Journal of Information Security, 2018, 9, 242-256
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

On the Significance of Cryptography as a Service

Nick Rahimi[1], Jacob J. Reed[2], Bidyut Gupta[2]

[1]Computer Science Department, Southeast Missouri University, Cape Girardeau, MO, USA
[2]Computer Science Department, Southern Illinois University, Carbondale, IL, USA
Email: srahimi@semo.edu, Jacob.reed95@siu.edu, bidyut@cs.siu.edu

How to cite this paper: Rahimi, N., Reed, J.J. and Gupta, B. (2018) On the Significance of Cryptography as a Service. *Journal of Information Security*, **9**, 242-256.
https://doi.org/10.4236/jis.2018.94017

Received: July 20, 2018
Accepted: September 1, 2018
Published: September 4, 2018

Abstract

Cryptography as a service is becoming extremely popular. It eases the way companies deal with securing their information without having to worry about their customer's information being accessed by someone who should not have access to it. In this overview, we will be taking a closer look at Cryptography as a Service. The ground we will be examining is the effectiveness of it for mobile/wireless and desktop computing. Since we will be looking at something that operates as a service, we will need to first cover the application program interface (API) basics [1] or standard software as a service (SaaS) [2]. Next, what exactly cryptography as a service means for each of the aforementioned platforms. Lastly, other possible solutions and how they compare to CaaS. For the purpose of this review, we will be looking at CaaS in a cloud environment since typical SaaS is used that way. Subsequently most cloud environments utilize a UNIX based operating system or similar solution, which will be the target environment for the purpose of this paper. Popular algorithms that are used in CaaS will be the final part that will be examined on the grounds of how they perform, level of security offered, and usability in CaaS. Upon reading this paper the reader will have a better understanding of how exactly CaaS operates and what it has to offer for mobile, desktop, and wireless users in the present and future.

Keywords

Cryptography, Software as a Service, Cryptography as a Service, Mobile, Cloud Computing

1. Introduction

Software as a service (SaaS) [2] [3] [4] is an emerging field in computer science that has been growing recently and is being adopted and implemented by many companies throughout the world. APIs in general are on the forefront of the world of technology today. Due to their popularity, they have been implemented for a variety of applications. This has led to a variety of different applications that have made breakthroughs in the way that we see software. Some of these applications include making backend, software, payment processing, front end, cloud as a service, or even the aforementioned cryptography as a service [5] [6] [7] [8]. The applications are endless and there are more emerging ones that are coming out all the time but the typical applications can be seen by referring to **Figure 1**. Cryptography as a service is an interesting example of this and will be the focus of this paper. By examining the functionality, effectiveness, and different possible applications we will be able to come up with an understanding of why cryptography as a service can be a good thing to have for a company's infrastructure [9].

The paper is organized as follows. In Section 2, we have briefly discussed the definition of SaaS, its implementation, and alternatives. Section 3 provides reader with the basics for understanding the exact reasoning, uses for cryptography as a service, and how it might possibly be implemented. Sections 4, and 5 discuss implementations of CaaS on mobile and cloud/desktop environments respectively. In Section 6, we have presented other alternatives to CaaS. Section 7 draws the conclusion.

2. Software as a Service

Before we get too deep into what Cryptography as a Service is, we must first take a look of what its older relative, Software as a Service entails. As mentioned before, software as a service is an ever-popular new platform for distributing

