Received: 17 February 2020 | Revised: 23 April 2020 | Accepted: 2 June 2020 | Published on: 1 July 2020

DOI: 10.1002/spy2.121

RESEARCH ARTICLE

WILEY

Password-based encryption approach for securing sensitive data

Ahmet F. Mustacoglu[1] | Ferhat O. Catak[2] | Geoffrey C. Fox[3]

[1]Cybersecurity Engineering, Istanbul Sehir University, Istanbul, Turkey

[2]Information Security and Communication Technology, NTNU Norwegian University of Science and Technology, Gjøvik, Norway

[3]School of Informatics and Computing, Indiana University, Bloomington, Indiana,

Correspondence
Ahmet F. Mustacoglu, Cybersecurity Engineering, Istanbul Sehir University, Istanbul, Turkey.
Email:
ahmetfatihmustacoglu@sehir.edu.tr

Abstract

The direction of computing is affected and lead by several trends. First, we have the Data Overwhelm from Commercial sources (eg, Amazon), Community sources (eg, Twitter), and Scientific applications (eg, Genomics). Next, we have several light-weight clients belong to many devices spanning from smartphones, tablets to sensors. Then, clouds are getting popular due to their advantages since they are cheaper, greener, and easy to use compared to traditional systems. Finally, sensitive data stored as a plain text on a cloud system would be vulnerable to unauthorized access and the security become an important aspect. We believe that these advancements steer both research and education, and will put together as we look at data security in the cloud. We introduce a password-based encryption (PBE) approach to protect sensitive data, investigate the performance metrics of the proposed approach, and present the experimental results for the key generation and the encryption/decryption calculations.

KEYWORDS

cloud computing, federation and unification, password-based encryption, privacy, security

1 | INTRODUCTION

Cloud computing has been rising as one of the most robust and widespread technologies that provide access to shared resources such as CPUs, hard disks, network devices, and so on that can be automatically assigned and freed with minimum administrative work.[1] Clouds offer improved functionality and better cost-performance than traditional approaches in many areas of scientific research, computational science, and engineering.[2] Many of these opportunities have not been explored in depth as there is currently no viable business model as clouds charged as operating funds (bearing overhead) must compete with no-cost resources available through universities and federal initiatives. On general principles, one can expect clouds to be the most economical computing resource as they offer economies of scale (one has around 100 000 servers in a large cloud data-center) and their internet access model can allow cloud-centers to be placed in optimal locations where operating costs are low and environmental impact is minimal. Of course current national supercomputer resources operate near 100% utilization (whereas clouds typically operate below full utilization allowing an attractive interactive model) and often are directly or indirectly subsidized by the host organization and this obscures the comparison of cloud and traditional scientific computing approaches.[3]

Clouds offer interesting opportunities as both infrastructure (IaaS) and software (PaaS) levels. Their software model has been designed for the wide-range data-intensive applications in the e-commerce, social media, and search fields. These have been reinforced by the commercial cloud focus as general next-generation enterprise data-center technology.

Comparing clouds, grids (distributed systems), and supercomputers, clouds have synchronization and communication costs that lie between those of distributed systems and supercomputers. Further clouds tend to be optimized for external access and not for inter-node communication performance. Thus highly parallel large scale simulations are not likely to move to clouds in the near future and should remain staple of traditional supercomputers. However, there are two important classes of applications where clouds could perform well and offer attractive cost-performance, interactive elastic (on demand) use, and powerful new software platforms. These classes are

- Pleasingly parallel applications and with some overlap
- Data-intensive applications

Clouds offer an interesting high throughput computing model for the pleasingly parallel case where there are two important cases—namely, parallelism over users and usages. The former is illustrated by the many users of a Web 2.0 site in commercial applications and by support of the "long tail of science" (the many small users with individual jobs) in scientific case. The success of the European Venus-C project on the Azure cloud is a good example here. Parallelism of usages could be illustrated by particle physics data analysis (each event set can be analyzed independently) or the support of Sensor nets or more generally the "Internet of Things" where over 20 billion devices are predicted on the Internet by 2020; each sensor is naturally connected elastically (as individual sensors such as smartphones do not have 100% duty cycle) to a core in the cloud.[3]

Cloud computing paradigm is based on sharing resources which offers integration of applications as a web services and storage services. There are many well-known cloud computing suppliers such as Microsoft, Yahoo, Amazon, Google and others. Amazon web services (AWS) is a major architecture provider for services since 2002 and then developments and new approaches for cloud computing had been proposed and utilized. Data can be store on servers in many ways. When cloud service providers store data on cloud, they should be careful to preserve the CIA triad (confidentiality, integrity, and availability) of data. Confidentiality is interested in preserving data private. Privacy is mostly focused on preventing the disclosure of sensitive data. Confidentiality can be provided by several approaches such as access control, encryption, authorized protection, and so on. Integrity is the assurance level that what data is expected to be in cloud, what is really in the cloud, and is restricted from intentional or unintentional access without permission. Availability represents the level of usable services that are available to spend by cloud users. Cloud computing technologies can be helpful to increase availability by taking advantage of broad internet-enabled access to resources, however the client is depending on the solid deployment of computing resources on time. Availability is provided in high ratios by well-defined architecture by the provider, as well as well-prepared contracts and terms of agreement. Security of data storage in cloud defines the level of data access supporting users with ability to specify access restrictions and security measures.[4] Cloud storage[5] identifies the inexpensive cloud storage and backup options for small businesses. Depend on data importance, data might be located on a single storage system or might be mirrored on multiple storage systems. An architecture of a popular cloud computing storage model composed of a master control server with connected clients. The cloud storage system is formed as four different layers: (a) a storage layer that is responsible for storing the data; (b) a basic management layer that is charged with ensuring the security and the stability of cloud storage itself; (c) an application interface layer that delivers service platform for applications; and (d) an access layer that is focused on the access platform.[4]

