mathematics

MDPI

Article

Optimal Elliptic-Curve Subspaces for Applications in Double-Authenticated Requests in Mobile Distributed Data Mining

Daniel Ioan Hunyadi *,†, Oana-Adriana Ticleanu †and Nicolae Constantinescu †

Faculty of Science, Department of Mathematics and Informatics, Lucian Blaga University of Sibiu, 550012 Sibiu, Romania

* Correspondence: daniel.hunyadi@ulbsibiu.ro

† These authors contributed equally to this work.

Abstract: Mathematical models based on elliptic curves have been intensively studied since their applicability in data security systems was discovered. In this article, the authors describe the optimal way to select particular subspaces over which elliptic curves are defined, showing the applicability of these subspaces in secure data transfer. Access to large databases and analyses of the requests made to these databases are required daily by a variety of users, including legal entities. An attack on these communication systems causes violations in privacy and damage to/theft of data that can be worth EUR tens of billions annually. For requests made between computers, encryption methods can be used as these systems have adequate computing power and energy. For requests made from fixed and mobile systems, if the data are distributed heterogeneously, the computing power required to authenticate both the users and the answering entities determines the efficiency of the proposed solution. To address this limitation, our study proposes a double-authentication method based on particular elliptic-curve systems.

Keywords: elliptic-curve cryptography; double authentication; data mining

MSC: 11G05; 11G07; 14H52

Citation: Hunyadi, D.I.; Ticleanu, O.-A.; Constantinescu, N. Optimal Elliptic-Curve Subspaces for Applications in Double-Authenticated Requests in Mobile Distributed Data Mining. *Mathematics* **2023**, *11*, 122. https://doi.org/10.3390/math11010122

Academic Editors: Pasquale De Meo, Oliviu Matei and Weihua Xu

Received: 28 October 2022
Revised: 30 November 2022
Accepted: 22 December 2022
Published: 27 December 2022

1. Introduction

In order to ensure the confidentiality of a request querying a complex database, techniques to authenticate the initiator and the answering entity and to encrypt the communication channel have been developed. Authentication based on the RSA-type digital signature—the Rivest, Shamir, and Adleman-based algorithm—was initially used ([1]). However, the power required to compute the parameters involved in this process increased with an increase in the length of the encryption keys. For a fixed computing system (computers and servers), the required power is provided by the involved entities; therefore, in the event that a request is made from and to a fixed computing system, the required computing power for the authentication protocols and for the communication encryption is provided by the query requester(s) and by the systems that store these distributed databases. In contrast, for mobile systems, energy consumption becomes an essential limiting factor; to address this limitation, we adopted elliptic-curve cryptography (ECC) as the solution for processes that require user authentication (as in [2], or quantum type on [3]). In the particular case of querying complex databases consisting of structures that are stored on several heterogeneous systems, as well as on physical structures and software, existing studies have proposed implementable solutions for each subcategory of case studies. In this sense, new solutions or their optimizations have continuously been illustrated, and the vulnerabilities of existing solutions have been highlighted. For the queries mentioned above, in situations in which an intelligent answer is necessary, only data mining (*DM*)

techniques can be used. These intelligent answers are answers that actually provide an analysis of complex databases and algorithms and highlight certain aspects in their conclusions. More precisely, for situations where databases are heterogeneous complexes (that is, of different types and sizes), the data are stored heterogeneously (that is, on several types of computing systems, which can be personal computers, high-power computers, servers, and even mobile systems); we call these systems heterogeneous database systems (*HDBSs*). For HDBSs, an analysis does not yield only an individual answer—for example, finding not only some records that meet certain conditions but also some items with certain properties that are correlated with other items—and can provide a conclusion based on the correlation of several analyses comparing several databases. In situations where a query would be made from a mobile system to an *HDBS*-type system, the use of an algorithm from the RSA–C category would require a computing power that consumes more energy than is available.

The solution proposed by the authors comes from their experience and expertise gained from research projects and government contracts. The proposed method addresses the case where a query, which involves *DM*, needs to be made from a mobile system with an *HDBS*-type structure. A variant of this method was implemented in a functional system and used within an institutional entity.

2. Existing Solutions for Related Problems

In this section, we briefly describe the existing solutions for problems related to authentication models between systems that involve multiple authentications. The difference between cases solved by other solutions and the case treated by our proposed solution is also described.

Starting from the basic advantage of ECC-type communication security systems, solutions that implement authentication models have been developed, such as the one in [4,5]; communication confidentiality systems, such as those developed in [6–8]; and analysis models of their limits, also described in [9–12]. These models are based on an essential property of ECC systems, namely, the size difference of the cryptographic credentials used in the systems based on elliptic curves, compared with those of systems based on RSA-type models and their variants. This translates into the number of calculations required to create the cryptographic primitives as well as the transformation from plain text to encrypted text for messages that need to be protected against attacks in order to be transmitted on public communication channels.

Numerous studies on authentication and secure communication have been carried out, and more are underway, both within university research laboratories and within the research and development departments of large companies, to provide solutions regarding particular mathematical models that address concrete problems in applications of secure communication in the case of heterogeneous systems. The key models used can be classified into three categories, depending on the type of devices in which they are implemented:

1 Systems that are implemented for authentication and secured communications, in which the involved devices in the communication process are classified as servers and computers. In these cases, mathematical models and algorithms of the type described in [13,14] can be used. Particular variants of these solutions were implemented for this kind of model, by the authors, in such cases. For these models, the computing power required to compute the parameters involved in obtaining the cryptographic credentials can be assured by the devices in which they are implemented.
In the case studied by the authors, one of the restrictions is related to the computing power available for use in the process generating cryptographic credentials. Therefore, both the mathematical model used and the implemented algorithmic model must take this aspect into account.

