Designing a Secure Architecture for m-Health Applications

Muhammad J. Alibasa[1], Marlon R. Santos[1], Nick Glozier[1], Samuel B. Harvey[2], Rafael A. Calvo[1],
Senior member, IEEE

***Abstract*— As health apps become increasingly popular, software architectures that maintain user privacy and data security are essential. We discuss a software architecture for storing and managing data collected in a mobile health apps. Identifiable and non-identifiable data are stored in separate servers and encrypted. We discuss design considerations in real case situations.**

I. INTRODUCTION

Data stored in m-health, wellness and wellbeing systems [1], both in server and client applications, must be secured and protected from data breaches. The Verizon Data Breach Investigation Review showed that reported data breaches in the healthcare have increased nearly ten times from 2014 to 2015 [2], [3]. In a single event, U.S. health insurer Anthem disclosed 80 million unencrypted client medical records in a cyber-attack. Data included private information, from patient names, to income. Due to the longevity of medical records, for hackers these medical records often have even higher value than credit card information [4]. Mobile health apps (mHealth) have particular challenges discussed in the related work section. Medical data includes personal identifiable and non-identifiable information.

Personal identifiable data can be used to identify a particular individual. Several approaches are available to protect identifiable and non-identifiable data in m-Health systems. Encryption makes it harder for hackers to change or read data, unless the decryption key is compromised. Another approach requires a strong authentication process before providing authorization to access data [5], [6]. The authentication followed by authorization process will reduce the risk of intruders obtaining access to the data in the first place. Encryption, authentication, authorisation and other aspects of the design of software architecture must be taken into account to produce a system with the lowest risk to privacy and security. Identifiable information, e.g., full name or home address, is considered the most sensitive [7] because it can be used and is financially valuable to criminals (e.g. identify theft).

Non-identifiable data is less valuable by itself but can also be misused, particularly if it can be used to distinguish two persons making it possible to identify them. Loosing non-identifiable data can also be a breach of the end-user-agreements of commercial products and/or human research ethics approvals for research projects. The combination of identifiable and non-identifiable data is the most valuable so designs that separate these two are the most secure. They require intruders to break into two systems in order to combine both types of data.

The software architecture described here is designed to separate identifiable from non-identifiable data that can be kept anonymous. Data anonymization is considered one of the privacy requirements when storing m-health data [8], [9]. This architecture is designed for mobile and web applications.

II. RELATED WORKS

Secure frameworks and architectures have been proposed for m-Health. For example, Han and colleagues [6] proposed a framework for authentication and authorization, but no architecture design or implementation was presented. More recently [10] described an architecture that focused on wearable device communication. This work is extended by the mHealth security protocol (MHSP) proposed in [11]. The CONCERTO project [12] suggests a cross layer optimised architecture for mHealth service. The architecture is designed by utilising wide-range communication components, e.g., LTE and WiMax. These technologies are compared and analysed by using simulation on some m-health use cases. Still, the architecture lacks a security mechanism for both the communication and data management process.

Sulaiman et. al. proposed a security architecture for m-Health by thoroughly defining multilevel security for each communication type [7]. Data with more sensitivity has higher security with better and longer key length of encryption algorithm. A succeeding study [13] improved on this by providing a more comprehensive analysis on key length and encryption algorithm selection for data transmission. Another secure architecture, Next Generation e-Health (NGeH) proposed in [14] was generally an integrated system which consists of networks, profiling, and security mechanisms. However, all these proposed architectures are limited to securing communication process with no further explanation about securing their data in their database or storage systems.

Thiranant et. al. [5] discussed the importance of encrypting data in file transfer processes but without explicitly discussing data storage system and focusing only on upload and download file scenarios. Related to encrypted data transfer,

*R. A. C. is supported by an Australian Research Council Future Fellowship FT140100824. M. J. A. is supported by the Indonesian Endowment Fund for Education (LPDP). This study was developed in partnership with *beyondblue* with donations from the Movember Foundation.