The encryption approach is the most generally used method to secure data in the cloud computing environment. The client data can be classified into two different categories as private or public data. The public data is accessible from any trusted clients leading to producing an environment for collaboration. On the other hand, private data is not sharable data. It is client's own private data and it must be carried in its encrypted form due to security and privacy reasons. Latest cryptosystem can be categorized as symmetric and asymmetric cryptosystem based on the key characteristics. An encryption key and a decryption key are shared between the receiver and the sender in a symmetric cryptosystem where as a public and a private key pairs are used in asymmetric cryptosystems. These two keys for symmetric system are the same or easy to deduct from each other. The popular symmetric cryptosystems include DES (Data Encryption Standard), Blowfish, RC5, 3DES, RC6, Two-Fish, and AES (Advanced Encryption Standard). In asymmetric cryptosystems, the public key can be revealed but the private key should be hidden. The famous asymmetric cryptosystems are RSA (Rivest Shamir Adleman) and ECC (Elliptic Curve Cryptosystem).[4]

Today, different type of data is stored on a cloud environment in an unencrypted form. The stored data might be isolated from others for security reasons but this may not be enough to protect the data. Since unknown vulnerabilities could be discovered at any time leading to security and privacy issues for the stored data. For example, recent security bugs called “Meltdown and Spectre”[6-9] affect approximately every two major computer processors made in the last 20 years. It could lead to compromised servers for cloud platforms resulting in access to even isolated sensitive data. We propose a password-based encryption approach that stores data in an encrypted state on the cloud to protect the sensitive data from these types of attacks. In our proposed password-based encryption approach, the registered users with the proposed system are authenticated through the usage of encryption/decryption keys without storing the users’ passwords. The associated encryption/decryption keys are generated automatically whenever users login to the system. This also guarantees that the attack surface only covers the decreased time of period. Moreover, our system guarantees the privacy of personal data by storing a user’s credentials for social web tools in an encrypted form on a cloud. So, the user’s credentials can only be accessed after being decrypted through the decryption key that is generated when the user authenticated. If a user does not login to the system, then the user’s data cannot be accessed and it is maintained in an encrypted form on the cloud.

This paper contains a Password-based Encryption (PBE) approach that is built on top of AES system and an empirical evaluation of the proposed approach. Our work aims to protect the user-related sensitive data (username and password pairs for social web tools) that is located on a cloud. The main goal of this evaluation is to put our abstract research into practice to analyze and verify its utility in a cloud environment. The literature fails to report on empirical case studies of a password-based encryption approach that is used by a cloud federation and unification service for enabling privacy and security of user-related sensitive data. We present our contributions below:

- Our proposed work guarantees the protection of the user-related sensitive data through the usage of password-based encryption approach. The username and the password credentials for social web tools are maintained on a cloud in an encrypted form.
- Our proposed system enables any cloud federation and unification service with ability to authenticate its users through the usage of encryption/decryption keys without storing the users’ password. The keys are generated automatically on the fly by using a key generating function and the keys are not stored anywhere else on the cloud.
- Our proposed system offers a modular security solution that is based on password-based encryption approach for any cloud federation and unification service, and it ensures the security of the sensitive data located on a cloud environment. Our research also includes some calculations regarding the password space and the necessary time to obtain a user’s password to break into the system.

Along with the description of the architecture, this article also includes the experimental analysis of the proposed approach, analyzing its usefulness by examining the overhead value due to the added PBE security layer. In our proposed research, we aim to have a system that is easy to use and to provide some level of security (useable security) for the user-related sensitive data. Our goal is not to have a system, which enforces more complicated rules for ensuring the security of the sensitive data in all possible ways. We are motivated to have a security module that is not complicated and it is easy to use. So, we prefer to have a security wrapper that works fast and smooth with some level of security without the usage of any type of two-factor authentication mechanism. This work should motivate the research of other password-based encryption systems for cloud services along with identical security handling of the sensitive data requirements.

The organization of the rest of the paper is as follows. Section 2 gives an overview about cloud computing and related security concepts. Section 3 explains the architecture of the proposed system. Section 4 presents evaluation test results for the prototype system running on Amazon Public Cloud. Finally, we conclude with some final remarks and future work in Section 5.

2 | BACKGROUND AND RELATED WORK

As explained in Reference 1, cloud composed of five major attributes, three service styles, and four deployment types. Cloud computing is an arising technology one whose main concentration is scalable elastic service, on-demand service, metered service, broad network access, sharing of resources resulting in saving of scale in performance, and electrical power (Green IT).[10] These related to Infrastructure as a Service but there are also solid new software approaches fitting into Platform as a Service and Software as a Service that are also important.[11] Cloud technology provides solid architectures to execute complex extensive computing tasks and makes available various IT capabilities from storage and computation to database and application services. Many organizations have attracted by cloud computing due to economic and usage advantages to store, process, and analyze huge amount of datasets.[12] A big number of scientific applications and the data have been migrated to cloud environments due to the lack of resources in local computing environments, higher costs, and increasing volume of data.[13] Also, cloud service providers have started to include parallel data processing frameworks to allow their users to have related services during the deployment of their programs.[14]

Based on NISTs definition; “Cloud computing is a model for ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction”.[1] Cloud computing has several charming features that allow companies to focus on their major operations rather than worrying about the issues related to infrastructure, various costs, availability of resources, flexibility, and maintenance.[15] Furthermore, elastic environment, resources, and services provided by the cloud are excellent opportunities for scientists to perform their experiments.[16,17] Cloud service approaches can be classified into three groups[18]:

- *Platform as a Service (PaaS)*: In this service type, cloud providers offer a platform for computing such as operating system, database, environments for code execution, application server, web server, and so on. Application developers can easily produce, compile, and run their software solutions on a cloud platform without worrying about the cost and complexity of the underlying hardware and software layers. Google’s AppsEngine, Salesforce.com, Force platform, and Microsoft Azure are the popular examples to PaaS for end users.[18]
- *Software as a Service (SaaS)*: In this service model, cloud providers serve software application that is installed and run by the cloud provider in the cloud environment so that users can access the software through a cloud client. Cloud users do not need to worry about handling the cloud infrastructure or platform where the software application is executed. SaaS model is sometimes called as “on-demand software” due to the nature of pay-per-use or subscription fee-based pricing policy. GoogleDocs, Gmail, Salesforce.com, and Online Payroll can be accessed through the Internet and can be given as examples to SaaS category.[18,19]
- *Infrastructure as a Service (IaaS)*: In this service approach, cloud providers offer actual machines, virtual machines, or other resources so that users are abstracted from the functional structure of infrastructure such as data partitioning, physical computing resources, location, load balancing, security, backup, network, and so on. Flexi scale and Amazon’s EC2 can be consumed by end users upon demand and can be given as examples to IaaS category.[18]