2 Smart mobile system interconnections, in which unitary models are implemented to secure the communications. For this case, the computation of cryptographic credentials is ensured by a centralized system, and the mobile device uses the cryptographic

primitives provided by a trusted party that manages the solution in a centralized way. Then, the mobile device has to use these parameters. Such models are described in [15,16]. The solution from [15] was proposed by a team that included one of the authors from the present study. This solution was implemented and is still being used. In our studied case, the system is heterogeneous both from the point of view of the devices that are involved in the communication process and from the point of view of the security model for each of the groups of devices, with a solution being proposed for the case where the authentication will be carried out based on the parameters computed also by the mobile systems, with these devices generating their own cryptographic primitives.

3 Security models in which mobile devices are involved in the correlation with fixed systems, included in *SOTA* (Secure Online Transaction Algorithm)-type models. In these cases, three parties are involved: consumer, retailer, and financial credit company. Solutions in this sense can be studied in [17].
The solution proposed by the authors is for the case where the volume of data transmitted is much higher than that in the case described above and where it is necessary to reduce the risk of malicious users, which in the above case does not need to be treated.

3. Description of The Parameters from Our Proposed Solution

This section describes the elements that define the treated case, the limitations imposed by the problem that needs to be solved, and the method of hierarchization for all the entities of the treated system.

This study was carried out on an actual case in which an entity required a high degree of data security and communication confidentiality. We describe this case, first, by illustrating the components of this system. Let us take a heterogeneous set of computing systems, denoted as follows:

A set of
- n_S server-type computing systems, denoted by $\mathcal{S} = \{S_1, S_2, ..., S_{n_S}\}$;
- n_C computers, denoted by $\mathcal{C} = \{C_1, C_2, ..., C_{n_C}\}$;
- n_M mobile devices, denoted by $\mathcal{M} = \{M_1, M_2, ..., M_{n_M}\}$;
- n_P smart mobile phones, denoted by $\mathcal{P} = \{P_1, P_2, ..., P_{n_P}\}$.

The aforementioned devices contain databases, with temporary databases stored in $\mathcal{M}$ and $\mathcal{P}$, which means that the data will occasionally be transferred to specialized servers. The credentials required for the authentication and secure communications assurance systems are managed by a specialized device, called a trusted server (TS). The presented system is described in Figure 1.

All the above systems have ensured interconnections, but our proposed system involves different communication protocols, different types of physical communication, and different *uptimes*—the time during which they can be accessed. Let us denote the sets of communication types throughall systems as $\mathcal{IS}$.

Let $\mathcal{T} = \langle \{\mathcal{S}, \mathcal{C}, \mathcal{M}, \mathcal{P}, TS\}, \mathcal{IS} \rangle$. Then, $\mathcal{T}$ represents an $HDBS$.

Let P_i be a smart mobile phone that makes a query in $\mathcal{T}$, where the result that will be provided involves an analysis of the whole $\mathcal{T}$ using DM algorithms. The data that pass between P_i and $\mathcal{T}$ within the query request, as well as the result that $\mathcal{T}$ provides to P_i, represent a communication that requires double authentication and confidentiality of the communicated data; this is the treated case.

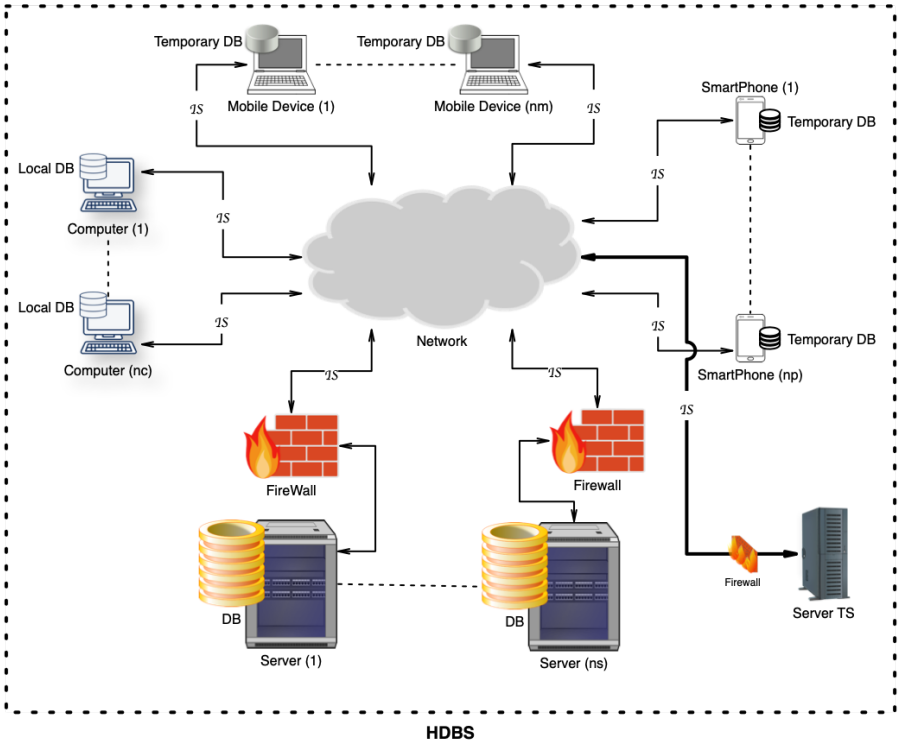

Figure 1. System description.

4. Our Solution to Ensure Authorized Access

This section presents the logical scheme of the studied system, the types of involved devices, the abstraction of the proposed model, and the proposed mathematical model for the generation of subspaces that are used in cryptographic processes.