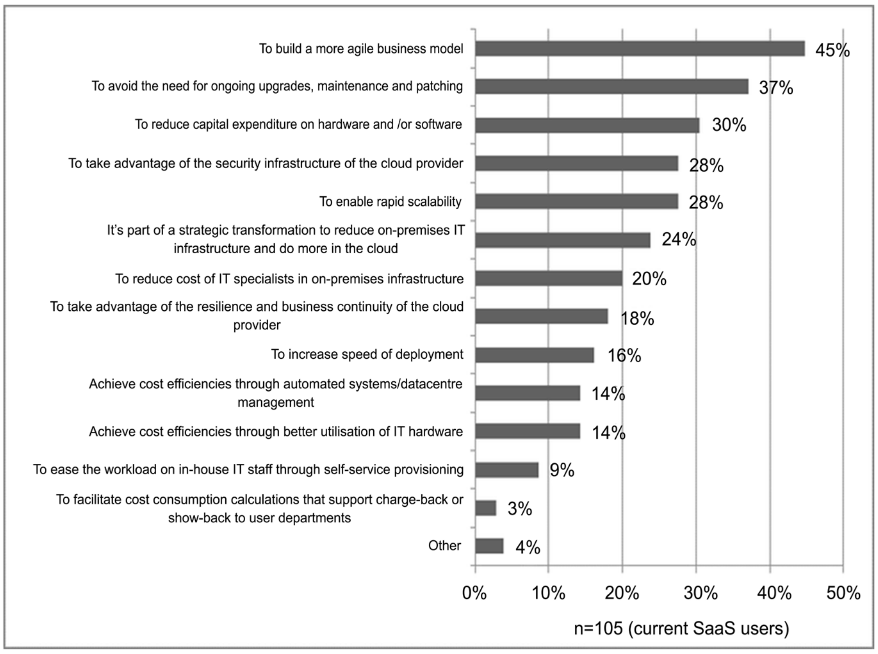


Figure 1. Reasons why companies use SaaS [2].

different kinds of software to clients that might be utilizing a cloud or non-cloud based environment.

When examining Software as a Service we find that this field is not trivial as it is relatively undefined and or abstract in implementation. Often times it is up to the provider to determine what the general design of the software should entail. This in turn opens up a variety of different definitions of what software as a service actually is. According to Gartner Research [2], the basic definition of software as a service is as the following statement, "Software that is owned, delivered, and managed remotely by one or more providers. The provider delivers software based on one set of common code and data definitions that is consumed in a one-to-many model by all contracted customers at any time on a pay-for-use basis or subscription based on use metrics". This definition can be used across almost all "as a service" products that many companies are reverting to as they offer a more cost effective approach to meeting their customer's needs without having to worry about possible extra costs that may include but are not limited to: media creating costs such as CDs, on site installation, travel for implementations teams, etc.

2.1. Implementation

As mentioned above, there are a lot of added benefits that come along with moving to an "as a service", one of them being its implementation. Typical software normally lives on the computers and might interact with a remote server or data center depending on the application. With software as a service that often is not the case. Often these systems are built around a specific client side consoles but often they come in the form of APIs based on open integration protocols such as HTTP, SOAP, and REST oriented services [10]. REST is usually the choice of many companies allowing clients to use GET, POST, PUT, DELETE, etc. request methods via HTTP to make changes or update data [10]. Many software as a service, implementations will follow a similar structure to the one that is depicted in **Figure 2**. This is the basic layout of the way that software as a service platform as a service, or even infrastructure as a service might be laid out. Interacting with the services is typically over the public internet through HTTPS for clients or SOAP.

2.2. Alternatives

With the structure and basic implementation in mind, it would be amiss to not go over the alternatives to this as it will be included along with the alternatives when we closely examine cryptography as a service.

First, there is the obvious one which is, do the work and implement the solution without relying on a third party's software, cloud, or infrastructure related offerings. While there are many out there that would choose not to do this, due to a lack of resources, there is a large audience that would take this route especially larger companies that want to maintain all responsibilities including

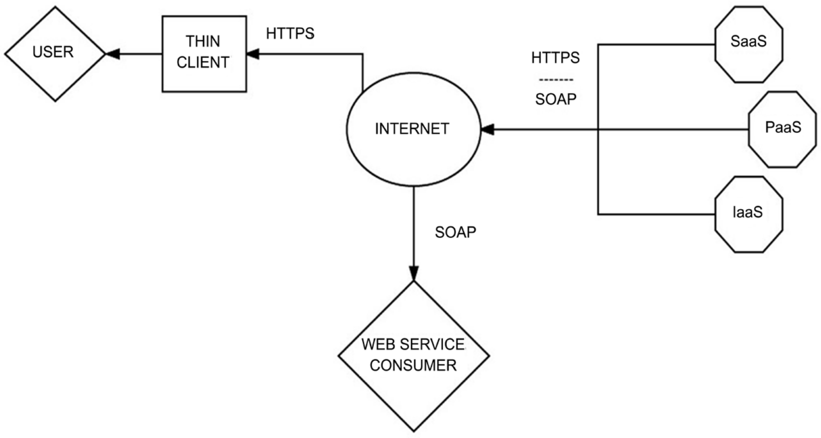


Figure 2. Typical API/SaaS structure.

managing their user's data in their own data centers or deriving the software solution on their own to prevent third party costs.

There is another looming reason behind why software as a service might not be the best choice. This might be due to the specific software not being tailored to fit the exact needs of the company. Typically, there are two paths to follow with software as a service:

1) Vertical Software as a Service.

