An Enhanced Lightweight Authentication Scheme for Secure Access to Cloud Data

Hamza Hammami[1], Mohammad S. Obaidat[2,3,4,*] and Sadok Ben Yahia[5,6]

[1]*Hamza Hammami, University of Tunis El Manar, Faculty of Sciences of Tunis, LIPAH-LR11ES14, 2092 Tunis, Tunisia*
[2]*Dean & Professor, College of Computing & Informatics University of Sharjah, U.A.E.*
[3]*King Abdullah II School of Information Technology, Universality of Jordan, Jordan*
[4]*University of Science and Technology Beijing, China*
[5]*University of Tunis El Manar, Faculty of Sciences of Tunis, LIPAH-LR11ES14, 2092 Tunis, Tunisia*
[6]*Tallinn University of Technology, Department of Software Science, Akadeemia tee 15a, Tallinn 12618, Estonia*

Keywords: Cloud Computing, Storage as a Service, Security, Authentication, Hybrid Cryptography, Costs, Confidentiality, Integrity, Attacks.

Abstract: The use of cloud computing has become increasingly important due to many factors, including the cost-effective architecture that supports data transmission, storage and computation. It has become indispensable to setting up and providing IT services. Among these services, outsourced data storage, or Storage as a Service (StaaS), which is one of the most popular services in cloud computing; it reliably stores large volumes of data. In return, apart from its benefits in terms of cost and ease of management, StaaS poses new problems related to the security of data and their treatments during access. This is due to the storage of data at a distance beyond the perimeters of users and the involvement of one or more third parties such as service providers or infrastructure. Indeed, the provision of sensitive data to an external entity is a serious concern. The major issues of security, privacy and trust remain the main concerns that hamper the mass adoption of the cloud. Therefore, an automatic focus when using cloud services is the presence of a good strong authentication mechanism to properly authenticate users and mitigate as many vulnerabilities as possible. Our work is part of the research theme on security challenges including the protection of personal data during the authentication process, posed in cloud environments. With this in mind, we introduce an authentication mechanism that takes advantage of the opportunities offered by the hybrid cryptography techniques to protect each user's personal data in the cloud environment while preserving its privacy. The experiments show that the authentication mechanism, we offer, surpasses its competitors in terms of communication and computational costs, data confidentiality and integrity, and resistance to various types of attacks.

1 INTRODUCTION

Cloud computing enables the on-demand delivery of computing power, database storage, and other computing resources through an Internet cloud service platform with progressive pricing. This is a new model that gives rise to three main types, commonly known as IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) (Marinescu, 2017). It provides a simple way to access servers, storage and databases, as well as a wide range of application services over the Internet. Cloud computing platforms own and maintain the networked hardware required for these application services. Users allocate and only utilize the resources they need through a web application (Vasiljeva et al., 2017).

Indeed, it is now possible to provide any business with what it needs, to the right extent and from a single point of entry. This change is a great opportunity to improve the quality of services and focus on changing the essentials business. Improved networking and virtualization technologies can now free the business from infrastructure management to only focus on their own business.

It cannot be denied that cloud computing is a great

[*] Fellow of IEEE

Hammami, H., Obaidat, M. and Ben Yahia, S.
An Enhanced Lightweight Authentication Scheme for Secure Access to Cloud Data.
DOI: 10.5220/0009824301100117
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - SECRYPT*, pages 110-117
ISBN: 978-989-758-446-6

solution that offers a wide range of benefits to users and businesses alike. Nonetheless, the security of losing control over certain personal data, disclosing sensitive data and protecting identity within the cloud is very delicate (Kumar et al., 2018) (Li et al., 2018).

Indeed, the secure management of the identities of cloud users is essential because it takes a more complex dimension for the cloud because it is more difficult to manage users remotely. In addition, the access of a cloud service provider to identification information presents a problem from the point of view of protecting personal data. There is also a lack of transparency in the cloud preventing users from monitoring their own personal information, which leads to a lack of trust. Users care about: who controls their authentication data in the cloud and who owns the rights administrative to access it (Djellalbia et al., 2016).

Because the identity of the user and sensitive information are regularly utilized in the authentication process when accessing cloud services, it becomes important to protect them (Djellalbia et al., 2016). In other words, how can we ensure a completely secure authentication?