[1]Muhammad Johan Alibasa, Marlon R. Santos, Nick Glozier, Rafael A. Calvo are with the University of Sydney, Sydney 2006 Australia (e-mail: johan.alibasa@sydney.edu.au, marlon.santos@sydney.edu.au, nick.glozier@sydney.edu.au, rafael.calvo@sydney.edu.au)

[2]Samuel B. Harvey is with the BlackDog Institute, Sydney Australia (e-mail: s.harvey@unsw.edu.au)

978-1-5386-1030-5/17/$31.00 ©2017 IEEE

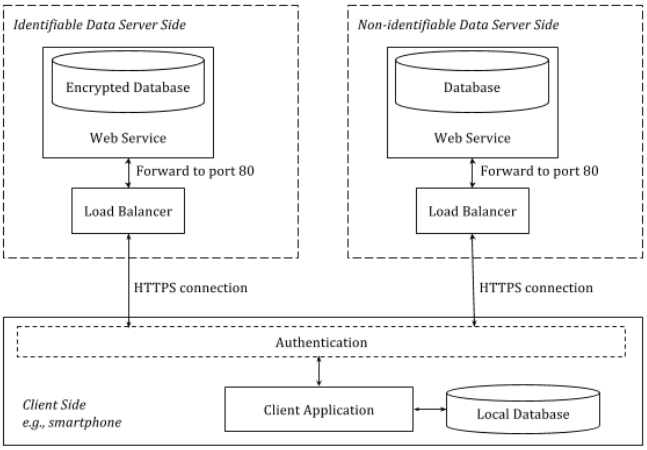

Fig. 1. Proposed architecture for e-health system

Luca et. al. [4] extensively illustrated the communication flow when users request any encrypted records. Their architecture has some comprehensive processes to manage encrypted data and decryption key transactions to client node. These kinds of security processes can be handled by multiple agents as discussed in [15]. However, the security measure principally aims to client side or node as it did not explicitly describe the security protection on the server side excluding data encryption.

The software literature does not generally describe a whole system, including server and client sides. Even a study on SaaS-Platform stops after utilizing encryption and secure communication process using HTTPS [16]. Our design and architecture contributes a secure server and client mobile app that stores identifiable and non-identifiable data separately. From this design, we then analyse security aspects and possible risk reduction in intrusion attacks.

III. DESIGN & ARCHITECTURE

Identifiable and non-identifiable data have different sensitivity so our architecture separates them into different physical servers. Each server has its own database and web service application. Because health data is only valuable if it can be matched to a person, this approach reduces the value of any data loss when a single server is compromised. For example, if attackers get access to the non-identifiable data they cannot learn anything about a particular person. If they get access to identifiable data only names and emails would be available.

A. Server Configuration

Two server categories handle requests from clients depending on the type of the requested data. An identifiable data server manages personal identifiable information, e.g., names, ages, genders, home addresses, emails, phone numbers, and other such sensitive information. This means that any requests related to register, login, and contact details will be handled by this identifiable data server. The other server type handles non-identifiable data, e.g., medical records and user self-reports. Since this server does not store contact

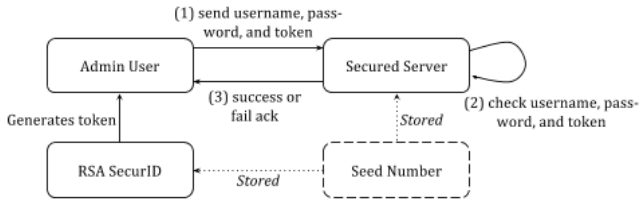

Fig. 2. Two-factor authentication process during admin login to server

information such as name, any records in this server must contain certain user Ids.

Figure 1 shows both servers and a load balancer to handle requests from clients. The load balancer only accepts HTTPS requests and only opens port 443. All these HTTPS requests will then be forwarded to the web service application via port 80. This forwarding approach means the client application never knows the real address of the web services. In addition, the web service is only available on port 80 and the system administrator can close all other ports. The servers do not talk to each other, thus the other server will be still secure even if one server is compromised.