2) Horizontal Software as a Service.

Vertical software as a service is geared to fit a specific need such as engineering, developer tools, finance, etc. while horizontal is focused on a specific industry such as software engineering, sales, etc. and has a much larger overall scope than with vertical software as a service [11]. If neither of these fit the needs of what the possible client is looking for it might not be a good fit for the company and it would be better for them to derive their own solution to better fit their and their own customer's needs.

3. Cryptography as a Service

In this section we will complete a brief overview of cryptography as a service. Cryptography as a service has been defined by Peter Robinson of RSA, the security of EMC as being, "Keyed cryptographic operations, such as encryption and decryption that are performed by a CaaS provider on behalf of a device via web service APIs" [12]. These will all be important points before we start examining how effective its implementation will be on different platforms.

3.1. Reasoning

There are a variety of different reasons why a company might want to utilize cryptography as a service. As mentioned before, a lot of companies might not want to put forth the extra effort in designing the extra infrastructure to accommodate such a feat as insuring that all information that is sent throughout their cloud based infrastructure is completely secured. This can take a consider-able amount of time when considering all of the work that would be involved with the creation of such a system that gets the job done fast and is also very effective in making sure that all of the information is properly secured. Such a feat might not even be considered possible for many smaller companies that have little staff or staff that is not well versed in proper information security standards.

3.2. Uses

There are a lot of different applications for cryptography as a service. One of the most popular is in cloud environments where there is a lot of data that is being handed back and forth between either micro services or other infrastructure standards that have been designed. As mentioned above, it is entirely dependent on the way that the infrastructure is organized and may need to be tailored to fit the needs of a specific client or clients that will be utilizing the environment. There exist a variety of different applications for such a service. We will be strictly focusing on cloud oriented cryptography as a service.

3.3. Implementation

We need to identify what the different types of security, cryptography as a service aims to address. These often include both protecting the user's information and those that are faced by the cloud service provider themselves when they are trying to implement software as a service, platform as a service, or infrastructure as a service [11]. These are the major points for concern with many cloud computing service providers to make sure that their customer's information is not compromised in any way, shape or form.

After this is done, the next step is to analyze practical ways that the information can be encrypted. This can be done using an attribute based encryption algorithm [13] such as Cipher-text-policy ABE (CP-ABE) or Key policy ABE (KP-ABE). Another option for encryption would be to use fully homomorphic encryption whilst the final option would be to try and utilize searchable encryption [11]. These will be explained later when effectiveness of Cryptography as a Service is examined.

The next and last step before actual implementation of the service would be to examine the structure of your infrastructure. Some key points to evaluate might include: "How many customers am I servicing if any?", "What sort of information am I storing?", "Can I really benefit from adding cryptography as a service to the infrastructure?" etc. These questions help determine what the system requires and whether this will have any benefit for the company's current infrastructure.

4. Mobile Platform

The very first platform that we will be examining for the overview of cryptography as a service, is the mobile aspect. This would be how user information is stored and accessed securely and what cryptography as a service means for that specific platform [14].

4.1. Possible Compromising

The Before we consider the specific means of protecting the information we must observe the way that a user's information might be compromised by a third-party attacker or other individual. These attacks, spoofs, etc. might come in the following forms that are listed:

1) Compromised keys or data that is stored on a person's device.

2) An attacker steals a backup of the user's device.

3) An app exists on a user's device with the sole purpose to steal the user's information in the form of malware, key-logger, or another spyware.

4) Malware app compromises entropy sources which result in more easily guessed keys [12].

These are all different possible attacks that can be carried out by attackers and there are others that are less common and usually are not geared or created for stealing user's information but might be used for making the phone unusable or severely affect the usability of device. An example is the text that could be sent to iPhones during the late part of 2016 that would cause the device to power cycle. This is where cryptography as a service comes into play.

Most of these attacks can be thwarted, for the most part, by implementing a secure system as cryptography as a service. This is especially true for the ones that have an emphasis on stealing user's information. The next portion will examine exactly how this is carried out by cryptography as a service and what it means for device and user security.

4.2. Implementation

Earlier in the paper we covered a high level overview of what exactly cryptography as a service is and possible ways that it might be used. For this implementation we will be looking at the implementation of such a service. We will also be delving a bit deeper into what makes this a service.