In order to access $\mathcal{T}$'s query service, a complete security process is required, through which P_i proves its identity and the $\mathcal{T}$ system identifies itself as the one to which P_i wants to connect. This is the process of a double authentication: P_i towards $\mathcal{T}$ and $\mathcal{T}$ towards P_i.

For the double-authentication technique, classic RSA-type algorithms can be used; such implementations require key lengths of 2048 bits for the involved security requests and the generation of keys on all computer systems within $\mathcal{T}$ for the treated case. For $\mathcal{S}, \mathcal{C}$, and $\mathcal{M}$, this requirement can be fulfilled, but in the case of $\mathcal{P}$, the computing power required to generate the asymmetric keys of RSA involves the consumption of energy that is not available on such systems. To solve this requirement, which represents a class of data and communication security subproblems in which the confidentiality requirement is of a high degree and computing systems with low energy resources are involved, an ECC-based system can be used. First, we determine the cases of communication requests, and, depending on the type of devices involved, the mutual authentication method (double authentication) of the systems starts a data transfer process.

Involved devices:

- Case 1: sets from $\langle \mathcal{S}, \mathcal{C}, \mathcal{M} \rangle$ and TS;
- Case 2: $\langle \mathcal{P}, \{AOT\} \rangle$, where AOT represents any other device type, including $\mathcal{S}, \mathcal{C}, \mathcal{M}, \mathcal{P}$, and TS.

For Case 1, the solution for the authentication was described in [15] and implemented, and it is still functional within a governmental entity. Let us describe the solution for Case 2.

4.1. Involved Device Authentication

In order to construct various types of quantiles of subspaces over which elliptic curves with cryptographic properties are defined, several studies have been carried out, among which we recommend [18–23]. From the authors' previous studies for cases in which

such subspaces were created for communication between devices with low computing power, i.e., smartphones, among those that demonstrated, in practice, superior results using models based on nonsupersingular elliptic curves, we recommend references [15,24] to interested readers. In order to define the number of subspaces required to obtain credentials in the double-authentication process for devices within $\mathcal{T}$, we build a model necessary for the particular case studied.

4.2. Mathematical Model for the Proposed Solution

Starting from the general space described in [24], let the values be generated by a series of integrals based on the partitions defining subspaces $\xi(K_{s_i})$, as follows:

$$\left(\int_{\xi} \frac{d\varphi}{\sqrt{4\varphi^3 - \tau_2\varphi - \tau_3}}\right) \frac{1}{|\xi|} \tag{1}$$

where ξ represents the quantile considered from the curves series, K_{s_i} is a partition of integer values, and φ is the variable defined over the subspace $\xi(K_{s_i})$ and τ_2, τ_3 constants and represents the credential parameters of the series.

For each partition, we define its inverse as an elliptic curve from which points that have cryptographic properties in the sense of defining the credentials needed in the process of computing the keys involved in double authentication can be selected. To build this credential system, two constants, ν_1 and ν_2, as well as a periodic function defined over a set of real numbers, are selected. In the case considered in the current study, the Weierstrass equation for each ξ-partition is given by the following formula:

$$(\delta')^2 = \left(4\delta^3 - \nu_1\delta - \nu_2\right)_{\xi} \tag{2}$$

where ν_1 and ν_2 are constants, and $\nu_1, \nu_2 \in < \xi >$.

The two values δ, δ' define a point of this curve:

$$\left(\iota^2 = 4\varphi^3 - \nu_1\varphi - \nu_2\right)_{\xi} \tag{3}$$

for which the possibility of being considered as a point with cryptographic properties is computed. These subsets are defined for each partition ξ of the above described system, thus obtaining a series of elliptic curves from which the points that will be part of the cryptographic algorithm are chosen.

5. Security Study of The Proposed Model

In this section, the security conditions of the proposed model are described.

Theorem 1. *The security condition, required by an HDBS system, is that the inequality*

$$|\xi| > (n_S + n_C + n_M + n_P) \tag{4}$$

has to be fulfilled.

Proof of Theorem 1. If two devices are considered, let use denote them by $d_i, d_j \in < \mathcal{S}, \mathcal{C}, \mathcal{M}, \mathcal{P} >$, where $i \neq j$, with ξ_i and ξ_j, where the intersection of the partitions ξ_i and ξ_j is nonempty. According to [15,25], we have collisions in the process of establishing a cryptographic key required in step three of the communication protocols involved; therefore, $< \xi_i > \cap < \xi_j > = \emptyset$. □

This theorem highlights the fact that each quantile considered in the series of studied elliptic curves will need to be defined independently from the others, a fact that emerges from the graphic representation illustrated in Figure 2. The quantities have to be selected in such a way as to comply with the requirement $< \xi_i > \cap < \xi_j > = \emptyset$.

Definition 1. *The elliptic curve corresponding to partition ξ_i is represented by the sets of points of the form $(\phi, \iota) \in \mathbb{Z}_\mu \times \mathbb{Z}_\mu$ that meet the congruence conditions:*

$$\iota^2 \equiv \left(\left(\phi^3 + \nu_1\phi + \nu_2\right)(mod\ \mu)\right)_\xi \tag{5}$$

and $\iota^2 = \phi^3 + \nu_1\phi + \nu_2$, with $(\nu_1, \nu_2) \in \mathbb{Z}_\mu$, where μ is a prime number greater than (ξ^2) and the $4\nu_1^3 + 27\nu_2^2 \not\equiv 0(mod\ \mu)$ relation is satisfied, to which a special point $\mathcal{O}$ is added, called the point at infinity.

For the operations performed in order to compute the cryptographic parameters, using points selected from each quantile, optimizations of the standard computation methods are necessary for the involved operations, namely, for the specific addition of two such points and the multiplication of such a point by a scalar, in order to obtain the primary credentials. Among the optimal methods used in such operations, we can mention the studies in [26–28], where fundamental mathematical models with transformations of these modules and implementation optimizations in various private computing systems were described.

