Mobile client security architecture: a practical approach

1. About the protection of cloud.
2. Mobile environment problems.
3. Offline mode problem.
4. Other works.
5. Our proposed methods: SSS, ABE, MOS
6. Our architecture.
7. Detailed workflow description.
8. Analytics in detail.
9. Common dangerous scenarios.
10. Analysis of security and complexity.
11. Implementation.

Cloud computing is a new paradigm in the world of distributed networking and computation. The basic feature of the cloud environment is providing the elastic, on-demand and secure service for the end-users. While the first two requirements are rather well supported by the cloud platforms in use, the security is the major concern of the cloud providers and governmental organizations as well as academia and research community [1], [2], [3]. For the small and medium-sized enterprises (SME) the cloud environment is often the most cost-effective and easily scalable solution. However, the security and privacy of the sensitive data in the cloud is a major issue.

A common practice to provide a stable security cloud solution is to use the specific type of cloud service: CASB – Cloud Access Security Broker or CAC - Cloud Access Control. These services are specifically designed to bring the security at a single access point and provide the coordination of the most important security measures. It is measured by Gartner report [4] that such systems will be used by 85% of companies by 2020. The reason is that the organization of the security measures at a single control point allows to control and monitor the level of cloud protection much more effectively. The basic features of the CASM are discovery of cloud services, encryption (along with tokenization for better search properties), access control, DLP services, authentication and auditing/alerting services [5]. The protection of the confidential data, according to the standards of CASB deployment should be provided at rest, in transit and in use [6].

The additional security problems and requirements need to be considered when the mobile devices are actively used in the cloud environment [7]. Today we live in the BYOD (Bring-Your-Own-Device) world and the mobile devices pose a serious risk to the SME cloud platforms as the bottleneck of the information security system (ISS). While the enterprise web interface and the cloud environment can be protected by the powerful third-party services such as CASB and CAC, the mobile client is usually light-weighted and generally less protected. The protection scheme used on a mobile device should be both computationally secure as well as resource-constrained due to the battery power limitations [8]. Therefore, the encryption on a mobile device is not a good solution: the proposed schemes are computationally good but lack the security analysis in many cases [9]. The common practice is the shadow user activities monitoring [7]. However, one area of the mobile device usage stays unprotected in all the proposed scenarios.

Suppose, SME uses CASB in order to protect data at rest (server protection), in transit (communication with server), in use (while the client is connected to the network). But what happens when the mobile client goes offline? And even worse - with some sensitive data on board? All powerful cloud-based tools cannot help and the mobile app has to secure itself with its own limited resources. Moreover, the critical point is the difference in strategy of online and offline protection. Due to the resource constraint, we cannot perform the extensive computation and encryption on a device.

In this paper, we consider the concept of the offline mobile client security. We propose a novel approach based on the powerful cryptographic preventive methods such as secret sharing [10] and ABE encryption [11]. Also, we propose to use the analysis of user behavior in order to reduce the risks and harm of the most common threats: the expired user misusing password or the intruder attack. We use the key expiration period and incorporate it safely into our system. Our main target is providing a maximum defence at the minimal resource cost.

The paper is structured as follows. In section 2, we analyze the most common security problems in the mobile cloud environment and the solutions proposed. We mainly concentrate on the offline protection in the BYOD world. In section 3, we give the basic definitions and explanations of the methods used: SSS, ABE, MOS. In section 4 we present our complete solution to the problem of offline mobile client security. In section 5 we trace the workflow of the client activity and analyze the common security breaches. We discuss security proofs for our system in section 5. Section 6 is the practical implementation and analysis of complexity.

**The previous works: problems and solutions**

The BYOD world requires slightly more from the traditional data protection services. Apart from authentication, DLP services and encryption (at rest in transit, in use) it is necessary to provide some additional contextual methods []. The contextual methods increase the security of the app at maximum at the minimum resource constraint. The traditional ones are:

1. Using geo location
2. Expiration of an app
3. Secured transfer between apps
4. Restricted access to app
5. Expiring pass/pin
6. Count of ailed tries
7. Offline protection approach.
8. Audit

Therefore, we can make a conclusion that it is harder to protect data on the mobile device so the owner or the SME should take more care of protecting its data leaving the organization. And the data on a mobile device should be considered as one leaving the organization. The most sensitive and confidential data should not be allowed to the mobile device at all. But what if the SME needs to allow the workers to work on such devices and even use the offline mode for the convenience and traffic reduce?

*From the theoretical point of view on this problem we consider several surveys []. The mobile cloud computing is a rapidly developing paradigm that poses many security and complexity problems. This type of systems requires new models of application and the new way of using the data storage services.* An analysis of the existing models of mobile cloud computing is presented in [8]. All the models and protection schemes concentrate on the encryption properties and either perform the computations on their own [12-15] or use the cloud provider to off-load the cryptographic operations [16-18]. It is natural that the mobile device can’t handle the operations securely without assistance of a cloud provider, due to the resource constraint and battery limitation.