Benefiting from Cloud computing allows a company to save on software, hardware, and infrastructure costs, however important business information may be improperly disclosed to others[18,20] due to keeping the company’s data on the service provider’s equipment. Some research works have recommended that users’ data kept on a service-provider’s hardware should be encrypted.[21] Encrypting data before saving to a database is a well-known technic for data protection. Furthermore, service providers could build firewalls to make sure that the decryption keys associated with encrypted user data are not revealed to intruders. Moreover, if the decryption key and the encrypted data are kept by the same service provider, it increases the chance that administrators who have super admin privileges would have access to both the decryption key and the encrypted data, hence bringing a risk for the unauthorized disclosure of the users’ data.[18]

Data encryption, authentication of users before accessing data, and setting up secure channels are well-known methods to protect users’ data.[18] Cryptographic algorithms and digital signature techniques are used for these methods. Symmetric and asymmetric cryptographic algorithms are used for data encryption in popular methods. Triple Data Encryption Algorithm (TDEA, also known as Triple-DES or 3DES), Advanced Encryption Standard (AES), and others[18] use symmetric cryptography in their implementation. A secret key is used for encryption and decryption processes in symmetric cryptography, whereas asymmetric cryptography uses a “public key” for encryption, and a “private key” for decryption purposes such as RSA cryptography[22] and Elliptic Curve Cryptography (ECC).[23] Symmetric cryptography is more efficient and more suitable for encrypting huge amount of data. On the other hand, asymmetric cryptography needs more computation time and is used for providing a mechanism for sharing the decryption keys that are needed for symmetric cryptography.[18]

Password-based encryption methods are based on cryptographic hashing mechanisms. Essentially, a password and a salt, which is only a random data and it can thwart dictionary or precomputation attacks, is fed in some fashion into a mixing function based around a secure hash and the function is applied several times defined by an iteration count. The iteration count increases the cost of exhaustive password search attacks by a significant amount. Once the mixing is complete, the output byte stream is used to create the key for the cipher and possibly initialization vector as well. In our proposed work, we have used a one way hashing function PBKDF2WithHmacSHA1 defined by NIST in Reference 24 to

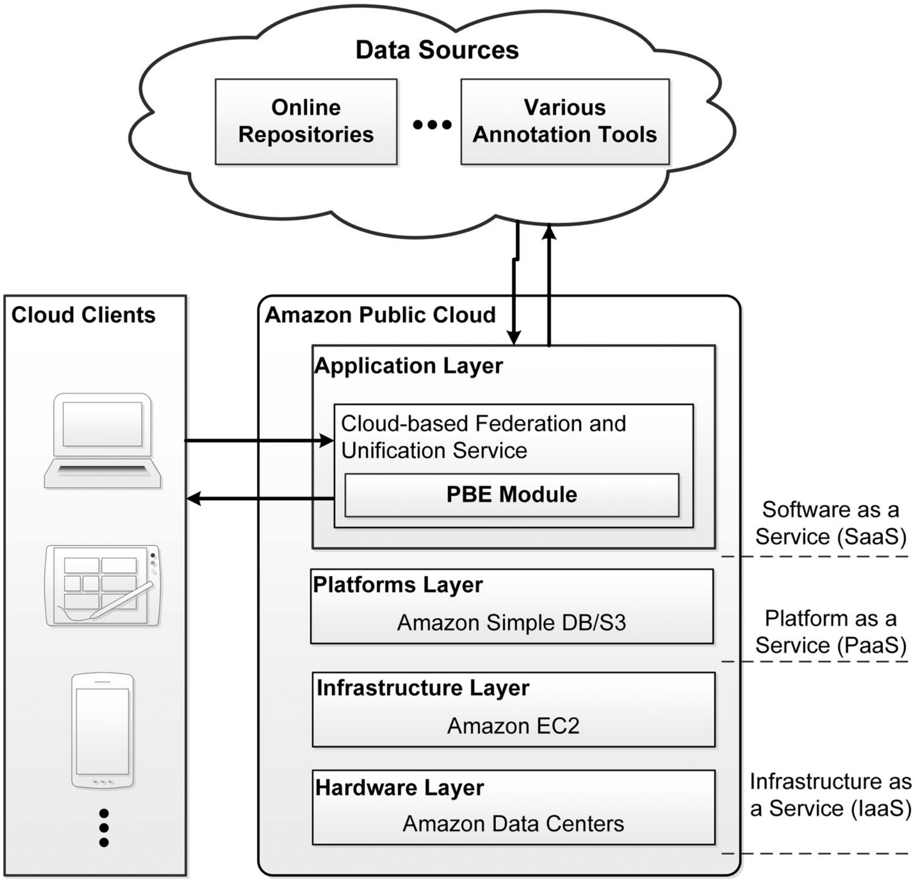

FIGURE 1 Architecture of the password-based encryption approach integrated on a cloud-based federation and unification service

generate the keys with "AES/CBC/PKCS5Padding" option and applied it to 1000 times. When using the password-based encryption, there must be a policy on passwords that is appropriate to the security requirements of the application and that the policy is enforced.