6. Implementation of the Proposed Model

This section illustrates the formulas used to compute the parameters of the subspaces over which the elliptic curves are defined, the graphic representation of these subspaces, and the computation algorithm proposed to compute these parameters.

For the implementations considered in the present case, we consider the optimal method of the type $\phi_3 = \gamma^2 - \phi_1 - \phi_2$, $\iota_3 = \gamma(\phi_1 - \phi_3) - \iota_1$, where

$$\gamma = \begin{cases} (\iota_2 - \iota_1)(\phi_2 - \phi_1)^{-1}, for\ the\ case\ that\ the\ points\ in\ the\ operand\ differ \\ (3\phi_1^2 + a)(2\iota_1)^{-1}, when\ we\ have\ identical\ points \end{cases} \tag{6}$$

These operations are used to compute the intermediate points Q of coordinates (ϕ_3, ι_3), using points P_1 of coordinates (ϕ_1, ι_1) and P_2 of coordinates (ϕ_2, ι_2). In the case of this type of computation, it is necessary to consider compliance with the restrictions established for each partition defined by ξ_i.

For each partition ξ_i, the number of points on the elliptic subcurve, defined over the corresponding space of this partition, can be computed by computing the trace of Frobenius for the particular studied case. Let χ_η be a subset of K_{s_i}, where η represents the corresponding subsets for partition s_i. Thus, we have this value defined as nop in the form $nop = \#\mathcal{E}(\chi_\eta)_\xi = \eta + 1 - \kappa$, so for a space of size n, meaning $|\xi| = n$, it becomes $nop = \sum_{i=1}^{n}\left((\eta_\xi + 1) - \kappa_\xi\right)$, where η_ξ represents a prime integer for partition ξ and κ_ξ represents the number of points from partition ξ.

Starting from the form described in [15], for the case of partitioning the parameter space with cryptographic properties for each partition, an endomorphism of the following form is defined:

$$\zeta = \begin{cases} \mathcal{E}(\chi_\eta) \to \mathcal{E}(\overline{\chi_\eta}) \\ (\phi, \iota) \to (\phi^\eta, \iota^\eta) \\ \mathcal{O} \to \mathcal{O} \end{cases} \tag{7}$$

For each partition ξ_i, with $1 \leqslant i \leqslant n$, we have

$$\left|\kappa_\xi\right| \leqslant 2\sqrt{\eta_\xi} \tag{8}$$

for each device involved in the double-authentication process.

There is the possibility to compute the number of points with cryptographic properties from the total set of points on these subspaces, according to the size of the partition and the chosen starting point in the process of computing a pair of primary credentials.

6.1. Graphic Representation

In this section, we provide graphic representations of the series of elliptic curves from which the ξ_i subspaces are chosen and a representation of a point model with cryptographic properties resulting from the ESG algorithm.

In this regards, in Figure 2, a series of curves generated according to predefined K_{s_i} subspaces is represented and the choice of subspaces is made in compliance with the conditions from Theorem 1.

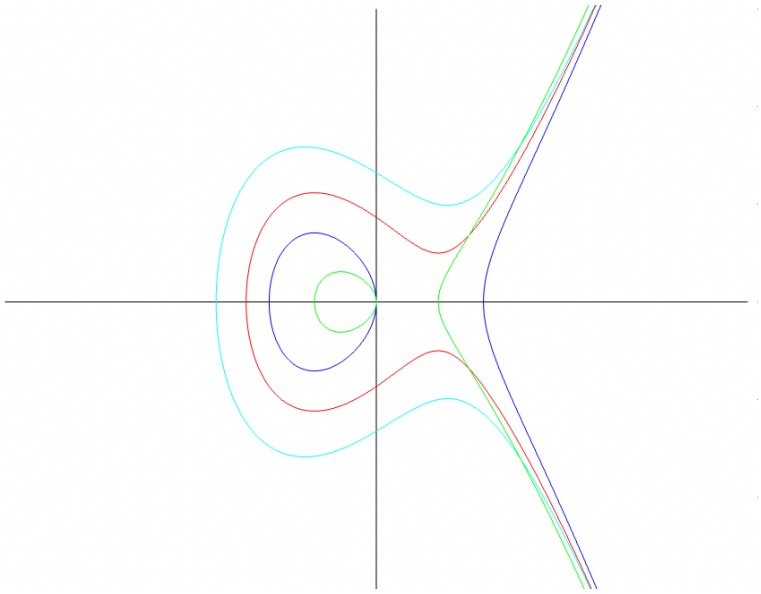

Figure 2. Series of elliptic curves.

In Figure 3, three types of points are presented: green, representing those that passed the ESG algorithm test, from step 4, and that have cryptographic properties; gray, those that passed the test at the returned step according to the minimum standard ξ_l^2 value imposed by the security policies; and red, those that did not pass the test at step 5. They represent a limited number from possible sets and are computed for an interval within the subspaces within d_i; P_j defines their credentials within the algorithm.

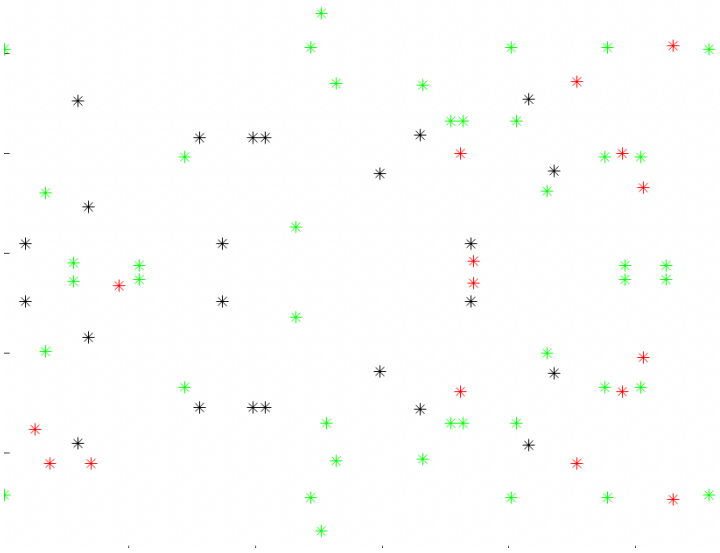

Figure 3. Cryptographic point detection.