It has therefore become of utmost importance to ensure the security of the personal data of remote users in order to guarantee greater security. Our solution relates to this problem. It provides a strong access mechanism to properly authenticate remote users to access their data stored in the cloud and minimize the risk of disclosure.

Indeed, we have developed a reliable and robust mechanism based mainly on hybrid cryptography techniques, the goal being to preserve the privacy of cloud users through the protection of their identities and their various communications. Our main motivation is to provide a robust authentication mechanism whose purpose is to ensure the optimal privacy for cloud users. Privacy is considered as a key element. The authentication offered by our mechanism guarantees the consumption of cloud services on demand. Therefore, our approach will ensure the preservation of the privacy of its users in terms of protection of their personal data, including identity. They can access and consume the services anonymously and legitimately without the cloud service provider knowing their identity. A series of experiments highlights the effectiveness and reliability of our proposed security scheme compared to those proposed in the literature in terms of security services provided and costs of communication and computation.

The paper is organized as follows: Section 2 presents a scrutiny of literature research on the security approaches proposed to ensure data security, in particular the management of cloud data access. It presents as well a comparative study of these approaches from which we try to inspire ourselves to develop our contributions in this area. Section 3 describes in detail the solution we propose and its different steps. Section 4 shows the performance and reliability of the suggested solution in terms of computational and communication costs, data confidentiality and integrity. It also describes the robustness of our solution against various types of attacks. Finally, Section 5 presents a general conclusion in which an evaluation of the developed work is carried out and perspectives for future work are provided.

2 STATE OF THE ART

In this section, we firstly review some related recent works to address issues related to data access security and the processing of data during the migration to the cloud for remote storage at data centers of cloud providers. Secondly, we carry out a comparative study to classify and address critics of all the protection systems mentioned in this study. As stated earlier, when data are placed in the cloud, the security of their access is critical. Since these data are no longer managed by their owner, the latter must be assured that security is preserved.

To respond to the security of stored data whose axes are based on the data integrity and confidentiality of remote users during their access to their data stored in the cloud, some work in the literature has been proposed to find effective solutions to this problem. In (Lee et al., 2015), the authors proposed an authentication method that allowed a remote user to securely access the intelligent learning system in the cloud environment. This suggested solution was based primarily on a two-factor method using the Universal Subscriber Identity Module Based Identity (USIM_ID) and access information (ID and password), to achieve an authentication approach ensuring data security and control their access in the cloud environment.

In addition, the authors in (Dey et al., 2016) suggested a new authentication scheme for mobile cloud computing. This Message-Digest based Authentication (MDA) scheme relied primarily on three phases: registration, authentication, and updating, to ensure a secure authentication process that would verify the confidentiality and integrity of users while accessing cloud data.

Moreover, the authors in (Lin et al., 2017) proposed an authentication solution based on the use of a dynamic identifier and on the key agreement without any trusted server. In addition, the suggested authentication scheme facilitated the access of the mobile terminal by optimizing the calculation processing performed on the client side since the mobile terminal had a limited processing capacity.

In addition, the authors in (Roy et al., 2017) put forward a new authentication scheme that provided secure and lightweight mobile user access to the cloud. This proposed authentication scheme utilized cryptographic hash, XOR bit-level and fuzzy extractor functions and relied primarily on three basic entities that were the mobile user, the cloud server or the cloud service provider, and the center trust registration.

Moreover, the authors in (Binu et al., 2018) propounded an authentication protocol that relied on the use of two-factor authentication forcing the user to enter its password and the parameters stored in the crypto-token / mobile-token in order to prove the identity of the user to the cloud authentication server. In this way, the suggested authentication protocol would not retain information about the password or user keys used and would not require a verifier table generated by the cloud authentication server to verify the identity of the user cloud.

In Table 1, we present a comparative study of all the cloud computing security approaches mentioned earlier.

Table 1: Advantages and limitations of authentication schemes in cloud computing environment.