As discussed earlier, the way that the "as a service" architecture works is through the implementation of HTTP and systems such as REST and SOAP. REST is commonly used for cryptography as a service especially since we are examining the mobile application of such a service. The overall architecture is extremely reminiscent of Public Key Authorities (PKA) and Certificate Authorities (CA). By examining **Figure 3** we can see the obvious similarities.

As can be seen it is extremely like the structure of a PKA or CA. The steps are different and might be for different applications whenever needed.

Process:

1) The data is first sent to the cryptography service to be encrypted by the service using a chosen algorithm which, in this case, is AES.

2) The service receives the data and then will generate a key that will only be known to the service and not to either the recipient or the sender so possible compromise of the data is not possible.

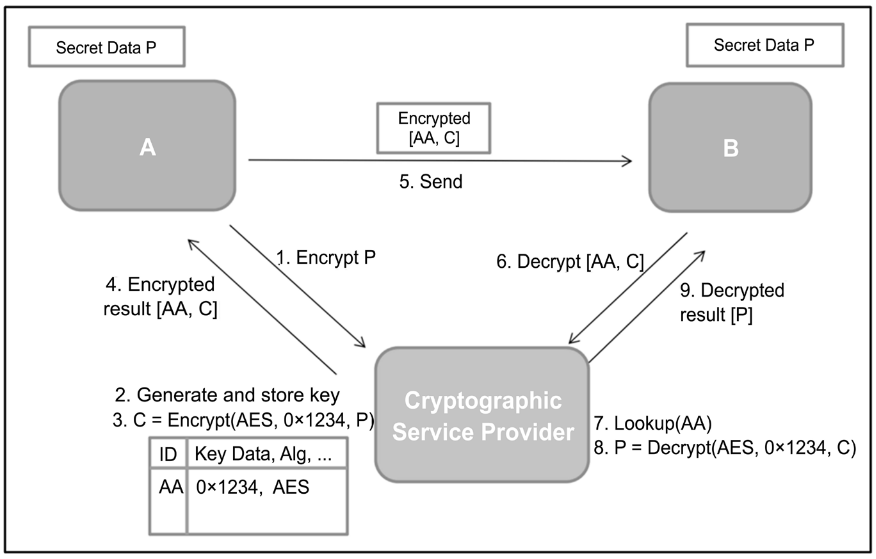


Figure 3. CaaS is extremely reminiscent of PKAs and CAs [11].

3) The message is then encrypted by the service using the chosen algorithm to generate the secured data.

4) The encrypted data is sent back to the sender so it can be sent with an ID that was generated by the service.

5) The encrypted data with the ID is sent to the recipient.

6) The recipient then sends the encrypted data to the service where it will be decrypted.

7) The service checks the ID to validate it to make sure it was not tampered with during transmission.

8) The data is decrypted using the same algorithm used for encryption.

9) The decrypted information is then sent back to the recipient over a secure channel.

This is the typical structure that is used for securing information. In **Figure 3**, the two parties might be mobile devices that might be sending anything from chats to actual files between each other. All users that are using the service or application also need to be authenticated. This handles the problems with securing data and making sure that information is not tampered with or visible to those who are not authorized to see the information.

As mentioned above the typical algorithm of choice is usually AES using an endpoint encryption method as the one depicted above due to the robustness of the algorithm. Another possible way that this can be done is through the generation of a signature by sending the ID from one of the endpoints to the cryptographic service with the data, the service then fetches the key for that ID, create a signature using the ECDSA/SHA 256 algorithm, and finally send the signature back to the endpoint to be used over a secure channel [12].

The next step is to examine the various applications and uses for cryptography as a service for the mobile devices. By examining these applications, we can then see if this method is a good use for cryptography as a service and how effective it is.

4.3. Possible Applications

There are a lot of possible applications as mentioned earlier. Since it has a variety of usages, its popularity has increased exponentially over the years that it has been available. None of these applications revolve around using just the encryption and decryption aspects of cryptography. These might include but are not limited to creating of signatures for different devices that want to share information or even just strictly encrypting end to end. A more comprehensive list includes [12]:

1) *Signatures*: A difficult code to replicate that can be used to identify information and or the sender.

2) *Simple Encryption*/*Decryption*: Encryption of the user's information using the methods mentioned in the previous section.

3) *Sophisticated Encryption*/*Decryption*: Like the previous point but utilizes more complicated encryption algorithms that might be used to secure more sensitive data that cannot be compromised at all.

4) *Bulk Encryption*/*Decryption*: This is different since instead of small amounts of data being encrypted there is a lot of data to be encrypted. This might be to secure large collections of data that might be only sent to the service and then directly back to the serviced party.