6.2. ESG Algorithm. Cryptographic Parameter Computation

In this section, we describe our proposed method to compute the cryptographic parameters for each partition ξ_i using an $HDBS$-type device.

Each of the devices involved in the communication process initially select optimal parameters that will be used in the double authentication.

6.3. Double-Authentication Method

In this section, the proposed algorithm for the cryptographic parameter computation necessary for double authentication and for agreement with the communication security credentials is presented.

Algorithm 1 is the method by which two devices authenticate each other in the case where at least one device is from $\mathcal{P}$.

Algorithm 1: Subspace generation.

Initializations: $\mathcal{E}_j^{prec}$ is initialized at $\mathcal{O}$.

1. A partition ξ_l is chosen for a value l generated pseudorandomly, and $\mathcal{E}_j^{current}$ is initialized at $\mathcal{O}$.

2. $\mathcal{E}_j$ is built based on χ_η coefficients.

3. An approximation of the number of points with cryptographic properties is computed, according to Equation (8), and $\mathcal{E}_j^{current}$ is created accordingly.

4. The degree of correlation between the parameters of the points on the elliptic curve is checked to verify the degree of resistance to cryptographic attacks, according to the methodology of [29,30]. In case this degree is greater than $\epsilon_{\xi_l^1}$, certified according to the security policies established for $HDBS$, $max(\mathcal{E}_j^{prec}, \mathcal{E}_j^{current})$ is retained in $\mathcal{E}_j^{prec}$, and it returns to step 1. In other case, the current parameters are retained, in $\mathcal{E}_j^{current}$

5. The simulation of the cryptographic computation of the primary parameters is carried out according to Equation (6). If the ξ_l^2 value is exceeded, the maximum accepted value for devices in $< \mathcal{P} >$, then it returns to step 1. Otherwise, the current parameters are retained in $\mathcal{E}_j^{current}$

6. If the system returns a failure after testing the entire allocated partition, $\mathcal{E}_j^{prec}$ is returned together with a risk parameter allocated to this device. If the algorithm ends successfully, $\mathcal{E}_j^{current}$ is returned.

All the communication processes that can be performed within an $HDBS$ can be reduced to several communications, including this particular type, because any multiple communication required in multiple query processes within data mining operations can be divided into a certain number of such dual communication processes, with the application of the protocol described by the algorithm in the following part.

Let P_j be a device from $< \mathcal{P} >$, and let d_i be another device from $< \mathcal{S}, \mathcal{C}, \mathcal{M}, \mathcal{P} >$. Let it be the case where the d_i device initiates the connection (the same procedure is followed in the opposite case). Device d_i is checked to determineif it is in Case 1 or Case 2 according to the description of the possible cases in List 4. For Case 1, the protocols described in [15] are applied.

In Case 2, the procedure is as follows: device d_j uses in the double-authentication process its own pair of keys:

$$\begin{cases} \left((d_{pb_1}^{\xi_{di}}, d_{pb_2}^{\xi_{di}}), d_{pr}^{\xi_{di}} \right) \\ \left(P_{pb_1}^{\xi_{Pj}}, P_{pb_2}^{\xi_{Pj}} \right) \end{cases} \tag{9}$$

obtained from TS, and P_j uses the key pair

$$\begin{cases} \left((P_{pb_1}^{\xi_{Pj}}, d_{pb_2}^{\xi_{Pj}}), d_{pr}^{\xi_{Pj}} \right) \\ \left(d_{pb_1}^{\xi_{di}}, d_{pb_2}^{\xi_{di}} \right) \end{cases} \tag{10}$$

obtained from TS.

The initiator of the process generates a pair of pseudorandom values, which are used in the combination process according to its partition and will initiate the connection *Challenge*. Upon such a request, the mobile device generates a pair of pseudorandom values that are combined according to the composition algorithm for mobile devices.

After this, the mobile device initiates the *Response* process, through which it transmits the primary credentials necessary in the intermediate process required in the authentication of P_j by d_i.

In order to perform a double authentication, a new round of communications is initiated from d_i to P_j, called *CredentialAuthentication*, through which the parameters computed in the previous rounds by each of the participants are used as primary input data.

On this basis, at the end of the three rounds of communication, the two devices involved in the protocol will have completed the double-authentication process, through which each proved to the other its identity within the $\mathcal{T}$ system. A representation of this process is briefly shown in Figure 4.

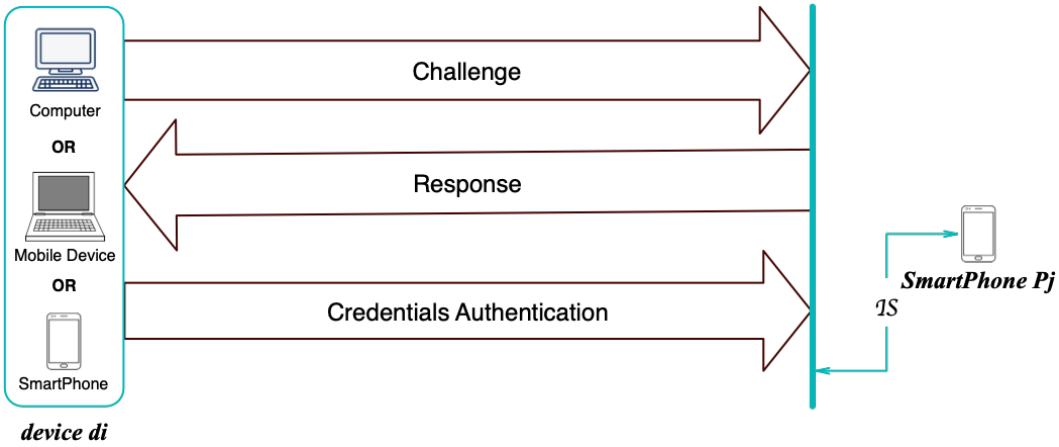

Figure 4. System description.