Schemes	Advantages	Limitations
Lee et al	- It used a two-factor authentication method utilizing the Universal Subscriber Identity Module Based Identity (USIM_ID) to allow a remote user to securely access the intelligent learning system in the cloud environment.	- It lacked an effective phase during which a remote user can change its password. - It was vulnerable to stolen mobile devices and privileged insider attacks.
Dey et al	- It relied on the utilization of the user's ID and password in combination with the hash functions for the purpose of establishing authenticated communication sessions between remote users and cloud computing. - It was able to resist against man-in-the-middle and replay attacks.	- It did not contain a password change phase allowing a remote user to change its password easily. - It was susceptible to stolen mobile devices and password guessing attacks.
Lin et al	- It did not require the use of a trusted server. It was mainly based on the utilization of a dynamic identifier with a key agreement. - It provided security against a variety of attacks.	- It required high computation and communication costs. - It did not provide security against session key disclosure attacks. - In this scheme, the remote user had no way to change its password.
Roy et al	- It did not require a trusted third party. - It took into account the secure exchange of keys, as well as the property of anonymity of users.	- This scheme provided high security. However, the major disadvantage of utilizing this scheme in the context of cloud computing was the high computing and communication costs.
Binu et al	-It provided a strong two-factor authentication mechanism to authenticate remote users to cloud service providers. - It utilized a protocol, called the Security Assertion Markup Language (SAML), to exchange user authentication information between the identity provider and service providers. - It provided a brokered authentication protocol that used two-factor authentication to prove the identity of the remote user to the authentication server.	- It did not ensure the anonymity of remote users. - It did not contain any phase that would enable the remote user to change its password. - It did not resist against the stolen mobile device attack. - It required a huge amount of computation and communication costs.

3 OUR PROPOSED APPROACH

After presenting, in the previous section, the solutions ensuring the security of data access of remote users in the cloud and the limits they contained, we describe in this section the details of our solution. The latter is mainly based on four phases: *(i)* parameters generation;*(ii)* identification;*(iii)* authentication; and *(iv)* changing of password in the case where the user wants to change his password. All these phases are performed to ensure the security of access to the data of remote users in the cloud and to stem any attempt to attack their operation. The details of each phase are as follows:

3.1 Parameters Generation Phase

During this phase, the Cloud Services Provider (CSP) first chooses $\hat{W}$ and $\check{C}$, two distinct prime numbers. Subsequently, it calculates their product $\acute{K} = \hat{W} \cdot \check{C}$ and also calculates $\varphi(\acute{K}) = \hat{W}\,\check{C} - \hat{W} - \check{C} + 1$. Then, it chooses a natural integer $\bar{A}$ prime with $\varphi(\acute{K})$ and strictly less than $\varphi(\acute{K})$. Subsequently, the CSP calculates the natural integer $\hat{G}$. This integer denotes the inverse of $\bar{A}$ modulo $\varphi(\acute{K})$. Since $\bar{A}$ is prime with $\varphi(\acute{K})$, there are two integers $\hat{G}$ and $\breve{y}$ such that $\bar{A} \cdot \hat{G} = 1 + \breve{y} \cdot \varphi(\acute{K})$, that is $\bar{A} \cdot \hat{G} \equiv 1 \pmod{\varphi(\acute{K})}$. Once the CSP has generated the list of keys, it keeps the two distinct primes $\hat{W}$ and $\check{C}$ and the natural number $\hat{G}$ secret and publishes the pair $(\bar{A}, \acute{K})$. Next, it chooses a one-way hash function $H(.): \{0, 1\}^* \to \{0, 1\}^x$ with x indicating the size of the binary string.

3.2 Identification Phase

In our solution, any remote user that wants to store its data in the cloud and to remotely access them safely can create an account easily. During the execution of the identification phase, the remote user chooses one login L and one password P. Subsequently, the terminal of the remote user calculates $\alpha = H(P \,||\, \Theta)$ with Θ indicating a random number identified by the remote user. Once the user has completed both the information of its login L and its password P and the hashing calculation performed on their password P, the result of the calculation of hash α and its login L are sent to the CSP through a secure communication channel.

Once the login L and the result of the calculation of hash α sent by the remote user are received by the CSP, the latter performs the following calculations: $\delta = H(L \,||\, \hat{G})$, $\beta = H(\delta \,||\, \alpha \,||\, L)$, $\varepsilon = \delta \oplus H(\alpha \,||\, L)$. As soon as the CSP completes the calculation, it sends $(\acute{K}, \bar{A}, \beta, \varepsilon)$ to the remote user through a secure channel.