For the most part the only ones typically utilized by mobile applications are the first three. The last one will be mostly for applications that are strictly cloud oriented in nature. As mentioned in the previous section, we can see how each of these might be implemented.

Now that we have briefly covered the different applications, we can start to look at the actual effectiveness and the possible problems that might be encountered with this solution. Other possible alternatives will be discussed later in this paper as they will include the desktop and cloud oriented methods of implementing cryptography as a service.

4.4. Effectiveness and Possible Issues

So far, we have covered the basics for implementing cryptography as a service for mobile applications. The next thing that we need to cover is how effective using cryptography as a service is for mobile applications. This may include but is not limited to speed, efficiency, and then of course other problems that might arise from trying to implement such a service.

There are many advantages and disadvantages of using cryptography as a service for mobile applications. Most of the issues for cryptography as a service are all encompassing which means that they tend to apply for all the different applications that it has. For this section, we will try to focus on those that are oriented towards mobile applications. The issues include but are by no means limited to overall latency, network connectivity, and authentication [12]. These all can severely impact the service. First, let us look at each of the possible issues in detail.

1) *Latency*: With making a lot of web oriented calls to a service the overall service can be severely impacted making the application less responsive and slow to use, hurting the overall user experience.

2) *Network Connectivity*: This is mandatory for the application to properly secure the user's information. Of course, this will be needed anyways to send the data to a possible recipient but weak signals may impact how effectively the application works overall.

3) *Authentication*: This is needed to make sure that only authorized users have access to the service. While it is necessary, it also can pose a problem to developers. This means taking the proper steps to authenticate users at appropriate points throughout the whole process.

This is by no means a comprehensive list that covers all the difficult challenges to overcome.

Finally, for mobile, we can cover the overall effectiveness of cryptography as a service for mobile applications. Regardless of the actual performance, with the added layer of security there is improved security since valuable items such as keys are not stored on any of the endpoints which in this case would be the mobile devices. One of the possible benefits of implementing the service is that it might allow for better performance or force better coding and design practices to make sure that the application is responsive enough. Overall, it can be said that cryptography as a service is worth implementing for mobile applications if the company would rather utilize a service instead of implementing one themselves.

5. Cloud/Desktop CaaS

In this section the application of cryptography as a service will be introduced and follow the same structure as the previously covered mobile application portion. Since a lot of the concepts discussed earlier, at this points, they will not be reintroduced.

5.1. Basic Overview

In this section the application of cryptography as a service will be introduced and follow the same structure as the previously covered mobile application portion. Since a lot of the concepts discussed earlier, at this points, they will not be reintroduced. The first thing we need to do is cover the basic structure of cryptography as a service for cloud computing. First, the different parts of the structure need to be discussed. We can start with the service itself. For many cloud applications, it is divided into virtual machines that could be running a variety of different services that are managed centrally [15] [16]. Once again, this can vary platform to platform and company to company. Since this is the case we will be examining one implementation of the service and its structure that seems to be a popular and standard implementation of the service itself. Refer to **Figure 4** for

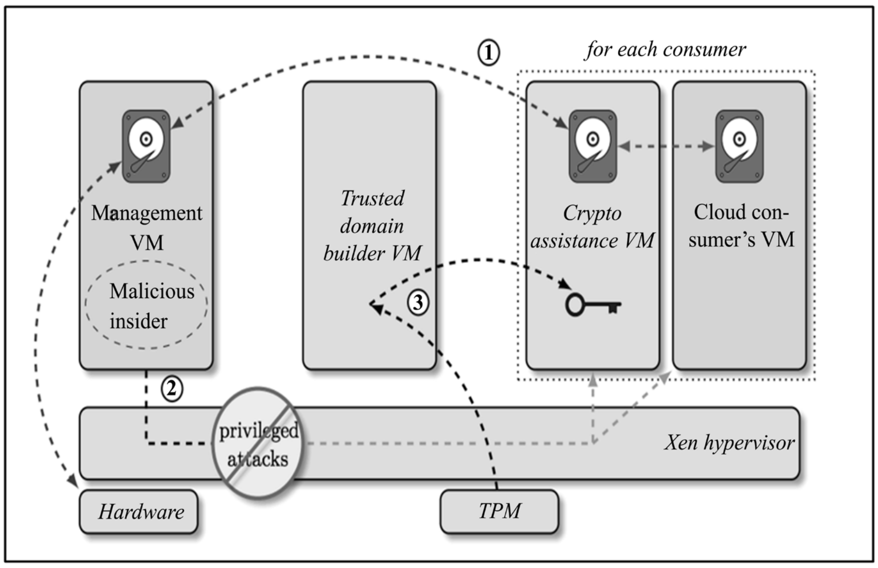


Figure 4. Basic CaaS structure for cloud [17].

an idea of the overall structure. **Figure 4** shows the basic architecture of the algorithm that is typically implemented. The key refers to the key used for the pass-through encryption [17]. Each part of the structure will then be covered in a more detailed fashion to avoid any confusion.