6.4. DWA Authentication Algorithm

In this section, in accordance with the steps presented in Section 6.3, we describe the proposed protocol, presented as the following Algorithm 2:

During the creation of the credentials used to secure the communication, d_i participates in the process using the data created by itself, namely, $Comp(\Delta_1, \Delta_2)$, and P_j participates in the construction of the credentials to secure the communication with the data generated independently of d_i, namely, $Comp(\mu_1, \mu_2)$. On this basis, the final credentials are computed according to the combination of the data independently generated by each device involved in this process, which results in agreement with the credential protocol, not imposed credentials. Through this process of creating these credentials, there are also ensured authentications of P_j by d_i, as well as of d_i by P_j, which means that the man-in-the-middle attack is not possible.

Furthermore, the algorithm creates authenticated credentials for secure communication processes that take place between devices within a repeated query used to create a final answer given by a data mining action on $HDBS$. This procedure creates the credentials and takes into consideration a method of ensuring that cryptographic parameters are generated according to the parameters specific to each category of devices within $\mathcal{T}$. In this whole set of procedures initiated by elements from $\mathcal{T}$, it will be necessary for TS to manage the

certificates of each device from $\mathcal{T}$. Within the information stored about devices from $< \mathcal{M}$ and $\mathcal{P} >$, it is necessary to have descriptive data about each device from each class of devices and at each connection; for a device that contains an empty *TemporaryDB*, it is first transferred to the database servers within *HDBS*, and only then will the query continue.

The security level of the algorithm presented is in accordance with the attack resistance of a model based on *ECC*, with the difference being that, for a possible attacker, the analysis must be conducted on $< \xi >$, which represents the composition of all the parts stored by the models from each device, and not on ξ_i, a fact that increases the difficulty of processing the parameters and therefore increases the security level of the system in the case of a real attack on the protection model described for mobile devices within $\mathcal{P}$.

Algorithm 2: DWA.

♢ d_i initiates the *Challenge* process.

1. $(\phi_1, \phi_2) \in \xi_{di}$ are generated, and $Combine(\phi_1, \phi_2)$ is called. τ_{di} is returned.
2. The parameter $\Delta_1 = \tau_{di}\left(\left(d_{pb_1}^{\xi_{di}}\right)^{-1} + d_{pb_2}^{\xi_{di}}\right)\frac{\phi_1}{\phi_2}$ is computed.
3. Δ_2 is computed as the hash of the composition $Comp\left(d_{pb_1}^{\xi_{di}}, \Delta_1\right)$.
4. Δ_3 is computed as the asymmetric encryption of the message Δ_2 using the secret key $d_{pr}^{\xi_{di}}$.
5. The $Comp(\Delta_1, \Delta_2)$ values are communicated to P_j (the *Challenge* process).

♢ Upon receiving the Δ_3 message, the P_j device initiates the *Response* process; generates two pseudorandom values $(\chi_1, \chi_2) \in (\xi_P)_j$; then calls the $Combine(P_{pb_1}^{\xi_{Pj}}, \Delta_1)$ function, which respects the entropy principles from ([31,32]); computes $\beta_1^{P_j}$ as the hash of the result given by the $Comp(P_{pb_1}^{\xi_{Pj}}, \Delta_1)$ function; and computes $\beta_2^{P_j}$ to be the encryption of Δ_2 using $d_{pb_1}^{\xi_{di}}$'s public key. If the two computed parameters are not identical, the authentication of d_i by P_j ends in failure. Otherwise, the construction of the cryptographic parameters continues.

6. P_j computes $\mu_1 = \tau_{P_j}\left(\left(P_{pb_1}^{\xi_{Pj}}\right)^{-1} + P_{pb_2}^{\xi_{Pj}}\right)\frac{\chi_1}{chi_2}$.
7. P_j computes μ_2 as the hash of $Comp(P_{pb_1}^{\xi_{Pj}}, \mu_1)$.
8. μ_3 is computed as the asymmetric encryption of the message μ_2 using the secret key $P_{pr}^{\xi_{Pj}}$.

♢ The *Response* process consists of P_j sending $Comp(\mu_1, \mu_2)$ values. Upon receiving the message from P_j, the device d_i verifies the identity of d_i in the same way as d_i verified the identity of P_j. If the verification fails, the protocol ends with a failure. Otherwise, the protocol initiates the process for the *Credential Authentication* phase.

9. d_i computes $\beta_1^{P_j}$ as the hash of the result returned by the $Comp(P_{pb_1}^{\xi_{Pj}}, \mu_1)$ function.
10. d_i computes $\beta_2^{P_j}$ to be μ_2's encryption using $P_{pb_1}^{\xi_{Pj}}$'s public key. If the two calculated parameters do not have the same value, the credential authentication step fails. If the two parameters have the same value, the process ends successfully by authenticating the credentials approved by the two parameters and proceeds to calculate the final credentials of secure communication: d_i computes a parameter Δ_4 as the hash applied to the first three calculated parameters, encrypted with its public key, and transmits the result to P_j.