As soon as the remote user receives the result of the calculation sent by the CSP, the remote user stores $(\acute{K}, \bar{A}, \beta, \varepsilon, H(.))$ on its terminal and also introduces one random number Γ in its terminal.

3.3 Authentication Phase

In case the remote user wishes to access his data stored in the cloud, he must first enter his login $L^{|}$ and password $P^{|}$. Subsequently, his terminal will perform the following calculation operations:

- It calculates first $\alpha^{|} = H(P^{|} || \Theta)$, $\delta^{|} = \varepsilon \oplus H(\alpha^{|} \,||\, L^{|})$.
- It also calculates $\beta^{|} = H(\delta^{|} \,||\, \alpha^{|} \,||\, L^{|})$ and verifies afterwards if the obtained $\beta^{|}$ result is equal to result β previously stored in the identification phase. In this case, the terminal of the remote user continues its calculation because this means that $L^{|} = L$ and $P^{|} = P$. Otherwise, the terminal of the remote user terminates the session directly.
- It calculates $\hat{A} = H(\delta^{|} || H(\alpha^{|}) || L^{|} || \ddot{U} || \hat{I})$ and $\hat{O} = (L^{|} || \hat{A} || \hat{I})^{\bar{A}} \bmod \acute{K}$, where $\ddot{U}$ is the current connection timestamp and $\hat{I}$ represents a random number generated by the terminal of the remote user.

Once the user terminal completes the calculations, it sends the results $(\hat{O}, \varepsilon, \ddot{U})$ to the CSP.

As soon as the calculation results $(\hat{O}, \varepsilon, \ddot{U})$ are received by the CSP, the latter will first check the legitimacy of the remote user executing the authentication phase. To do this, the CSP will follow the following steps:

- It first checks the timestamp validity $(\ddot{U}c - \ddot{U}) <= \Delta T$, where ΔT represents the maximum transmission delay and $\ddot{U}c$ is the timestamp during which the connection message is received by the CSP. If the verification result is correct, then the CSP decrypts $\hat{O}$; thanks to the decryption key $\hat{G}$ as $\hat{O}^{\hat{G}} \bmod \acute{K}$ in order to obtain $(L' || \hat{A}' || \hat{I}')$. Subsequently, it calculates $\delta' = H(L' \,||\, \hat{G})$; $[H(\alpha^{|} \,||\, L^{|})]' = \varepsilon \oplus \delta'$.
- It calculates $\hat{A}'' = H(\delta' \,||\, [H(\alpha^{|} \,||\, L^{|})]' \,||\, \ddot{U} \,||\, \hat{I}')$, and then it verifies whether the two results $\hat{A}'$ and $\hat{A}''$ are equal or not. This means that the remote user is a legitimate user who can access all functions from the moment they authenticate. Otherwise, access will be denied, and the session will be ended.

- It calculates $\boldsymbol{\hat{E} = H(\hat{U} \| \delta')}$ and $\boldsymbol{\ddot{O} = \hat{I}' \oplus \hat{U}}$ and then sends $\boldsymbol{(\hat{E}, \ddot{O}, \ddot{U}c)}$ to the terminal of the remote user; where $\boldsymbol{\hat{U}}$ is a random number generated by the CSP and $\boldsymbol{\ddot{U}c}$ is the current CSP timestamp.

Once $\boldsymbol{(\hat{E}, \ddot{O}, \ddot{U}c)}$ are received at the time $\boldsymbol{\hat{U}t}$ by the remote user's terminal, the latter will first check $\boldsymbol{(\hat{U}t - \ddot{U}c) <= \Delta T}$. If this condition is verified, then the terminal of the remote user will first calculate $\boldsymbol{\hat{U}^{|} = \hat{I} \oplus \ddot{O}; \hat{E}^{|} = H(\hat{U}^{|} \| \delta^{|})}$ and then check whether the result $\boldsymbol{\hat{E}^{|}}$ is equal to the result received. If both results are verified, this indicates that the remote user is authenticated to the CSP. Subsequently, the remote user and the CSP have agreed on a common trust key ***(TK)*** providing secure communication between them. This confidence key is calculated as follows: $\boldsymbol{TK = H(\hat{I} \| \delta^{|} \| \hat{U}^{|}) = H(\hat{I}' \| \delta' \| \hat{U})}$.