Now that the basic layout is given, each part needs to be covered to avoid further confusion. The different parts of the overall structure break down into about three main parts. Each of these parts is labeled inside of the diagram in **Figure 4** which will be the basis for the next portion of the paper where the functionality of each will be discussed to provide a basic understanding of how CaaS can work in a cloud environment.

Portion #1 shown in **Figure 4** is called the cryptographic assistance domain. This portion of the structure is separated from the rest of the overall structure as it will be the way that the client interacts with the service. This needs to be supplied with the appropriate keys from the cloud consumer to be used. By separating this from the rest of the overall structure, security is insured. Even in the event where the keys are safe guarded during runtime from attackers. This is the case since keys are directly accessible during runtime even in the case of a full breach.

Portion #2 represented in **Figure 4** is the management portion of the service itself. This is done through the implementation of a management virtual machine (VM) and a hypervisor which in this case is a Xen Hypervisor. The reasoning for the implementation is that it prevents privileged attacks from being carried out against the other VMs that are a part of the service. By having this access control, privileged attacks/insider attacks can be thwarted from being carried out through hyper-calls. This means a typical task must be moved to the third portion of the structure which is the trusted domain builder.

The last portion of the overall basic architecture is the Trusted Domain Builder. As aforementioned when talking about the last portion of the architecture, the access control is restricted and therefore does not have the ability to start and stop machines. The tasks of starting, stopping, and suspending the machines is then left up to the trusted domain builder which can have those special permissions. The keys (HVKs) are encrypted by the TPM so that the trusted domain builder is the only one that can decrypt them. Theses keys are then sent to the consumer's cryptographic assistance domain or DomC [18]. This portion then does not require the implementation of the verification portion of the management VM.

5.2. Implementation

The overall basic implementation is similar to that mentioned earlier in the paper. The interactions are very similar to that in regards to how the customer or client will implement the service. Since these structures could possibly be implemented in a variety of different ways, its implementation would then be highly dependent on the overall structure of the service. This make implementation abstract as it does not follow a basic set of guidelines as has been the theme of this analytical overview of cryptography as a service [19].

5.3. Algorithms

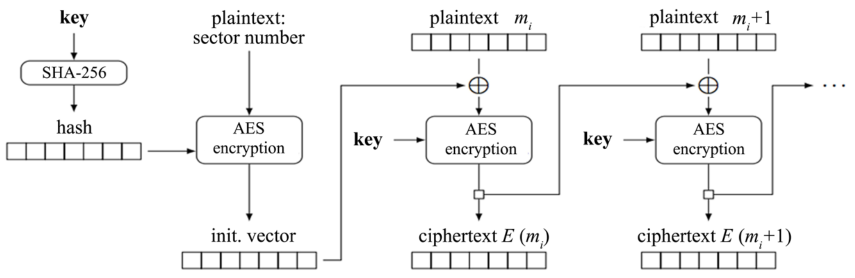
Another important topic that still needs to be covered when analyzing cryptography as a service for the implementation of it in cloud computing is the algorithms that are utilized. This will be key in identifying how effective the system is as the algorithms play a key role in the speed, responsiveness, and the effectiveness of the implementation. For this paper, we will be examining the implementation of algorithms based on the architecture depicted in **Figure 4**.

The first step we need to do is identify the possible algorithms. The typically used algorithms for this architecture include, for the pass-through encryption discussed earlier, an *Encrypted Sector-Salt Initialization Vector* (ESSIV). ESSIV is a simple algorithm that does not have any known vulnerabilities. If there were though the implementation could easily be changed to fit the needs and make CaaS safe again without too much effort. For this section ESSIV with Cipher Block Chaining (CBC) will be considered. This is the same algorithm that has been used for the Linux *dm-crypt* kernel driver. The formula for the algorithm is the following when calculating the initialization vector (IV).

$$IV(i) = \varepsilon_s(i), \text{ where } s = \text{hash}(k),$$

where i=target vector [20].