Recently, much of growing interest has been pursued in the context of data storage security on cloud services.[25-33] These techniques, which can be useful to ensure the data security and the privacy in cloud computing, are all focusing on using cryptographic and access-based control approaches. These schemes all provide efficient secure storage and data availability services, where our proposed work enhances these approaches by adding number of capabilities: (a) authentication of users through an automatic generated encryption keys without storing the users' password; (b) no necessity for the encryption/decryption key storage on a cloud environment; (c) storing sensitive data (user credentials for social web tools) in an encrypted form on a cloud by using an associated encryption key. As a result, our goal in this study is to construct an efficient protection scheme for sensitive data based on password-based encryption approach. Furthermore, our proposed system also supports the authentication of the users to the associated cloud system by using encryption keys and the encrypted data can only be reached after the user login to the system successfully.

3 | THE ARCHITECTURE OF THE PROPOSED APPROACH

Figure 1 shows the overall architecture of a cloud-based system, which utilizes the proposed password-based encryption approach to secure user-related sensitive, data runs on Amazon Public Cloud. This system consists of three main component: (a) the cloud clients; (b) the online data sources; and (c) a cloud-based federation and unification system wrapped with security module operates on Amazon Public Cloud. The cloud clients can be any clients such as smartphones, tablets, laptop PCs, and so on that interact with the system over the HTTP protocol. The online resources represent data sources located on the web such as repositories, scientific databases, social bookmarking, and annotation tools, and so on.

This system is a collection of services for managing social data scattered on the internet. During the deployment phase of the proposed system on the Amazon Cloud, the properties of the Amazon Public Cloud have been utilized and its properties are described in detail:

1. *The hardware layer*: This layer is responsible for managing the physical resources of Amazon Public Cloud such as physical servers, routers, switches, power, and cooling systems. In real life, data-centers are the places where the hardware layer is typically implemented in a data-center generally contains around thousands of servers that are organized in racks and interconnected through switches, routers, or other components. Hardware configuration, fault tolerance, traffic management, power, and cooling resource management are the typical issues of the hardware level.[34]
2. *The infrastructure layer*: The infrastructure layer is also known as the virtualization layer, this layer generates a pool of storage and computing resources by partitioning the physical resources by using virtualization technologies such as Xen,[35] KVM,[36] and VMware.[37] This layer is a crucial component of cloud computing paradigm due to many key features are only made available through virtualization technologies such as dynamic resource assignment etc. Amazon EC2 service has been utilized for the deployment of the cloud-based software services.[34]
3. *The platform layer*: The platform layer is built on top of the infrastructure layer and it consists of operating systems and application frameworks. The main purpose of this layer is to minimize the burden of deploying applications directly into VM containers. For instance, Google App Engine operates on the platform layer to provide API support for implementing database, storage, and business logic of typical web applications. Amazon Simple DB service has been used for satisfying the storage needs of the cloud-based software services.[34]
4. *The application layer*: The application layer is located at the highest level of the hierarchy and it consists of the actual cloud applications. Cloud applications can leverage the automatic-scaling feature to achieve better performance, availability, and lower operating cost when compared to traditional applications.[34] The password-based encryption module wraps the cloud-based software services and they have been deployed as a SaaS model on the Application Layer of the Amazon Cloud. The security module is used for authenticating the system users and encrypting/decrypting the user-related sensitive data located on the cloud storage. The proposed approach is composed of two major parts, namely, the registration phase and the authentication/access to social websites phase. The registration phase is performed only once, and the authentication/access to social web sites phase is executed every time a user logs into the system. The proposed work does not need to store users' passwords related to users' login. In other words, there is not any password file for managing authentication process. Users are authenticated through the usage of the generated encryption/decryption keys by comparing the entered usernames with the decrypted one. If the generated keys are correct which means that the entered password is also correct then the encrypted usernames will be matched. Details of the authentication process is explained in Section 3.2. The overall process consists of the registration, the authentication, and the usage of sensitive data and reaching sensitive data requires the cloud services to log in to remote social web tools by using the related user's credentials. The whole process is depicted in Figure 2.

3.1 | Registration phase

This phase is invoked whenever a user registers with the remote system. The registration phase is provided through the deployed cloud-based federation and unification service system and works as follow:

- A user who wants to register; enters first name, last name, selects a username, and a password for the cloud-based system. Also, the user enters the related user credentials for social web tools (sensitive data: usernames and passwords) that will be used for accessing to the remote annotation sites to retrieve data by the cloud-based services.
- The selected password by the user, automatically system generated unique salt value for each user (randomly generated number) and the number of iteration count are used through a one-way hash function to generate an encryption key. The unique salt value is stored in a database on the cloud system.
- The Profile info (sensitive data: usernames and passwords) of the user for the remote annotation sites are encrypted by the generated encryption key and stored into a database on the cloud system.
- The selected username for the proposed system is also encrypted by the generated encryption key and then stored into the cloud-based system database in two forms: an encrypted and a plain format.