3.4 Password Change Phase

If remote users wishes to change their password, they must first enter their login and old password. Subsequently, terminal will list the following calculations:

- $\boldsymbol{\alpha_{\{Current\}} = H(P\{_{Current}\} \| \Theta)\ ;\ \delta_{\{Current\}} = \varepsilon \oplus H(\alpha_{\{Current\}} \| L)}$
- $\boldsymbol{\beta_{\{Current\}} = H(\delta_{\{Current\}} \| \alpha_{\{Current\}} \| L)}$

Once the terminal completes the calculation, it will verify thereafter the result $\boldsymbol{\beta_{\{Current\}}}$ obtained at result $\boldsymbol{\beta}$ previously stored. If the two results are equal, the terminal continues in the following steps because this means that $\boldsymbol{P_{\{Current\}} = P}$. Otherwise, the terminal rejects the requested operation.

After that, the remote user enters in his terminal his new password $\boldsymbol{P_{\{New\}}}$. Once he has entered in his terminal his new password, his terminal makes the following calculations: $\boldsymbol{\alpha_{\{New\}} = H(P_{\{New\}} \| \Theta)\ ;\ \beta_{\{New\}} = H(\delta_{\{Current\}} \| \alpha_{\{New\}} \| L)}$ and $\boldsymbol{\varepsilon_{\{New\}} = \delta_{\{Current\}} \oplus \mathrm{H}\,(\alpha_{\{New\}} \| L)}$. Once the terminal of the remote user finishes the calculation, it replaces thereafter $\boldsymbol{\beta_{\{New\}}}$ and $\boldsymbol{\varepsilon_{\{New\}}}$ by $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$.

4 APPROACH VALIDATION AND RESULTS EXPLOITATION

This section is reserved for the experimental part of our approach. It has two main parts. First, we begin by experimentally describing the reliability and performance of the solution we propose in terms of computational and communication costs, data confidentiality and integrity. Secondly, we validate the security aspects of the suggested solution based on security analysis.

4.1 Analysis of Performances of Other Proposed Schemes

To illustrate the reliability and the performance of our authentication scheme, which basically consists of two stages: *(i)* the initiation and registration stage; and *(ii)* the connection and authentication internship. Indeed, the first stage is invoked only once the user wishes to create an account. Whereas, the second stage is used each time the user wants to access his account in the cloud. For this, we will focus on the second stage of connection and authentication because it refers to the stage most invoked during the authentication process.

We illustrate in Table 2 the communication costs of our suggested solution with other solutions presented in the literature. We note that in our scheme the connection parameters (the login $\boldsymbol{L}$ is on 64 bits as well as the password $\boldsymbol{P}$ also on 64 bits). The timestamps $\boldsymbol{\ddot{U}, \ddot{U}c, \hat{U}t}$, the arbitrary numbers $\boldsymbol{\hat{I}}$, $\boldsymbol{\hat{U}}$ and the hash function $\boldsymbol{H(.)}$ used are all 128 bits long. As a result, the total number of bits utilized during the second stage, i.e. during the connection and authentication phase, is of a (128 + 128 + 128 + 128)= 512-bit length. We can say by this that the communication cost of our scheme is very low compared to the existing schemes in the literature.

Table 2: Comparison of our solution with other solutions in terms of communication cost.

Scheme	Communication cost
Lee et al	1184
Dey et al	1280
Lin et al	1536
Roy et al	864
Binu et al	2304
Ours	**512**

Table 2 clearly shows that our solution uses less computing operations and therefore our solution has a better computational cost than the solutions presented in the literature.

In Table 3, we show the computational cost of our solution compared to other existing solutions in the literature. At the level of the computational cost performance analysis, we associate notations with the cryptographic operations used by our authentication scheme. Its notations are as follows: ***Texp*** designates the elapsed time in order to perform the exponentiation operations, ***Th*** represents the elapsed time in order to perform the hash calculation and

Tmult denotes the elapsed time in order to perform the multiplication operations.