6.5. Performance Analysis

In this section, the performance of the implemented model (ESG) is described in comparison with two models previously implemented by the beneficiary, namely, $GN1$ (a variant based on solution presented in [15]) and KMK (a variant based on solution presented in [33]), which were solutions that contributed to the dual authentication system, with the common feature being the space over which elliptic curves are defined, for an environment which contains 3 server centers, 87 users from the $\mathcal{C}$ category, and 212 users from the class $\mathcal{M}$, with heterogeneous systems.

The presented model was analyzed according to the computation time required to generate $\mathcal{E}_j^{prec}$-type subspaces for the presented solution, and the time generation and communication time of the results for previous solutions (in Table 1).

Table 1. Parameter generation.

	***GN*1**	***KMK***	***ESG***
Generation time	1.17	2.11	1.8
Assignment time	4.15	4.15	0.3
Recomputations	27	0	41
Generation errors	129	128	215

Below are descriptions of the properties represented in the table, as well as some remarks on the results.

Generation time —the time (in seconds) required to compute the parameters for elliptic curves. The time to compute the parameters is correlated with power consumption.

Assignment time —the time (in seconds) required to transfer data to other devices that are involved in the communication. The assignment time is correlated with power consumption.

Recomputations —the number of parameter recalculations (at each 1000 calls) according to the security policies established by the beneficiary. It should be noted here that they change over time depending on the problems found during the periodic security audit. The recomputation time is correlated with power consumption.

Generation errors —the number of failures (at each 1000 calls) in the process of generating a subspace in a space.

Moreover, a performance analysis with the required computation time was performed every 6 months to establish the cryptographic credentials and the number of detected attacks on the system (the last analysis is presented in Table 2).

Table 2. Credential generation.

	***GN*1**	***KMK***	***ESG***
Cryptographic credential generation time	0.72	1.12	0.3
Detected attacks	6	19	25

The data from this table are explained and interpreted below.

Cryptographic credential generation time —the time required to generate the cryptographic credentials used in double authentication and to secure the communication. The cryptographic credential generation time is correlated with power consumption.

Detected attacks —the number of detected attacks (at each 1000 calls).

As a general statement, the method of computing the cryptographic parameters, from the point of view of the allocated times/power consumption, depends on the initial generation method of the involved parameters and the security policy established by the beneficiary, which influences the number of recomputations and the generation time of the cryptographic parameters. This decision is made depending on the degree of security that is requested to be implemented.

6.6. Limitations of the Proposed Model

The proposed model ensures the creation of the necessary credentials in the cryptographic processes involved in securing communications and double authentication; otherwise, communications will be disallowed. This main limitation of the model is due to step 5 of Algorithm 1, where a $\tilde{\varsigma}_l^2$ value is exceeded only if it meets the maximum conditions for this parameter, established as the maximum value accepted for the devices in $< \mathcal{P} >$, according to the security policies, so the establishment of the security policies determines the number of repetitions of the process in certain cases. This practically translates into longer creation times for cryptographic credentials.

The second limitation involves the way in which double authentication is performed and the protection against man-in-the-middle attacks. This is present in all systems that respect this principle, namely, the creation of common cryptographic credentials is ensured, and in the case of an attack of the type mentioned above, communication is not allowed; the presence of such an attack can thus be signaled, but communication is not allowed.

7. Conclusions and Future Work

In this paper, we presented a method to determine the cryptographic parameters in the case of queries in an *HDBS*-type system; more precisely, we proposed a secure communication system for the case where smartphone-type devices are involved in this process and other heterogeneous devices are implied. The solution takes into account the particularities of these devices in terms of computing power and describes a way to generate cryptographic credentials for communications involving data-mining-type queries, as well as the transfer of base models defined as being of the *TemporaryDB* type. This system was implemented and is functional within an entity that uses these types of queries. During the implementation and after consultation with the users of the system, as future research, an estimation of the degree of risk for the subspaces that are used and the establishment of methods to determine optimal $\epsilon_{\tilde{\varsigma}_l^1}$ and $\epsilon_{\tilde{\varsigma}_l^2}$ parameters within security policies were requested. These represent the next steps in the research to be undertaken.

Author Contributions: N.C. investigated the ideas, formal analysis and review; O.-A.T. implemented the method and wrote the original draft of manuscript; D.I.H. provided the conceptualization, validation of the method, resources and funding support; O.-A.T. and D.I.H. revised the draft of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Lucian Blaga University of Sibiu through the research grant LBUS-IRG-2022-08.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RSA–C	Rivest, Shamir, and Adleman cryptography
ECC	Elliptic-curve cryptography
DM	Data mining
HDBS	Heterogeneous database system
TS	Trusted server