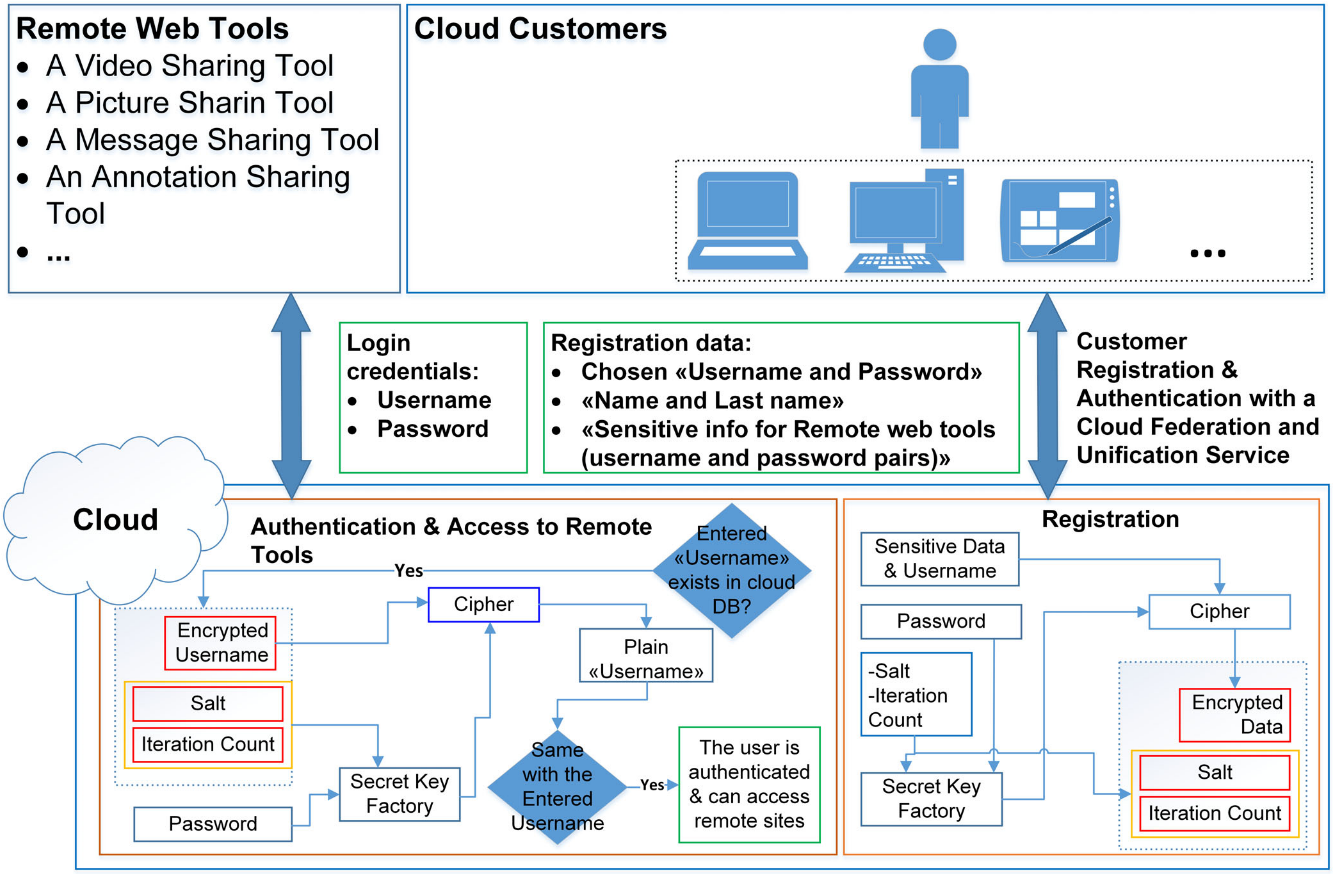
Remote Web Tools
• A Video Sharing Tool
• A Picture Sharin Tool
• A Message Sharing Tool
• An Annotation Sharing Tool
• ...
Cloud Customers
...
Login credentials:
• Username
• Password
Registration data:
• Chosen «Username and Password»
• «Name and Last name»
• «Sensitive info for Remote web tools (username and password pairs)»
Customer Registration & Authentication with a Cloud Federation and Unification Service
Cloud
Authentication & Access to Remote Tools
Entered «Username» exists in cloud DB?
Yes
Cipher
Encrypted Username
Plain «Username»
Salt
Iteration Count
Same with the Entered Username
Yes
The user is authenticated & can access remote sites
Secret Key Factory
Password
Registration
Sensitive Data & Username
Password
Cipher
-Salt
-Iteration Count
Encrypted Data
Secret Key Factory
Salt
Iteration Count

FIGURE 2 Architecture of the password-based encryption approach integrated on a cloud-based federation and unification service

3.2 | Authentication phase and access to remote social tools

This phase is invoked whenever a user wants to login to the proposed cloud-based system. After successful verification of the user, the remote system allows the user to access the system. The login and the verification phase work as follows:

- The entered username for logging into the system is compared with the one that was stored in the proposed system database in a plain format during the registration phase. If there is any match then it continues with the step 2, otherwise:
  - Throw an error message stating that username or password is invalid.
- The entered password of the user, unique salt value retrieved from the proposed system database during the registration phase and the number of iteration count are used through a one-way hash function to generate encryption/decryption keys. The encrypted username associated with the plain username located in the cloud database is retrieved and decrypted by using the decryption key. Then, it is compared with the username that is entered by the user to login:
  - If the decrypted username matches with the one that is entered by the user to login, then the user is set into the session and continue as successful login. Whenever the authenticated user wants to access to remote web tools, the federation and unification cloud services can access to remote sites by using the associated profile info. To do so, first, the regarding username and password pairs are decrypted through the decryption key that is automatically generated after the user authentication. Then the remote social tools can be accessed and targeted metadata can be retrieved easily via the provided cloud services.

- If the decrypted username does not match with the one that is entered by the user to login, then throw an error message stating that username or password is invalid.

4 | PROTOCOL ANALYSIS

The client program can be run on several environments such as smartphones, tablets, and PCs to reach the federation/unification services wrapped by the proposed password-based encryption module. The federation and unification services are deployed on Amazon Public Cloud and used as a test-bed in this work. We tested the proposed system running on Amazon Public Cloud through EC2 services. We have also used Amazon Public Cloud Simple DB/S3 for our storage needs. Before looking at a detailed inspection, let us present the sub-sections of the protocol analysis. First, we performed extensive series of measurements to evaluate the prototype implementation of the proposed architecture and investigate its practical usefulness in real-life applications. In our general experiments methodology, we have sent various requests from a client program to our proposed system implementation to test the performance metrics of our proposed system. Then, we have investigated the password space of the proposed approach and calculated the necessary time for obtaining a user's password. Finally, we have mentioned about the most possible attacks that are prevented by our proposed work and show the advantages of our work over password-based authentication.