When we want to make the device more independent and less dependable on the cloud provider (suppose, an application needs to run securely without connection to the network) we can use only the schemes that function without putting load on a provider. All the currently known schemes of encryption, performing the computation, as it is shown in the work [8] either use the a cloud provider [12] third party trusted agent [13] combination of both [14] or concentrate on computational complexity without taking care of user privacy and security[15].

In other words, the security schemes proposed, are not working offline. In many cases the industrial providers of the secured application wrap (like Mocona, operating along with SAP [16]) prefer to completely forbid the offline access to the protected app.

In some cases such access is still necessary. Due to such constraints as traffic load, travelling, ease of access and many more. So the question is how to protect the device/app, in an offline mode when the functions of data protection cannot be offloaded to a cloud or a trusted party. This problem was not approached neither in academia nor in the industry.

**Offline mode**

We propose an open model of the mobile device protection mechanisms where the security is supported both in online and offline modes. Currently, the systems of mobile device protection, to our knowledge, follow the model where the protected mobile client can operate only when it is connected to the cloud, which is not always convenient for the end-user. The basic principles of the mobile device protection that we support are:

1. Optimized communication with the cloud when the device does not need to be constantly connected to the server due to the resource constraint and necessity to secure this communication.
2. Implementation of reliable cryptography standards, i.e. the algorithms that we use (AES, ABE, SSS) are approbated and well-known. We do not invent on obscure concepts and do not invent the new methods that should be evaluated.
3. Optimized combination of the security mechanisms so that the mobile client does not need to perform complex computation like encryption and key generation due to the resource constraint.

The most important security issues in our model arise when the device goes to the offline and the user is still allowed to get the access to the strategic organization documents. In this case, the server can neither monitor the user activity nor provide the protection methods. The security becomes the responsibility of the mobile client. Additionally, the maximum protection should be provided at the minimum resource cost.

In the online mode the device communicate with the server in order to check the validity of user password. On the contrary, the offline protection should implement the different approach. In other words, the authentication/authorization mechanisms in the offline mode should utilize the specific proof of the user identity. The requirements for the proof are as follows:

1. The proof is derived from the previous session in order to verify that the user is still authorized.
2. The proof should not give access to user password, i.e. it should not be stored on the device.
3. The proof shoud be temporary and have expiration period.
4. It should not be directly used in communication with the server in order to prevent the malicious user from mimicking
5. It should be resilient to the off-line dictionary attack
6. It has to stay effective both in the scenarios of the malicious outsider and leakage of information when the formerly authorized user leaves the group.

If such proof can be constructed, then the offline mode can be secured. The most important requirement is that the password (or the proof of the password) cannot be stored on the client device, as we cannot guarantee that it is perfectly protected in this case. The client cannot be in possession of the user knowledge as these are the separate entities. Rather, this proof should be shared between the client and the user in a secure manner. This is the only way the client can support the security without performing complex computation or storing the function of the user password.

In our mobile client the user keys are protected by the combination of user password and PIN as well as the ABE keys that have an expiration period. The additional argument against the traditional password verification is the necessity to check the PIN which is very small so the construction of valid one-way function resilient to the offline dictionary attacks is a difficult task.

(HERE WILL BE A CHAPTER WITH DEFINITIONS AND NOTATIONS)

**Offline mode: our solution**

We propose the following workflow of the mobile application in the offline-mode:

The approach is based on the combination of AES-ABE-SSS methods. The key feature of the approach is that the client does not actually store any part of the user password to be verified. The client combines its own key with the user share (PIN and password-derived) in order to restore the initial KEY\_SET\_KEY. If the user provides the wrong share the client will not be able to recognize it, but will decrypt the files incorrectly.

*Security analysis*

The client doesn’t store the user password i.e. no information of the password leaks, and also there is no possibility for the malicious user to check if the password he is trying to enter is correct. The only possible scenario for the information leakage in the case of the malicious outsider is:

* 1. The hacker steals all the parts of KEY\_SET\_KEY
  2. The hacker steals the KEY\_SET
  3. The hacker try the brute force offline dictionary attack on all the previous values, they have to belong to ONE TIME SESSION (the values of a and b belong to one period of time i.e. TIME, DEV\_PASS, KEY\_SET).
  4. Steal the permanent FILE\_KEY (this is protected by the KEY\_SET).
  5. Steal the file and try to decrypt it with offline dictionary.

The hacker has to get 4 values: TIME, DEV\_PASS, KEY\_SET, FILE\_KEY - from one session. At the same time he should try the offline dictionary attack on PIN+PASS. Moreover, the 4 values provide access only to 1 single file. So practically, it is very difficult to perform such attack due to the key expiration period.

