, \dots x_kR)$ to Alice.

2: Alice computes $A = (x_1R, x_2R, \dots x_kR) + Q$. She verifies if A is a doubled point. If not she sends "1" to Bob because $(x_1R, x_2R, \dots x_kR) \neq (x_1M, x_2M, \dots x_kM)$. If it is a doubled point she sends "0" assuming that $(x_1R, x_2R, \dots x_kR) = (x_1M, x_2M, \dots x_kM)$.

3: Bob compares Alice's result with the right value. If they are equal he accepts the authentication.

The scalar multiplication for elliptic curve points can be done with various methods. To improve the efficiency of such an algorithm, we have to improve the scalar multiplication which represents the most complex operation applied to an elliptic curve point. One of the most popular methods for scalar multiplication was introduced by P. Montgomery in [24]. The main idea is to generate q such that $c + qp$ is a multiple of r. The values c, p and r are given, r being a power of 2. Another performance scalar multiplication method for prime fields was presented in [25] and uses the Frobenius endomorphism. Clavier and Jove presented in [26] a new idea to ease the computation of kP. They propose to define k as $k_1 + k_2$ where $k_1 = k - r$ and $k_2 = r$, r being a random integer. Therefore, kP becomes $k_1P + k_2P$. This idea is very usefully because the values of k_1P and k_2P can be computed simultaneously. This can be applied to almost all the algorithms for computing scalar multiplication. An improvement to this idea was given by Ciet in [27].

4. Conclusions

Our communication system is made up of two parts: the authentication and the process of communication itself. The communication part implies a cryptosystem for encrypting and decrypting the messages. These two parts can contain only classical methods, elliptic curve methods or a combination of the two. Using the same type of methods for both parts is more efficient mainly because some of the generated values of the authentication are also used in the second part. On the other hand, using different kind of methods implies generating different values for each part. The optimal situation occurs when there is no need to generate additional values in the second part. For the second part, the elliptic curve methods have proved to be the most adequate for encrypting and decrypting messages because they need shorter keys in order to provide the same performance and security level than the classical ones. For the authentication we recommend our method because it is less complicated and it needs less resources than using a classical method for the first part and an elliptic curve one for the second. The authentication process is accomplished by using a third trusted part. This third part has a very important double role: it is an impartial witness to the communication and it also provides the authentication and the keys needed in the second part for the cryptosystem used. All in all, authentication is the first step to an efficient and secure communication system, which can be accomplished by using our elliptic curve method.

Acknowledgments

The authors acknowledge the support through Grant of The Executive Council for Funding Higher Education, Research and Innovation, Romania-UEFISCDI, Project Type: Advanced Colaborative Research Projects - PCCA, Number 23/2014 (V. Rădulescu and N. Constantinescu) and University of Craiova, Romania, Project Type: Advance in Researche, Number 43C/2014 (N. Constantinescu).

Author Contributions

Equal contributions of each author. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Halevi, S.; Krawczyk, H. Public-key cryptography and password protocols. *ACM Trans. Inf. Syst. Secur.* **1999**, *2*, 230–268.
2. Bellovin, S.M.; Merritt, M. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA, 4–6 May 1992; pp.72–84.
3. Bellovin, S.M.; Merritt, M. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In Proceedings of the ACM Conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; pp. 244–250.
4. Jablon, D. Strong password-only authenticated key exchange. *ACM SIGCOMM Comput. Commun. Rev.* **1996**, *26*, 5–20.
5. Wu, T. The secure remote password protocol. In Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium, San Diego, CA, USA, 11–13 March 1998; pp. 97–111.
6. Mohr, A. *A Survey of Zero-Knowledge Proofs with Applications to Cryptography*; Research Report; Southern Illinois University: Carbondale, IL, USA, 2007.
7. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209.
8. Miller, V. Uses of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO '85*, Proceedings of CRYPTO '85, Santa Barbara, CA, USA, 18–22 August 1985; Williams, H.C., Ed.; Lecture Notes in Computer Science, Volume 218; Springer: Berlin/Heidelberg, Germany, 1986; pp. 417–426.
9. Buchmann, J.; Baier, H. Efficient Construction of Cryptographically Strong Elliptic Curves. In *Progress in Cryptology—INDOCRYPT 2000*, Proceedings of First International Conference in Cryptology in India, Calcutta, India, 10–13 December 2000; Roy, B., Okamoto, E., Eds.; Lecture Notes in Computer Science, Volume 1977; Springer: Berlin/Heidelberg, Germany, 2000; pp. 191–202.