4.1 | System registration and authentication experiments

Our main goal in doing this experiment is to measure the baseline performance of the proposed security module wrapped around the cloud unification/federation prototype services deployed on Amazon Public Cloud. In our experiments, we have used the "PBKDF2WithHmacSHA1" function to generate the key with "AES/CBC/PKCS5Padding" option to encrypt the message with 1000 iteration. We also use the similar settings to decrypt the message. In this work, time for AES and hashing operations are not considered since they are negligible. So, we have measured the total time for password-based-encryption protocol for key generation, encryption, and decryption operations. We have tested the performance of the proposed system by measuring the time spent to generate encryption/decryption keys, encryption, and decryption of the sensitive data by using the generated keys. We have also calculated the password space and the minimum required time to break the system through the brute force attacks via the usage of a personal laptop PC with a regular graphic card and the NVidia GTX 1080 graphic card. The client programs were run on a personal laptop to make a request to access social web tools from the cloud-based system, while cloud-enabled system was running on Amazon Cloud. In this experiment, we were exploring the performance metrics of our methodology for "generating encryption/decryption keys", "encryption," and "decryption" services of the proposed password-based encryption approach. In our, each testing case, the clients send sequential requests for login standard operation resulting in generating an encryption/decryption key and encryption/decryption functions are executed 1000 times. We recorded the average time for generating an encryption key, encryption/decryption of the sensitive data, and this experiment was repeated 50 times. Figure 3 shows the design of these experiments.

4.2 | System registration and authentication experiments results

We conduct experiments where we investigate the base performance of the proposed system. We have implemented the password-based encryption module in Java Language, using Java Standard Edition compiler with version 1.8.0_121-b13. The configuration of the testing environment where our client code written in java language and is running to communicate the implemented services on Amazon Cloud is given in Table 1. During the generation of the keys, encryption, and decryption operations; we have set the key length to 128 bits, salt value to 16 bits, and the iteration counter to 65 536. The test data consist of a password value (10 bytes) that obeys the rules defined in Section 4.3 and a 16-bit salt value added to a user's password resulting in a total of 12 bytes. Figure 4 represents the required times for basic key generation, encryption, and decryption operations of our system. In this experiment we recorded processing times for the generating encryption/decryption key, the decrypt/encrypt service to measure the processing times of the proposed service. Key generation, encryption, and decryption operations are also used whenever the proposed system reaches the remote annotation sites to retrieve users' data. This experiment shows the necessary time requirements to generate a key and encrypt/decrypt

FIGURE 3 Testing case for the key generation and encryption/decryption experiment

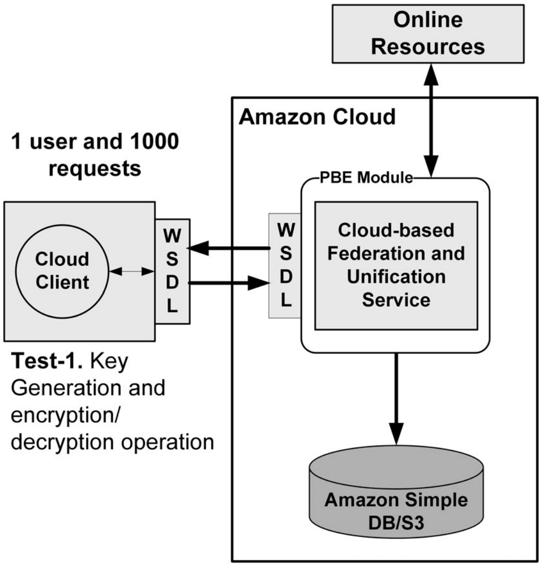

TABLE 1 Testing environment for client nodes

PC configuration	
Processor	**Intel® Core i5-3317U CPU @ 1.70 GHz**
RAM	6 GB
OS	Windows 10 (64 Bit)

operations that are necessary for major services that interacts with the online annotation systems (eg, logins with the proposed system, etc.).

Our test results for security calculations given in Figure 4 show that the key generation, encryption/decryption operations bounce between 101 005 milliseconds and 101 928 milliseconds for 1000 iterations, and the SD for the given result is 266.6921. One can easily calculate that one security operation takes approximately 101 milliseconds. Hence, we can perform around 10 key generations and encryption/decryption operations in 1 second with our testing environment. In addition to these time requirements for the security calculations of the proposed system, we also need to spend time for communicating with the remote social web tools and for retrieving a user's meta-data presented in Table 2.[38] The values in Table 2 are shared to provide better understanding how much time we need in order to retrieve data from remote social webtools without including any time for security calculations. Finally, we can calculate the total time required for encrypt/decrypt operations and for reading/saving data to remote social web tools.

4.3 | The password space and security calculations

We assume that a password value created by a user during the registration process must be at least eight characters long and required to consist of the following characters:

- Punctuations: possible 32 characters
- Capital Letters: possible 26 letters (A … Z)
- Small Letters: possible 26 letters (a … z)
- Numbers: possible 10 numbers (0 … 9)

Under these assumptions, our whole password space for a minimum eight character long password will be:

$$(26+10+26+32)^8-(10+32)^8-(26+26+32)^8+(32)^8=2^{51.68}$$

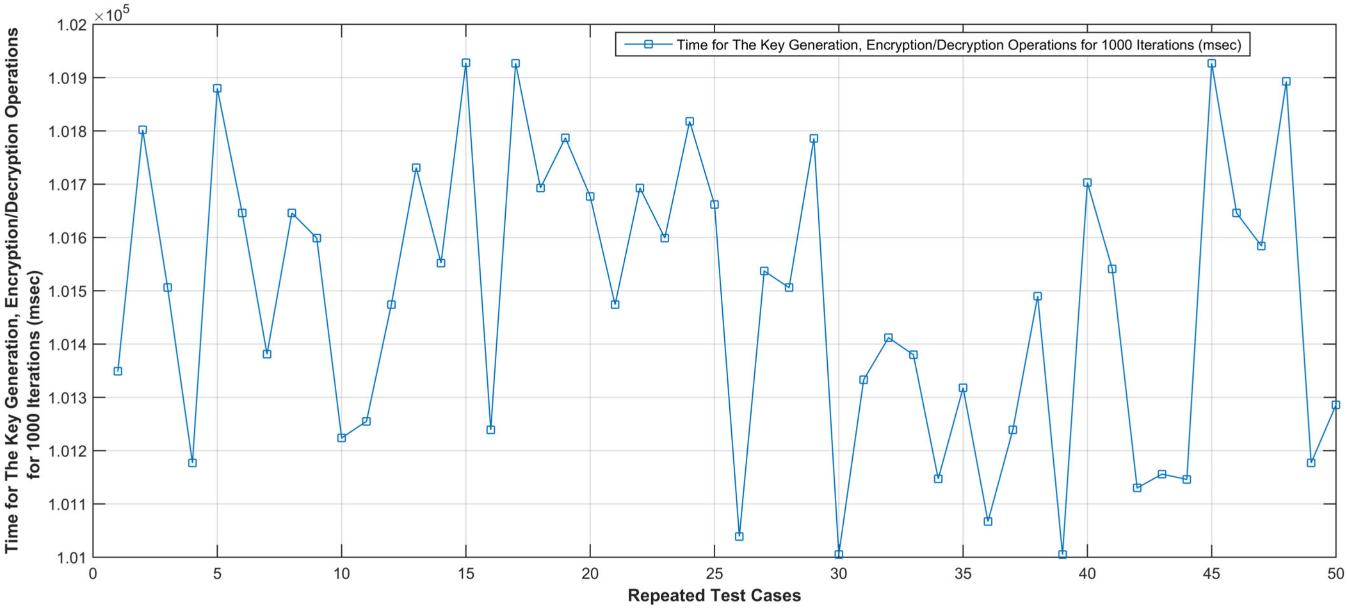

FIGURE 4 The necessary time for the key generation, encryption/decryption operations

TABLE 2 The necessary time without security calculations for communicating with the remote tools

Repeated test cases	1	2	3	4	5
Metadata retrieval time (milliseconds)	145.44	146.49	145.72	147.77	147.37
Standard deviation of metadata retrieval time	12.74	13.64	13.09	14.54	13.94

On the security perspective, the NVidia GTX 1080 graphic card can perform 51 500 ($\sim 51 \times 10^3$) PBKDF2 execution per second[39] whereas we can perform 10 PBKDF2 executions with our testing environment given in Section 4.2. Under the assumption that one NVidia Titan X graphic card is used with 65 536 ($\sim 65 \times 10^3$) iterations for obtaining the password from out of 251.68 password space, then the necessary time needed to reach the password will be:

$$\frac{2^{51.68}}{51 \times 10^3 \times 60 \times 60 \times 24 \times 365} \cong 2221 \ \text{years}$$

If instead of one, 1000 of NVidia GTX 1080 graphic cards are used parallel for obtaining the password, then the necessary time needed to reach the password will be:

$$\frac{2221}{1000} \cong 2 \text{ years}$$

As a result, our proposed password-based encryption approach can guarantee the security of user-related sensitive data located on a cloud environment based on the above calculations showing the minimum required time to obtain a user's password.

4.4 | Benefits of the proposed work over password-based authentication

Our proposed work allows users to login into the cloud system where their login credentials for social web tool can become available for accessing social web tools. After successful login to the system, encryption/decryption keys are generated for the logged user and the user's sensitive data will be accessible for being used by cloud services to communicate with the associated social web tools. The proposed work does not need to store users' passwords related to users' login. In other words, there is not any password file for managing authentication process. There exists a number of attacks in literature for password files that are used in existing password-based authentication systems (eg, brute force, dictionary etc.). It is easy to use these types of attacks in password-based authentication systems if a user's password is weak or a weak hashing algorithm is used for getting has a value of the password. On the other hand, our proposed work ensures the authentication of the users and the encryption of the user-related sensitive data located on a cloud. To provide these services, our proposed system does not contain any password file type structures. The users are authenticated when they are successfully login to the cloud system that integrates our proposed password-based encryption approach. During the authentication of the users, the encryption/decryption keys that are also used for encrypting/decrypting the users' credentials for social web tools are generated.

As a result, our proposed work guarantees the secure storage of the sensitive data in a cloud environment and provides authentication services. The security of the data stored on a cloud is crucial and users should not be able to access each other's data stored in a cloud environment. However, recent Meltdown and Specter attacks[6-9] make it possible to access data, which is stored on a cloud without any encryption, by other cloud users. In general, possible attacks to the sensitive data located on a cloud could lead to severe consequences. To prevent data from these types of possible threats, our proposed work defines an approach for maintaining the privacy and the security of the data. Eventually, it is obvious that storing data in encrypted form on a cloud is becoming more critical and gaining in importance.

5 | CONCLUSION

We introduced a novel architecture for a Password-based Encryption approach that stores data in an encrypted form on the cloud to protect the sensitive data. The Password-based Encryption Service is an add-on architecture that runs one layer above existing information service implementations.

To achieve data privacy and protection on the cloud, we have introduced and discussed our Password-based Encryption approach that is based on symmetric cryptosystems and provides the security of the user-related sensitive data stored on a cloud environment. The proposed work does not need to store users' passwords related to users' login. Another saying is that there is not any password file for managing authentication process. Users are authenticated through the various steps by password-based authentication approach using generated symmetric keys.

We performed a set of experiments to evaluate the performance of the Password-based Encryption Service to understand whether it can achieve information encryption with acceptable costs. We shared our experiment results for required minimal timing values for security calculations and the least time to obtain a user password.

With this research, we discussed the proposed Password-based Encryption approach for a cloud-based federation/unification service framework and its ability to handle privacy and security of the user-related sensitive data. We intend to further improve this approach by focusing on the various techniques to identify the unsuccessful login tries to prevent the attackers from trying to login to the system. As the development of cloud computing technology and security issues is still at an early stage, we hope our work will provide a better understanding of the design challenges of cloud computing and security needs, and pave the way for further research in this area.