After the IV has been calculated, the result should be a length that can be used in conjunction with SHA-256 and AES. As can be seen, AES is used in both the cloud and mobile applications for CaaS. See **Figure 5** for details regarding the algorithm. **Figure 5** Algorithms are familiar to those who have used block ciphers before but include the implementation of SHA-256 and the sector number [17].



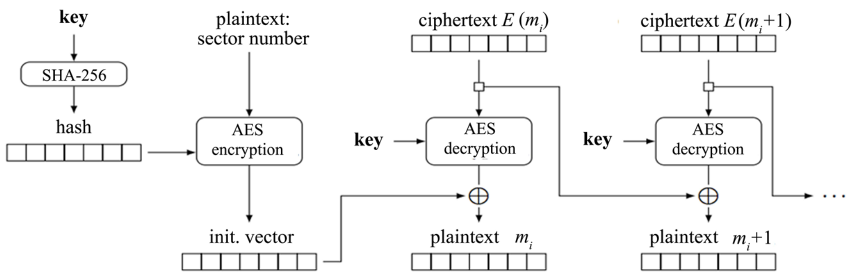


Figure 5. Encryption and decryption algorithms [17]. (a) Encryption operation; (b) Decryption operation.

5.4. Possible Implementation Issues and Effectiveness

As with all infrastructure implementations, especially software as a service, we can expect there to be issues that might hurt the overall effectiveness of the software and or the service. To have a thorough overview speed, security of the system, and responsiveness need to be examined.

We can first start with the security aspect since that will be most important when convincing companies or other clients that they should be using the service for their infrastructure. The one issue with the entire structure, in terms of security, is the key management. This is in part since there really is not any control over the dataflow security when transferring to different steps within the service. This can possibly be solved by assigning key management to a third party but in turn might be difficult to find a third party to properly manage the keys the way that the infrastructure requires.

It is also important to consider the speed at which service can operate. Due to the use of CBC and overall block ciphers can be detrimental to the speed of the service. There are a variety of different factors that can be attributed to speed issues. These can include hardware capabilities, type of algorithm, and/or networking capabilities as mentioned previously when mobile CaaS was introduced. Even though "remote" encryption and decryption might not be the fastest solution that is already known to those who are interested in the implementation beforehand and many still utilize the service regardless. Many see it that the speed is not as important as ensuring that the customer's user's and client's information is protected.

The last thing to be discussed for this portion of the paper is the overall effectiveness of the service for cloud computing applications. It is very effective in ensuring that user's information is kept secure and private from prying eyes. This creates an extremely popular and effective service that many are utilizing through a third party. As mentioned before, the speed might not be the best but the tradeoff is making sure that the information is kept safe which is exactly what cryptography as a service does for cloud environments and distributed applications.

6. Alternatives

When considering a solution to a problem it is important that we observe all options and what the advantages and disadvantages might be of each possible solution. This is to make sure that what you are choosing is the absolute right choice for a client that will fit their and the company's goals to ensure user information is secure. The other possible solutions that will be briefly discussed are Security Enforcement Mechanisms (SEMs) and custom built solutions [21]. These are the two typical other forms of thwarting that companies may implement to protect user's information and data.

SEMs include items such as firewalls, safes, guards, cryptography software, etc. Even though this might introduce cryptography into the equation, the overall implementation might be totally different. This system can be better in a variety of different ways. It has the capacity to be more responsive, allows better design decisions, etc. On the other hand, it might be lacking in some areas. Unlike CaaS, the company will be tasked with configuring the networks with the hardware solution and then a security team to create and configure a software solution. This is added overhead that must be tackled if this is the chosen route over CaaS [19].

The last alternative before the conclusion that will be observed is creating our own solution. When this is done, a company must build their software from the ground up and might require extra time for tweaking and making sure that all of the bases are covered. On the other hand, it beats CaaS in the realm of the level of flexibility. The party can then customize the software to meet their exact needs instead of trying to make CaaS work for what they have or want. This means their choice of architecture, algorithm(s) used, etc. CaaS does come out as being ready to use out of the box for the most part and maybe be more convenient in certain situations.