Table 3: Comparison of our solution with other solutions in terms of computational cost.

Scheme	**Computational cost**
Lee et al	4 *Th* + 3 *Tmult*
Dey et al	5 *Th* + 4 *Tmult*
Lin et al	10 *Th* + 2 *Tmult*
Roy et al	9 *Th* + 1 *Tmult*
Binu et al	9 *Th* + 3 *Tmult*
Ours	**4 *Th* + 1*Texp***

Once we present the performance and reliability of our solution with other similar solutions proposed in the literature in terms of computational and communication costs, we further describe the two properties as to the confidentiality and integrity of the data provided by our solution. The details of these properties are as follows:

Confidentiality. It is the ability to make data incomprehensible and inaccessible to any attacker; i.e., only authorized users can access the content of data (Obaidat et al., 2007) (Hammami et al., 2016). This property is ensured by our approach, since the terminal of the remote user performs the following calculations $\hat{A} = H(\delta^{|} \| H(\alpha^{|}) \| L^{|} \| \ddot{U} \| \hat{I})$ *and* $\hat{O} = (L^{|} \| \hat{A} \| \hat{I})^{\bar{A}} \bmod \acute{K}$; $\varepsilon = \delta \oplus H(\alpha \| L)$ on the information used during the authentication phase. These same results are verified only by the CSP whether they are correct. As a result, it is very difficult for an attacker to retrieve information from the results of the hash functions to obtain one of the connection data. Thus, the approach, we put forward, verifies the property of the confidentiality of information.

Integrity. Once we verify the confidentiality of remote users' data, it is time for us to be able to ensure their integrity (Obaidat et al., 2019). This property proves that data from remote users have not been tampered with by unauthorized user during their cloud migration, i.e. between the time they were issued by the user and the time they were sent to the CSP and received by him. This property is verified by our solution since the terminal of the remote user generates a connection timestamp of the remote user $\ddot{U}$. The same terminal performs a hash function $H(.)$ on the data of connections entered by the remote user and sends the result to the CSP. The latter checks the timestamp validity of the result $(\ddot{U}c - \ddot{U}) <= \Delta T$, where Üc is the timestamp during which the login message is received by the CSP. As a result, the CSP can check and detect any falsification by an unauthorized user.

4.2 Security Analysis

In the following, we describe the validation of the suggested approach for security in terms of properties provided by each solution. In Table 4, we compare the features of our solution versus a few existing solutions in the literature.

Table 4: Security analysis.

Scheme	Lee et al	Dey et al	Lin et al	Binu et al	**Ours**
Resistant to replay attack	Yes	Yes	Yes	Yes	**Yes**
Efficient password change phase	No	No	No	Yes	**Yes**
Resistant to session key disclosure attack	Yes	Yes	No	Yes	**Yes**
Resistant to stolen mobile device attack	No	No	Yes	No	**Yes**
Resistant to privileged insider attack	No	Yes	Yes	Yes	**Yes**
User anonymity	Yes	Yes	Yes	No	**Yes**
Resistant to password guessing attack	Yes	No	Yes	Yes	**Yes**

4.2.1 Resistant to Replay Attack

This is an attack that involves intercepting data packets and transmitting them as they are to the recipient (Singh et al., 2018). In our scheme we curb this type of attack by using randomly generated numbers $\hat{I}$ and $\hat{U}$ and different timestamps $\ddot{U}$ and $\ddot{U}c$. As a result, any communication of the request / response form exchanged $(\hat{O}, \varepsilon, \ddot{U})$; $(\hat{E}, \ddot{O}, \ddot{U}c)$ between the remote user and the CSP are different from one session to another, which makes it impossible for an attacker to apply this type of attack in our solution.

