Multi-Authority Attribute Based Encryption for Public Health Records

Lance Aaron Go See – CSCI 531 – Spring 2018

Computer Science M.S. – University of Southern California

Introduction:

Online Public health records (PHRs) were once just an imaginary concept; a dream of improving the antiquated system of paper records. However, with the timely rise of the internet and the accessibility of smart phone devices increasing dramatically in the past decade, this vision has come to fruition. In general, online PHRs are more beneficial than paper records, as paper records can be stolen, hard to transfer and insecure compared to their online counterparts. For instance, there is no record of the number of hard copies that have been made of a person's health record, and no tracking number to see where these copies have gone. One might ask why a person's height or prescriptions should be private, but it is sensitive data like HIV diagnoses or embarrassing medical test that require secure PHRs.

In typical PHR executions, a user uploads his PHR to a cloud service and lets a health service such as Blue Cross take care of distribution and security of his PHR. However, this leads to a few issues. First, a user no longer has control over his PHR, giving full control to Blue Cross. This leads to a key escrow problem – if Blue Cross servers were attacked, there would be a catastrophic event where everyone who trusted Blue Cross's keys would be compromised, leading to leaked health records. Also, if an employee at the health service were to turn rouge, he or she could vengefully leak a person's public health info.

These issues lead to this proposed implementation and framework of a multi-authority attribute based encryption (ABE) protocol. In this proposed system, a modified ABE is implemented to create a more secure PHR system. By utilizing an encrypt-then-upload idea, this protocol solves some of the security issues introduced when uploading and distributing PHRs and steps away from relying on a health service to implement security, placing the power in a user's hands.

Background Information – IBE:

In order for one to better understand the proposed system, it is essential to understand the underlying components that it relies on. Attribute based encryption was built as an extension of the idea of identity-based encryption or IBE. IBE was originally a concept by Adi Shamir [1]. In his paper, he described the concept of a system that would allow for users to have secure communications without needing to exchange public/private key pairs. In public key encryption or PKI, a user would have to create and upload his public key to a server before anyone can send the user an encrypted message. This causes issues since Bob would need to wait for Alice to do this before he can send her a message. In addition, PKI protocols rely on a key center that must always be active. If it fails, Bob has no way of communicating with Alice. In addition, another big issue with PKIs are that public keys must also involve certificates. It is a complex procedure that adds another point of failure, and leaves trust in the hands of a certification authority to guarantee public key authenticity.

This is what lead Shamir to suggest a system in which publicly known attributes can be used as the public key. For instance, a person's email address or phone number could be used as a public key. By making the key a publicly known attribute, Bob no longer needs to wait for Alice to upload her private key, nor does he need to verify certificates with a CA, as the keys are publicly known attributes associated with the person. However, when Shamir wrote his paper, he did not include a usable implementation for IBE, but included a proposal for an IB signature scheme.

Successful IBE schemes generally fall into two categories, pairings based and quadratic residue based. Cocks [2] was the first one to propose a IBE implementation, using quadratic resides. However, Boneh and Franklin [3] were the first ones to propose a widespread usable IBE implementation utilizing bilinear graph pairings. Boneh and Franklin outlined a 4 step algorithm that consisted of setup, extraction, encryption, and decryption. Bilinear graph pairings crypto is a growing field based on creating a pairing such that $e(g^a,g^b)=e(g,g)^{ab}$ and utilizing El Gamal's encryption theory. Intriguingly, pairings have previously been used as an attempt to break elliptic curve cryptography, but was used by Boneh and Franklin to implemented an IBE scheme. Essentially, the scheme takes the public identity concept suggested by Shamir[1] and uses pairing set properties on bilinear maps to allow users to generate private keys from those public attributes. This whole process is done by contacting a TTP (trusted third party).

The benefits of this IBE scheme is that it was the first time that IBE was able to be widely used. Through this scheme, Bob can send messages to Alice ahead of time using her public identity, and Alice can simply request a key from the TTP later and still be able to decrypt the message. However, we run into the issue of key escrow. There is a big issue if the TTP turns malicious or is compromised; the TTP has the ability to generate keys for any user so this would be catastrophic. The proposed PHR system solves this using a multi attribute approach. Throughout the years, different authors improved upon Boneh and Franklin's IBE scheme, including Sakai and Kasahara[4] who improved its performance and security, extending the scheme to be secure against random oracle attacks.