Authentication requires two-factors [17] as shown in Fig. 2. The login process can be started by asking username and password, then followed by requesting RSA SecurID tokens. Optionally, the non-identifiable data server can also use this multifactor authentication.

For extra security, multiple web services can handle different identifiable data types. For example, one can store contact details with userId values and another can store the names of users for each userId. This approach will add another ”door” that is needed to be opened by attackers if they want to get information about specific users.

B. Data Storage Encryption

Identifiable data is encrypted and it will be secure if the decryption key is not compromised. Best practice is to not store the decryption key into a file located in the server. The key could be stored and shared using a cloud service. This way, intruders will not be able to find the decryption key in the server. The recommended encryption algorithm is the Advanced Encryption Standard (AES) [13] as it has a fast encryption and decryption speed.

Backup and restore procedures are also an important consideration of a security design. Encrypted backup are crucial to defend against the increasingly common ransomware or similar attacks [4] that lock the machine and makes it inaccessible.

C. Client Application

The mobile client application must authenticate before sending requests to either server. The authentication process can be initiated by sending secret key or application token to certify that this application is authorised. After this first authentication is complete, the client app will be able to send other requests to the servers API. In addition, there should be another authentication process if the requests are related to user or sensitive data. This authentication process can be

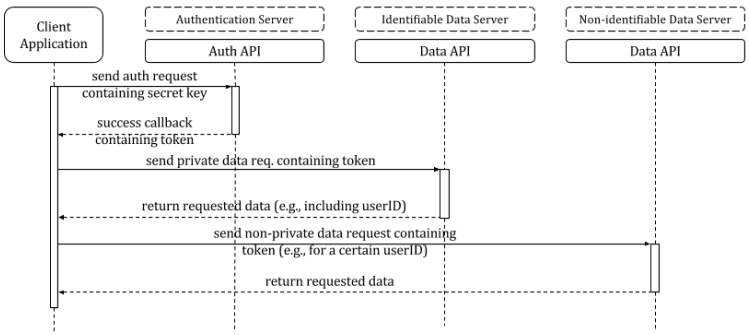

Fig. 3. An example of communication sequences

a login or an OAuth process which will send a password or a token to the corresponding server which manages the request. The server will then check whether the credential sent by the client is valid and authorised to receive the requested data. If the request is approved, then the server will send the requested data to the client. After receiving the data, the client can optionally store the data into its local database. However, it never stores any identifiable data to the local database. The local database can also be encrypted, but it is not always necessary since the data are not sensitive.

One single server can serve as authentication to enable single a sign-on (SSO) feature. This sign-in process will be simpler as mobile client applications only need to authenticate to this particular server before being able to request data from both data servers. The authentication process can use the OAuth protocol that will enable multiple client application integration where each application can share their data. This approach also enables scalable architecture where the client application can be added without considerable changes.

When client applications also collect non-identifiable data from third parties (e.g., Fitbit) the servers do not store access tokens, cookies, or any related login keys. The credentials to access the third-party data is only stored in the mobile client applications. This standard protects the users privacy since the service will not be able to collect any of this third-party data after a user uninstalls the app.

IV. SECURITY ANALYSIS

The most important design aspects for security are confidentiality, integrity, access control, availability, and non-repudiation. The architecture addresses each as follows:

A. Confidentiality

Servers only receive HTTPS requests which ensures confidentiality of the data in transfer [18]. All data, both identifiable and non-identifiable, is encrypted between server and client. The identifiable data storage should also be encrypted.

B. Integrity

Since the servers only accept HTTPS connections, data integrity can also be assured. HTTPS connections use TLS as its transport layer protocol. The TLS packets have a message authentication code (MAC) field which can check the message authenticity. MAC are generated from by using both messages and a shared secret key as input to a hash function. This method is secure since only people who know the shared secret key can change the MAC codes.