7. Conclusion

In this paper we discussed cryptography as a service for both the mobile and cloud environments. We have seen that CaaS is extremely flexible and can be implemented in a variety of techniques. Since this is the case, it means that this might not be the best solution. Overall, CaaS is a growing and emerging field currently and is being implemented by many companies across the globe and might change the way that information security is viewed in the future.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Turner, M., Budgen, D. and Brereton, P. (2003) Turning Software into a Service. *Computer*, **36**, 38-44. https://doi.org/10.1109/MC.2003.1236470

[2] Gartner IT Glossary (2017) Software as a Service (SaaS) from the Gartner IT Glossary. http://www.gartner.com/itglossary/software-as-a-service-saas/

[3] Buxmann, P., Hess, T. and Lehmann, S. (2008) Software as a Service. *Business Informatics*, **50**, 500-503. https://doi.org/10.1007/s11576-008-0095-0

[4] Godse, M. and Mulik, S. (2009) An Approach for Selecting Software-as-a-Service (SaaS) Product. 2009 *IEEE International Conference on Cloud Computing*, Bangalore, 21-25 September 2009, 155-158. https://doi.org/10.1109/CLOUD.2009.74

[5] Miller, S.K., Mandato, F.E. and Gursahaney, S.K. (1995) System, Data Processing Method and Program to Provide a Programmable Interface between a Workstation and an Archive Server to Automatically Store Telephone Transaction Information. US Patent No. 5402474.

[6] Berson, T., Dean, D., Franklin, M., Smetters, D. and Spreitzer, M. (2001) Cryptography as a Network Service. *Proceedings of the ISOC Network and Distributed System Security Symposium* (*NDSS*).

[7] Bugiel, S., Nurnberger, S., Sadeghi, A. and Schneider, T. (2011) Twin Clouds: An Architecture for Secure Cloud Computing. In: De Decker, B., Lapon, J., Naessens, V. and Uhl, A., Eds., *Communications and Multimedia Security*. CMS 2011. Lecture Notes in Computer Science, vol 7025. Springer, Berlin, Heidelberg.

[8] Schneier, B. (2007) Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, New York.

[9] https://www.tutorialspoint.com/http/http_requests.htm

[10] Robinson, P. (2017) Cryptography as a Service. https://www.rsaconference.com/writable/presentations/file_upload/adsr01-cryptography-as-a-service.pdf

[11] Robinson, P. (2017) Applying Cryptography as a Service to Mobile Applications. https://www.rsaconference.com/writable/presentations/file_uplad/csv-f02applying-cryptography-as-a-service-to-mobile applications_final.pdf

[12] Kumar, K., Liu, J., Lu, Y.H. and Bhargava, B. (2013) A Survey of Computation Offloading for Mobile Systems. *Mobile Networks and Applications*, **18**, 129-140. https://doi.org/10.1007/s11036-012-0368-0

[13] Bethencourt, J., Sahai, A. and Waters, B. (2007) Ciphertext-Policy Attribute-Based Encryption. *IEEE Symposium on Security and Privacy*, Oakland, 20-23 May 2007, 321-334. https://doi.org/10.1109/SP.2007.11

[14] Berson, T., Dean, D., Franklin, M., Smetters D. and Spreitzer, M. (2017) Cryptography as a Network Service. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.212.3075&rep=ep1&type=pdf

[15] Gampala, V., Inuganti, S. and Muppidi, S. (2012) Data Security in Cloud Computing with Elliptic Curve Cryptography. *International Journal of Soft Computing and Engineering*, **2**, 138-141.

[16] French, G. and Cvrcek, D. (2017) Cryptography as A Service Barclays Crypto Application Gateway and Beyond. http://spi.unob.cz/papers/2013/2013-08.pdf

[17] Hossein, R., Elankovan, S. and Zulkarnain, A. (2017) Encryption as a Service (EaaS) as a Solution for Cryptography in Cloud. Elsevier, New York.

[18] Takabi, H., Joshi, J.B. and Ahn, G.J. (2010) Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security & Privacy*, **6**, 24-31. https://doi.org/10.1109/MSP.2010.186

[19] Parno, B., Kuo, C. and Perrig, A. (2013) U.S. Patent No. 8,352,738. U.S. Patent and Trademark Office, Washington DC.

[20] Aw Ideler, H. (2012) Cryptography as a Service in a Cloud Computing Environment.

[21] Hamlen, K.W., Morrisett, G. and Schneider, F.B. (2006) Computability Classes for Enforcement Mechanisms. *ACM Transactions on Programming Languages and Systems*, **28**, 175-205. https://doi.org/10.1145/1111596.1111601