4.2.2 Efficient Password Change Phase

This allows a remote user to easily change its password when they want. The process is simple: Just enter in the terminal, login L and password $P_{\{Current\}}$. After this information, its terminal will compute $\alpha_{\{Current\}} = H(P_{\{Current\}} \| \Theta)$ and $\delta_{\{Current\}} = \varepsilon \oplus \mathrm{H}(\alpha_{\{Current\}} \| L)$; $\beta_{\{Current\}} = H(\delta_{\{Current\}} \| \alpha_{\{Current\}} \| L)$. Once the terminal of the remote user completes this calculation, it then checks whether $\beta_{\{Current\}} = \beta$ was stored previously. If so, this means that $P_{\{Current\}} = \mathrm{P}$, where the new password is updated. Otherwise the terminal ends the session.

4.2.3 Resistant to Session Key Disclosure Attack

Our solution is resistant to session key disclosure attack by securing all requests / replies from the communication exchanged between the remote user and the CSP by a trust key generated by the random numbers $\hat{I}, \hat{U}$ and by the identifier of the remote user L as well as the secret key $\hat{G}$ generated by the CSP. This confidence key TK is calculated in the following manner $TK = H(\hat{I} \| \delta \| \hat{U}) = H(\hat{I} \| H(L \| \hat{G}) \| \hat{U}))$. Indeed, this key of confidence is different from one session to another according to the random numbers $\hat{I}, \hat{U}$. We note that the identifier of the remote user is sent in an encrypted manner as follows: $\hat{O} = (L \| \hat{A} \| \hat{I})^{\bar{A}} \bmod \acute{K})$. As a result, an unauthorized user will not be able to decrypt $\hat{O}$ because it will need to have the secret key $\hat{G}$ generated by the CSP. By exposing this, an untrue user has no opportunity to calculate the trust key generated between the remote user and the CSP.

4.2.4 Resistant to Stolen Mobile Device Attack

If the terminal of the remote user is stolen or lost, a non-legitimate user may have the parameters $(\beta, \varepsilon, \acute{K}, \Theta)$ from this terminal. Afterwards, they can extract the connection parameters $(\hat{O}, \varepsilon, \ddot{U})$, but they cannot access the account of the remote user because they do not have any information on the password of the remote user as well as the secret parameter $\hat{G}$ generated by the CSP. Consequently, we can say that our solution is resistant to stolen mobile device attacks.

4.2.5 Resistant to Privileged Insider Attack

This is an attack on a computer system by someone with authorized access to the system (Ramachandra et al., 2017). This type of attack does not work in our solution because an internal attacker cannot obtain any information on the password P of the remote user because it is secured by a hash function in combination with a randomly generated number $\alpha = H(P \| \Theta)$. Therefore, an internal attacker must compute the inverse of the hash function and must know the random number Θ in order to have the password of the remote user. Doing so, we can say that our secure authentication solution is resistant to privileged insider attacks.

4.2.6 User Anonymity

Anonymity of the user means that an attacker cannot trace the remote user from the execution of their actions. Anonymity provides the ability to provide robust security for the identity of the remote user. In this way, an attacker cannot follow the actions of a remote user (Kumar et al., 2020). This property is ensured by our solution. If an illegal user has the terminal of the remote user and has subsequently obtained the following parameters: $(\beta, \varepsilon, \acute{K}, \bar{A}, \Theta, H(.))$; $\delta = H(L \| d)$; $\beta = H(\delta \| \alpha \| \mathrm{L})$; $\varepsilon = \delta \oplus \mathrm{H}(\alpha \| \mathrm{L})$, then a non-legitimate user cannot know the identifier of the remote user because $\delta$and β are secured by the hash function $H(.)$ and this same non-legitimate user cannot obtain L and ε because L , P and $\hat{\mathrm{G}}$ are unknown.

4.2.7 Resistance to Password Guessing Attack

It is an attack in which a remote user's right of access to his resources in the cloud is compromised by testing, one by one, all possible combinations of logins and passwords to find the right one. Our solution is resistant to this type of attacks. Indeed, an unauthorized user cannot guess the password of the remote user P from parameters $(\beta, \varepsilon, \acute{K}, \bar{A}, \Theta)$ and from the response requests exchanged by the remote user and the CSP$(\hat{O}; \varepsilon; \ddot{U})$; $(\hat{E}; \ddot{O}; \ddot{U}c)$ during the authentication process. Moreover, this same non-legitimate user cannot guess the password of the remote user P because he has to have the randomly generated number Θ, login L as well as the secret key $\hat{G}$ and he must also calculate the inverse of the hash function $H(.)$. From the above, we can say that our solution is resistant to password guessing attacks.