ORCID

Ahmet F. Mustacoglu https://orcid.org/0000-0002-5236-3917
Ferhat O. Catak https://orcid.org/0000-0002-2434-9966

REFERENCES

1. Mell P, Grance T. The NIST Definition of Cloud Computing. 2011. Available from https://doi.org/10.6028/NIST.SP.800-145. Accessed March 16, 2020.
2. Yang X, Wallom D, Waddington S, et al. Cloud computing in e-science: research challenges and opportunities. *J Supercomput*. 2014;70(1):408-464.
3. Fox GC. Large scale data analytics on clouds. In Proceedings of the Fourth International Workshop on Cloud Data Management; 2012.
4. Kant DC, Sharma Y. Enhanced security architecture for cloud data security. *Int J Adv Res Comput Sci Softw Eng*. 2013;3(5):570-575.
5. Kumar A, Lee BG, Lee H, Kumari A. Secure storage and access of data in cloud computing. In 2012 International Conference on ICT Convergence (ICTC); 2012; Jeju Island.
6. Fox-Brewster T. forbes.com. 2018. Available from: https://www.forbes.com/sites/thomasbrewster/2018/01/03/intel-meltdown-spectre-vulnerabilities-leave-millions-open-to-cyber-attack/#6084f67c3932. Accessed March 16, 2020.

7. Greenberg A. Wired.com; 2018. Available from: https://www.wired.com/story/meltdown-spectre-bug-collision-intel-chip-flaw-discovery/. Accessed March 16, 2020.
8. Kocher P, Genkin D, Gruss D, et al. Spectre attacks: exploiting speculative execution. In 2019 IEEE Symposium on Security and Privacy (SP), 2019, San Francisco. p. 1–19.
9. Lipp M, Schwarz M, Gruss D, et al. Meltdown; 2018. Available from: https://arxiv.org/abs/1801.01207. Accessed March 16, 2020.
10. Lee HM, Jeong YS, Jang HJ. Performance analysis based resource allocation for green cloud computing. *J Supercomput*. 2014;69(3):1013-1026.
11. Armbrust M, Fox A, Griffith R, et al. A view of cloud computing. *Commun ACM*. 2010;53(4):50-58.
12. Liu H. Big data drives cloud adoption in Enterprise. *IEEE Internet Comput*. 2013;17(4):68-71.
13. Pandey S, Nepal S. Cloud computing and scientific applications—big data, scalable analytics, and beyond. *Future Gener Comp Syst*. 2013;29(7):1774-1776.
14. Warneke D, Kao O. Nephele: efficient parallel data processing in the cloud. In Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS '09). New York, NY: ACM; 2009.
15. Aceto G, Botta A, Donato W, Pescapè A. Cloud monitoring: a survey. *Comp Netw*. 2013;57(9):2093-2115.
16. Gunarathne T, Zhang B, Wu TL, Qiu J. Scalable parallel computing on clouds using Twister4Azure iterative MapReduce. *Future Gener Comp Syst*. 2013;29(4):1035-1048.
17. Durao F, Carvalho JFS, Fonseka A, Garcia VC. A systematic review on cloud computing. *J Supercomput*. 2014;68(3):1321-1346.
18. Bhargavi RV, Rao KN. Trusted third party framework for data security in cloud computing environment. *Int J Sci Res*. 2015;4(12):917-923.
19. O'Driscoll A, Daugelaite J, Sleator RD. 'Big data', Hadoop and cloud computing in genomics. *J Biomed Inform*. 2013;46(5):774-781.
20. Hawthorn N. Finding security in the cloud. *Comput Fraud Security*. 2009;2009(10):19-20.
21. Parakh A, Kak S. Online data storage using implicit security. *Inform Sci*. 2009;179(19):3323-3331.
22. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Commun ACM*. 1978;21(2):120-126.
23. Miller VS. Use of elliptic curves in cryptography. *Advances in Cryptology—CRYPTO '85 Proceedings*. Berlin, Heidelberg: Springer; 1986:417-426.
24. Turan MS, Barker E, Burr W, Chen L. NIST Special Publication 800–132. 2010. Available from: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf.
25. Arthur RB, Olsen DR. Privacy-aware shared UI toolkit for nomadic environments. *Softw Pract Exp*. 2012;42(5):601-628.
26. Itani W, Kayssi A, Chehab A. Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC'09. 2009; Chengdu, China.
27. Kamara S, Lauter K. Cryptographic cloud storage. In International Conference on Financial Cryptography and Data Security, Springer, Berlin, Heidelberg; 2010.
28. Kaufman LM. Data security in the world of cloud computing. *IEEE Security Privacy*. 2009;7(4):61-64.
29. Wang C, Wang Q, Ren K, Lou W. Privacy-preserving public auditing for data storage security in cloud computing. In Proceedings of the IEEE Infocom, 2010; San Diego, CA; 2010.
30. Yu S, Wang C, Ren K, Lou W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the Infocom, 2010; 2010, San Diego, CA.
31. Yoon YB, Oh J, Lee BG. The establishment of security strategies for introducing cloud computing. *KSII Trans Internet Inform Syst*. 2013;7(4):860-877.
32. Wang X, Ning Z, Zhou M, et al. Privacy-preserving content dissemination for vehicular social networks: challenges and solutions. *IEEE Commun Surveys Tutor*. 2018;21(2):1314-1345.
33. Wang X, Ning Z, Zhou M, et al. A privacy-preserving message forwarding framework for opportunistic cloud of things. *IEEE Internet Things J*. 2018;5(6):5281-5295.
34. Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research challenges. *J Internet Serv Appl*. 2010;1(1):7-18.
35. Inc. CS. Citrix Hypervisor. 2016. Available from: https://www.citrix.com/products/xenserver/overview.html. Accessed March 16, 2020.
36. KVM C. Kernel Virtual Machine (KVM). 2016. Available from: http://www.linux-kvm.org/page/Main_Page. Accessed March 16, 2020.
37. VMware. ESXi. 2016. Available from: https://www.vmware.com/products/esxi-and-esx/overview. Accessed March 16, 2020.
38. Mustacoglu AF, Fox GC. A novel digital information service for federating distributed digital entities. *Inform Syst*. 2016;55:20-36.
39. GitHub. 8x Nvidia GTX 1080 Hashcat Benchmarks. 2016. Available from: https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40. Accessed March 16, 2020.

How to cite this article: Mustacoglu AF, Catak FO, Fox GC. Password-based encryption approach for securing sensitive data. *Security and Privacy*. 2020;3:e121. https://doi.org/10.1002/spy2.121