10. Van Oorschot, P.C.; Wiener, M.J. Parallel Collision Search with Cryptanalytic Applications. *J. Cryptol.* **1999**, *12*, 1–8.
11. Smart, N.P. The Discrete Logarithm Problem on Elliptic Curves of Trace One. *J. Cryptol.* **1999**, *12*, 193–196.
12. Menezes, A.; Okamoto, T.; Vanstone, S. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. In Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing, New Orleans, LA, USA, 5–8 May 1991; pp. 80–90.
13. Constantinescu, N. *Criptografie*; Romanian Academy: Bucharest, Romania, 2009.
14. Obersnel, F.; Omari, P. Positive solutions of elliptic problems with locally oscillating nonlinearities. *J. Math. Anal. Appl.* **2006**, *323*, 913–929.
15. Njoku, F.I. Some remarks on the solvability of the nonlinear two-point boundary value problems. *J. Niger. Math. Soc.* **1991**, *10*, 83–98.
16. Fernandes, M.L.C.; Omari, P.; Zanolin, F. On the solvability of a semilinear two-point BVP around the first eigenvalue. *Differ. Integr. Equ.* **1989**, *2*, 63–79.
17. Schoof, R. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.* **1985**, *44*, 483–494.
18. Avanzi, R.M.; Cohen, H.; Doche, C.; Frey, G.; Lange, T.; Nguyen, K.; Vercauteren, F. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*; Cohen, H., Frey, G., Eds.; Chapman and Hall/CRC: London, UK, 2006.
19. Cohen, H. *A Course in Computational Algebraic Number Theory*; Graduate Texts in Mathematics, Volume 138; Springer-Verlag: Berlin/Heidelberg, Germany, 1993.
20. Coron, J.S.; Lefranc, D.; Poupard, G. A New Baby-Step Giant-Step Algorithm and Some Applications to Cryptanalysis. In *Cryptographic Hardware and Embedded Systems—CHES 2005*, Proceedings of 7th International Workshop, Edinburgh, UK, 29 August–1 September 2005; Rao, J.R., Sunar, B., Eds.; Lecture Notes in Computer Science, Volume 3659; Springer: Berlin/Heidelberg, Germany, 2005.
21. Feige, U.; Fiat, A.; Shamir, A. Zero knowledge proofs of identity. *J. Cryptol.* **1987**, *1*, 77–94.
22. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*, 5th ed.; Chapman and Hall/CRC: London, UK, 2001.
23. Grigoriev, D.; Shpilrain, V. No-leak Authentication by the Sherlockk Holmes Method. *Groups Complex. Cryptol.* **2012**, *4*, 177–189.
24. Montgomery, P.L. Modular Multiplication without Trial Division. *Math. Comput.* **1985**, *44*, 519–521.
25. Muller, V. Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two. *J. Cryptol.* **1998**, *11*, 219–234.
26. Clavier, C.; Joye, M. Universal exponentiation algorithm a first step towards provable SPA-resistance. In *Cryptographic Hardware and Embedded Systems—CHES '01*, Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems, Paris, France, 14–16 May 2001; Naccache, D., Paar, C., Eds.; Lecture Notes in Computer Science, Volume 2162; Springer: Berlin/Heidelberg, Germany, 2001; pp. 300–308.

27. Ciet, M. Aspects of Fast and Secure Arithmetics for Elliptic Curve Cryptography. Ph.D. Thesis, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium, 2003.