The temporary nature of all parameters obliges the user to connect to server when necessary and prevents the malicious actions from the user side. The only possible scenario when the user wishes to prolong his old credentials is:

* 1. He has to steal the DEV\_PASS and TIME synchronized with his credentials.
  2. He has to be able to combine.
  3. He has to steal the KEY\_SET synchronized with his credentials.
  4. He has to steal the protected FILE\_KEY (also synchronized) and file.
  5. He has to do everything without the client (because the client checks the TIME and renews the PUBLIC\_SHARE\_KEY)

Basically for malicious user there is practically no way to use the client with the old keys. The fact that client doesn’t contain any check data doesn’t prevent the user from seeing the contents of decrypted files but with the wrong password the decrypted files will be totally wrong.

The client still has to count the tries (to prevent the hacker brute force attack) within one session.

*Complexity analysis*

The client actions are:

1. Combine the PASS+PIN+TIME+DEV\_PASS=KEY\_SET\_KEY ------- SSS secret restoring
2. Decrypt the KEY\_SET with the KEY\_SET\_KEY --------- symmetric decryption
3. Select the SHARE\_KEY from the KEY\_SET -------------- no calculation
4. Decrypt the FILE\_KEY with the SHARE\_KEY ---------------- ABE decryption
5. Decrypt the file with the FILE\_KEY ----------------------------symmetric (AES) decryption
6. Modify the TIME periodically ------------------------------ timer
7. Count the tries within the TIME ---------------------------count
8. Modify or delete PUBLIC\_SHARE\_KEY

From this analysis we can see that the client does not perform the complex calculations and does not use to many resources due to the fact that the initial key is shared and the client performs only decryption which is not a time-consuming operation. We support the concept of the light-weighted client, i.e. the most consuming operations are ABE and AES decryption.

*(maybe now I should separate from previous)*

How to secure the data when the user leaves the group.

Reduce power consumption

Improve reliability

Enhance processing

Shamsi, J., Khojaye, M.A., Qasmi, M.A.: Data-intensive

cloud computing: requirements, expectations, challenges,

and solutions. Journal of Grid Computing **11**, 281–310

(2013)

2. Khan, A.R., Othman, M., Madani, S.A., Khan, S.U.: A

survey of mobile cloud computing application models.

Communications Surveys & Tutorials, IEEE **16**, 393–413

(2014)

3. Kumar, K., Lu, Y.H.: Cloud computing for mobile users:

Can offloading computation save energy? Computer **43**,

51–56 (2010)

4. Lorido-Botran, T., Miguel-Alonso, J., Lozano, J.A.: A

review of auto-scaling techniques for elastic applications in

cloud environments. Journal of Grid Computing **12**, 559–

592 (2014)

5. Khan, A.N., Mat Kiah, M., Khan, S.U., Madani, S.A.:

Towards secure mobile cloud computing: a survey. Futur.

Gener. Comput. Syst. **29**, 1278–1299 (2013)

[1] <https://cloudsecurityalliance.org/>

[2] <http://www.ciphercloud.com/blog/cloud-data-security-and-eu-data-privacy-rules-compliance-with-encryption-and-tokenization/>

[3] <http://www.gartner.com/technology/topics/cloud-computing.jsp>

[4] <https://www.skyhighnetworks.com/cloud-university/what-is-cloud-access-security-broker/>

[5] <http://www.gartner.com/technology/reprints.do?id=1-2RUEH70&ct=151110&st=sb>

[6] <http://www.ciphercloud.com/blog/cloud-data-encryption-easy/>

[7] <https://securityintelligence.com/how-to-protect-mobile-apps-essentials/>

[8] Abdul Nasir Khan, M. L. Mat Kiah, Mazhar Ali, Shahaboddin Shamshirband, Atta ur Rehman Khan. A Cloud-Manager-Based Re-Encryption Scheme for Mobile Users in Cloud Environment: a Hybrid Approach

[9] Abdul Nasir Khan, M. L. Mat Kiah, Mazhar Ali, Sajjad A. Madani, Atta ur Rehman Khan, Shahaboddin Shamshirband. BSS: block-based sharing scheme for secure data storage services in mobile cloud environment

[10] TODO: reference to SSS papers

[11] TODO reference to ABE papers

[12] Zhao, G., Rong, C., Li, J., Zhang, F., Tang, Y.: Trusted data sharing over untrusted cloud storage providers, presented at the IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom’10), Washington, DC, USA (2010).

[13] Yang, J., Wang, H., Wang, J., Tan, C., Yu, D.: Provable data possession of resource-constrained mobile devices in cloud computing. Journal of Networks **6**, 1033–1040 (2011).

[14] Itani, W., Kayssi, A., Chehab, A.: Energy-efficient incremental integrity for securing storage in mobile cloud computing, presented at the International Conference on Energy Aware Computing (ICEAC ’10) Cairo, Egypt (2010).

[15] Ren, W., Yu, L., Gao, R., Xiong, F.: Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing. Tsinghua Science & Technology **16**, 520–528 (2011).

[16] SAP Mobile Platform Secure Mobile with Mocana: webinar whitepaper