C. Access Control

Both servers in our design ask for secret keys or application tokens before the client is able to communicate with them. This mechanism ensures the access control or authorisation for each request from clients. Moreover, the servers will also ask for username and password if the client requests for identifiable data. Therefore, only authorised users are granted accesses to certain identifiable data depending on their roles. Admins or similar authorized people need to complete two-factor authentication to access or remote the server. This authentication also request RSA tokens, e.g., generated from authorised users smartphone via RSA SecurID application. This will ensure the security when non-authorised people have the username and the password of a certain authorised person.

D. Availability

The server in a secure architecture must limit their open ports to maintain its availability. For example, we only open port 443 for load balancers and port 80 for web services. Open ports can be exploited by attackers to launch an attack which causes the server to fail to provide its services. This attack could be an attempt to find vulnerable services which use a certain open port. By limiting the number of open ports, we can reduce the number of possible services which are vulnerable. Our system uses load balancers to receive requests from clients before forwarding the requests into the web service. These load balancers can serve as front doors thus other nodes cannot see the actual servers. In this case, attackers can only see the addresses of the load balancers and not the web service machine addresses. The availability of the service will be assured if the load balancer can handle certain attacks, e.g., DoS attacks.

E. Non-repudiation

Digital signatures can ensure non-repudiation of the transferred data. Every time clients send requests to a server, the clients will need to add their digital signatures to their requests. Therefore, the clients are not able to deny the requests

made to the servers. However, digital signatures are based on asymmetric cryptography which use higher computational power. This computational addition may significantly lower the traffic speed. The digital signature implementation also needs additional infrastructures to manage public and private keys. Optionally, tokens generated by the servers can serve as digital signatures to verify the identity of the users as they are stored privately.

V. DISCUSSION AND LIMITATION

The software architecture described here is being used to enhance security in our mobile eHealth applications. We believe that this architecture is also suitable for web applications and wearable device systems which communicate through smartphones.

Possible limitations include that the server side in our proposed design uses a web service application that handles all requests. Therefore, the design may not be applicable for other architectures, e.g., publish-subscribe architectures.

The web service application in this design should be in executable file form. This means that the source code of the web application should not be located or inside the running machine. This approach will reduce the possibility that attackers can change the application or find the vulnerabilities. For example, if the web application is based from Java Spring framework, then it is better to only put the jar file of the application in the server and run the application from it.

Our design can follow SafeProtect model and protocol for data sharing in eHealth system [19]. This model and protocol ensures the privacy and policy of shared data. The data sharing process must comply to certain XACML policy which is defined by the data owner. In a case where the data owners can share their data to doctors or specialist, our design can follow a security protocol proposed by Thilakanathan et. al. [20]. The protocol illustrates on how encryption key partitioning algorithm can assure the security of data sharing process between patient and doctor via social networks.

Our future work will include a performance analysis and a comparison with a single server design. We believe that the performance of our design is better and more secure than standard single service models since requests from clients will be handled by different servers depending on data types. However, our design has higher implementation and maintenance cost as it needs more than one server.

VI. CONCLUSIONS

In this paper, we describe a secure and privacy protecting architecture for mobile e-health applications. Extensions to earlier work include the addition of a load balancer node and the separation between identifiable and non-identifiable data server. The added load balancer only accepts HTTPS connections that maximise the confidentiality and the integrity of the data communication. Meanwhile, the separation into two categories maximises the confidentiality of stored data and adds a security layer to protect the data. This architecture is particularly suitable for e-health mobile application and other e-health systems which are based on web service applications.

REFERENCES

[1] R. A. Calvo and D. Peters, *Positive computing: Technology for wellbeing and human potential*. MIT Press, 2014.

[2] Verizon, "Verizon 2014 data breach investigations report."

[3] ——, "2015 data breach investigations report."