References

1. Halevi, S.; Krawczyk, H. Public-key cryptography and password protocols. *ACM Trans. Inf. Syst. Secur.* **1999**, *2*, 230–268. [CrossRef]
2. Kumari, A.; Kapoor, V. Competing secure text encryption in intranet using elliptic curve cryptography. *J. Discret. Math. Sci. Cryptogr.* **2020**, *23*, 631–641. [CrossRef]
3. Sudharson, K.; Arun, S. Security Protocol Function Using Quantum Elliptic Curve Cryptography Algorithm. *Intell. Autom. Soft Comput.* **2022**, *34*, 1769–1784. [CrossRef]
4. Mehibel, N.; Hamadouche, M.H. A new enhancement of elliptic curve digital signature algorithm. *J. Discret. Math. Sci. Cryptogr.* **2020**, *23*, 743–757. [CrossRef]
5. Grigoriev, D.; Shpilrain, V. No-leak Authentication by the Sherlockk Holmes Method. *Groups Complex. Cryptol.* **2012**, *4*, 177–189. [CrossRef]
6. Subramanian, E.K.; Tamilselvan, L. Elliptic curve Diffie-Hellman cryptosystem in big data cloud security. *Clust.-Comput.- J. Netw. Softw. Tools Appl.* **2020**, *23*, 3057–3067 [CrossRef]
7. Alimoradi, R.; Arkian, H.R.; Razavian, S.M.J.; Ramzi, A. Seiied-Mohammad-Javad and Ramzi, Ali. Scalar multiplication in elliptic curve libraries. *J. Discret. Math. Sci. Cryptogr.* **2021**, *24*, 657–666. [CrossRef]
8. Gun, S.; Murty, V.K. Lifting of Elliptic Curves. *Pac. J. Math.* **2019**, *301*, 101–106. [CrossRef]
9. Nitaj, A.; Susilo, W.; Tonien, J. Improved Cryptanalysis of the KMOV Elliptic Curve Cryptosystem. In Proceedings of the 13th International Conference on Provable and Practical Security (ProvSec), Cairns, Australia, 1–4 October 2019.
10. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*, 5th ed.; Chapman and Hall/CRC: London, UK, 2001.
11. Martínez, V.G.; Encinas, L.H.; Muñoz, A.M.; Díaz, R.D. Secure elliptic curves and their performance. *Log. J. Igpl* **2019**, *27*, 227–238.
12. Bennett, M.; Gherga, A.; Rechnitzer, A. Computing elliptic curves over Q. *Math. Comput.* **2019**, *88*, 1341–1390. [CrossRef]
13. Faz-Hernandez, A.; Lopez, J.; Dahab, R. High-performance Implementation of Elliptic Curve Cryptography Using Vector Instructions. *ACM Trans. Math. Softw.* **2019**, *45*, 25–37. [CrossRef]
14. Barański, M.; Gliwa, R.; Szmidt, J. Cryptographically Strong Elliptic Curves of Prime Order. *Int. J. Electron. Telecommun.* **2021**, *67*, 207–212.
15. Stephanides, G.; Constantinescu, N. The GN-authenticated key agreement. *Appl. Math. Comput.* **2005**, *170*, 531–544. [CrossRef]
16. Caruso, X.; Eid, E.; Lercier, R. Fast computation of elliptic curve isogenies in characteristic two. *J. Lond. Math.-Soc.-Second. Ser.* **2021**, *104*, 1901–1929. [CrossRef]
17. Gualdoni, J.; Kurtz, A.; Myzyri, I.; Wheeler, M.; Rizvi, S. Secure Online Transaction Algorithm: Securing Online Transaction Using Two-Factor Authentication. *Procedia Comput. Sci.* **2017**, *114*, 93–99. [CrossRef]
18. 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.
19. Abdaoui, A.; Erbad, A.; Al-Ali, A.K.; Mohamed, A.; Guizani, M. Fuzzy Elliptic Curve Cryptography for Authentication in Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 9987–9998. [CrossRef]
20. Baashirah, R.; Abuzneid, A.; Mellouki, S.A.; Siraj, Z.; Zhan, C. Low-Cost RFID Authentication Protocol Based on Elliptic Curve Algorithm. *Int. J. Interdiscip. Telecommun. Netw.* **2021**, *13*, 1–11. [CrossRef]
21. Saudy, N.F.; Ali, I.A.; Al Barkouky, R. Error analysis and detection procedures for elliptic curve cryptography. *Ain Shams Eng. J.* **2019**, *10*, 587–597. [CrossRef]
22. Aka, M.; Luethi, M.; Michel, P.; Wieser, A. Simultaneous supersingular reductions of CM elliptic curves. *J. Fur Die Reine Und Angew. Math.* **2022**, *786*, 1–43. [CrossRef]
23. Ciet, M. Aspects of Fast and Secure Arithmetics for Elliptic Curve Cryptography. Ph.D. Thesis, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium, 2003.
24. Constantinescu, N.; Non Singular Elliptic Curves—From Theory to Application. Algorithm Attacks Discussions. *Mathematica* **2008**, *50*, 177–186.
25. Gupta, V.; Acu, A.M.; Srivastava, H.M. Difference of Some Positive Linear Approximation Operators for Higher-Order Derivatives. *Symmetry* **2020**, *12*, 915. [CrossRef]
26. Wu, T.; Wang, R. Fast unified elliptic curve point multiplication for NIST prime curves on FPGAs. *J. Cryptogr. Eng.* **2019**, *9*, 401–410. [CrossRef]
27. Smart, N.P. The Discrete Logarithm Problem on Elliptic Curves of Trace One. *J. Cryptol.* **1999**, *12*, 193–196. [CrossRef]
28. Aljamaly, K.T.R.; Ajeena, R.K.K. The elliptic scalar multiplication graph and its application in elliptic curve cryptography. *J. Discret. Math. Sci. Cryptogr.* **2021**, *24*, 1793–1807. [CrossRef]
29. 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; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3659.
30. Hashimoto, Y.; Takashima, K. Improved supersingularity testing of elliptic curves. *Jsiam Lett.* **2021**, *13*, 29–32. [CrossRef]
31. Acu, A.M.; Maduta, A.; Otrocol, D.; Rasa, I. Inequalities for Information Potentials and Entropies. *Mathematics* **2020**, *8*, 2056. [CrossRef]

32. Acu, A.M.; Hodis, S.; Rasa, I. Estimates for the Differences of Certain Positive Linear Operators. *Mathematics* **2020**, *8*, 798. [CrossRef]
33. Kumari, A.; Abbasi, M.Y.; Kumar, V.; Khan, A.A. A secure user authentication protocol using elliptic curve cryptography. *J. Discret. Math. Sci. Cryptogr.* **2019**, *22*, 521–530. [CrossRef]