In summary of this security analysis sub-section, we can say that the effectiveness of our proposal in terms of requested security services shows that our solution provides a high level of security compared to other solutions proposed in the literature.

5 CONCLUSION

The solution suggested in this work is a reliable contribution and it provides an effective solution to the problem of data access security for remote users when accessing to their data stored in cloud environments. Our contribution aims to put forward an adaptive authentication mechanism to protect the personal data of each user in the cloud environment. Thus, this mechanism strengthens effective authentication and motivates remote users to access services provided by the CSP with greater confidence and greater ability to control their personal information. Through this work, we have propounded a new approach to protect the confidentiality and integrity of data access from remote users anywhere in the cloud. First, we have started with the definition of cloud computing. Then, we have introduced the problem of data security for remote users, while presenting the data security risks that hinder the use of cloud computing. After that, we have reviewed some approaches to the data security problem of remote users by performing a comparative study between them at the end. In conclusion of this study, we have proposed our approach of protection, exposed in detail, whose performance has been demonstrated in terms of communication and computational costs, confidentiality and integrity of data and resistance to various types of attacks, which proves that this approach can outperform its competitors.

REFERENCES

Marinescu, D. C. (2017). Cloud computing: theory and practice. Morgan Kaufmann.

Vasiljeva, T., Shaikhulina, S., & Kreslins, K. (2017). Cloud computing: business perspectives, benefits and challenges for small and medium enterprises (case of Latvia). Procedia Engineering, 178, 443-451.

Kumar, P. R., Raj, P. H., & Jelciana, P. (2018). Exploring Data Security Issues and Solutions in Cloud Computing. Procedia Computer Science, 125, 691-697.

Li, J., & Li, Q. (2018). Data security and risk assessment in cloud computing. In ITM Web of Conferences (Vol. 17, p. 03028). EDP Sciences.

Djellalbia, A., &Boukerram, A. (2016). Authentification Anonyme dans un environnement Cloud (Doctoral dissertation, Université Abderrahmane Mira-Bejaia).

Lee, A. (2015). Authentication scheme for smart learning system in the cloud computing environment. Journal of Computer Virology and Hacking Techniques, 11(3), 149-155.

Dey, S., Sampalli, S., & Ye, Q. (2016). MDA: message digest-based authentication for mobile cloud computing. Journal of Cloud Computing, 5(1), 18.

Lin, H. Y. (2017). Efficient mobile dynamic ID authentication and key agreement scheme without trusted servers. International Journal of Communication Systems, 30(1), e2818.

Roy, S., Chatterjee, S., Das, A. K., Chattopadhyay, S., Kumar, N., & Vasilakos, A. V. (2017). On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. IEEE Access, 5, 25808-25825.

Binu, S., Misbahuddin, M., & Raj, P. (2018). A strong single sign-on user authentication scheme using mobile token without verifier table for cloud based services. In Computer and Network Security Essentials (pp. 237-261). Springer, Cham.

Obaidat, M., & Boudriga, N. (2007). Security of E-systems and Computer Networks. Cambridge University Press.

Obaidat, M. S., Traore, I., & Woungang, I. (2019). Biometric-Based Physical and Cybersecurity Systems (pp. 165-187). Springer.

Singh, V., & Pandey, S. K. (2018, December). Revisiting Cloud Security Threats: Replay attack. In 2018 4th International Conference on Computing Communication and Automation (ICCCA) (pp. 1-6). IEEE.

Ramachandra, G., Iftikhar, M., & Khan, F. A. (2017). A comprehensive survey on security in cloud computing. Procedia Computer Science, 110, 465-472.

Kumar, M., Sharma, M., Raina, I., Kawatra, M., Sharma, S., & Sinha, A. (2020). Help Me Invest: Investment Tools and Security Risks. In Forensic Investigations and Risk Management in Mobile and Wireless Communications (pp. 257-269). IGI Global.

Hammami, H., Brahmi, H., Brahmi, I., & Yahia, S. B. (2016). Security Issues in Cloud Computing and Associated Alleviation Approaches. In 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 758-765). IEEE.