[4] G. D. Luca, M. Brattstrom, and P. Morreale, "Designing a secure e-health network system," in *2016 Annual IEEE Systems Conference (SysCon)*, April 2016, pp. 1–5.

[5] N. Thiranant, M. Sain, and H. J. Lee, "A design of security framework for data privacy in e-health system using web service," in *16th International Conference on Advanced Communication Technology*, Feb 2014, pp. 40–43.

[6] S. Han, G. Skinner, V. Potdar, and E. Chang, "A framework of authentication and authorization for e-health services," in *Proceedings of the 3rd ACM Workshop on Secure Web Services*, ser. SWS '06. New York, NY, USA: ACM, 2006, pp. 105–106.

[7] R. Sulaiman, D. Sharma, W. Ma, and D. Tran, "A security architecture for e-health services," in *2008 10th International Conference on Advanced Communication Technology*, vol. 2, Feb 2008, pp. 999–1004.

[8] D. Kotz, S. Avancha, and A. Baxi, "A privacy framework for mobile health and home-care systems," in *Proceedings of the First ACM Workshop on Security and Privacy in Medical and Home-care Systems*, ser. SPIMACS '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[9] S. Avancha, A. Baxi, and D. Kotz, "Privacy in mobile technology for personal healthcare," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 1–54, Dec. 2012.

[10] A. Molina-Markham, R. Peterson, J. Skinner, T. Yun, B. Golla, K. Freeman, T. Peters, J. Sorber, R. Halter, and D. Kotz, "Amulet: A secure architecture for mhealth applications for low-power wearable devices," in *Proceedings of the 1st Workshop on Mobile Medical Applications*, ser. MMA '14. New York, NY, USA: ACM, 2014, pp. 16–21.

[11] F. Goncalves, J. Macedo, M. J. Nicolau, and A. Santos, "Security architecture for mobile e-health applications in medication control," in *2013 21st International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013)*, Sept 2013, pp. 1–8.

[12] L. Iacobelli, G. Panza, E. Piri, J. Vehkaper, M. Mazzotti, S. Moretti, S. Cical, L. Bokor, N. Varga, and M. G. Martini, "An architecture for m-health services: The concerto project solution," in *2015 European Conference on Networks and Communications (EuCNC)*, June 2015, pp. 118–122.

[13] A. Boonyarattaphan, Y. Bai, and S. Chung, "A security framework for e-health service authentication and e-health data transmission," in *2009 9th International Symposium on Communications and Information Technology*, Sept 2009, pp. 1213–1218.

[14] M. A. Fengou, G. Mantas, D. Lymberopoulos, N. Komninos, S. Fengos, and N. Lazarou, "A new framework architecture for next generation e-health services," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 1, pp. 9–18, Jan 2013.

[15] R. Sulaiman and D. Sharma, "Enhancing security in e-health services using agent," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, July 2011, pp. 1–6.

[16] R. D. Berndt, M. C. Takenga, S. Kuehn, P. Preik, G. Sommer, and S. Berndt, "Saas-platform for mobile health applications," in *International Multi-Conference on Systems, Sygnals Devices*, March 2012, pp. 1–4.

[17] R. L. de Souza, L. C. Lung, and R. F. Custdio, "Multi-factor authentication in key management systems," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, July 2013, pp. 746–752.

[18] S. Mukherjee, K. Dolui, and S. K. Datta, "Patient health management system using e-health monitoring architecture," in *2014 IEEE International Advance Computing Conference (IACC)*, Feb 2014, pp. 400–405.

[19] D. Thilakanathan, S. Chen, S. Nepal, and R. Calvo, "Safeprotect: Controlled data sharing with user-defined policies in cloud-based collaborative environment," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 301–315, April 2016.

[20] D. Thilakanathan, A. R. Calvo, S. Chen, S. Nepal, and N. Glozier, "Facilitating secure sharing of personal health data in the cloud," *JMIR Med Inform*, vol. 4, no. 2, p. e15, May 2016.