ABE:

A novel idea was created when Sahai and Waters[5] first introduced the idea of "Fuzzy Based IBE". In their paper, they wished to overcome the issues of implementing biometric security for IBE; that most biometric implementations rely on error tolerance to work. Their proposed IBE scheme solved this issue by allowing for error tolerances. However, they also stated another novel idea, attribute-based encryption or ABE. The error tolerances that would allow biometric IBE to function also allowed for ABE. Essentially, ABE uses attributes to determine a specific access control protocol, and provided security against collusion attacks, where multiple parties with different attributes collaborate to gain access to files they don't individually have access to.

The basic construction suggested by Sahai and Waters has each attribute of a user have a private key component linked to a random polynomial $q(x)$[5]. If a user can match at least a certain amount of components of ciphertext with a private key, they can decrypt the file. This was an extension of the pairings based IBE suggested by Boneh and Franklin, and is the concept used in this PHR construction.

ABE methods generally split into two types, Key Policy Attribute (KP-ABE) and Ciphertext Policy Attribute (CP-ABE). In KP-ABE, the owner of the data will create the master keys. Using these master keys, the data is encrypted in such a way that it is labeled with the user's access control policy. Private keys are generated with this access structure and given to the appropriate users, who in turn can access parts of the document that they have been allowed. However, the main issue with this implementation is that the encryptor does not have granular control over the access structure as keys are issues by a TTP.

In CP-ABE, a user encrypts a message that will specify an access structure over attributes. A key generator system will generate private keys for any attributes that it can certify a user has. If the key holder fits the attributes specified by the ciphertext, a user will be able to decode the message. However, if the key does not meet the attributes specified in the ciphertext, the user will not be able to decrypt the message. The main difference between CP-ABE and KP-ABE is that a user has more granular control under CP-ABE as they can decide which attributes are allowed access, and can encrypt this directly into the ciphertext. The key generator system does not need to contact a user ahead of time for his policy, as the key generator system only has to certify attributes to users (the encryptor will specify policies themselves) versus KP-ABE where the key generator must be trusted to grant or deny access to the proper users to fit the encryptor's policies. However, even with all of these developments, IBE/ABE systems are still not fit for PHR applications. There are two main issues, the key escrow issue, and revocation issues.

ABE/IBE Issues:

The key escrow problem still exists in these two ABE schemes. Because there is a central authority in both ABE implementations, there is a big issue with key escrow. Although the certification authority or key generator system is assumed to be trustworthy, attacks on these systems would lead to many people's keys being leaked. In addition, what if a user wishes to revoke access to his files, after he has granted a person access? For instance, if a patient was in a hospital at one point, but moved to a new city, how would he move his PHR access from one city to another? CP-ABE itself only has a small amount of implementations that work with attribute withdrawals, and those solutions often still have issues with key escrow.

Proposed Solution:

Multi-Authority ABE or MA-ABE was first proposed by Chase[6] and solves the issue of key escrow. In addition, a modification on this system was proposed by Li [7] which is used in this implementation of PHRs. MA-ABE with revocation works along the lines of the system proposed by Chase, where multiple authorities govern a disjoint set of attributes. Because each authority does not have full control over all attribute certifications, the key escrow issue is prevented. However, this MA-ABE proposed by Chase is a KP-ABE. A CP-ABE is wanted, as this will allow for user-based revocation. Using agreement of key policies and rules specified by required attributes in the ciphertext, Li[7] was able to show that MA-ABE can be shown similar to CP-ABE. This exploits the same threshold policies suggested by Sahai and Waters for fuzzy IBE[5].

MA-ABE revocation is facilitated by the MA-ABE authorities. An authority can revoke a user's attributes by re-encrypting the ciphertext and updating the unaffected key holder's secret keys. In addition, by having version numbers on the attributes and ciphertext, an attribute authority can issue a version update in addition to the earlier steps to revoke access to a user. This system allows for revocation, as if a user that has been revoked attempts to access the file, his attributes (and version number) will no longer match or verify with the PHR.

System Overview:

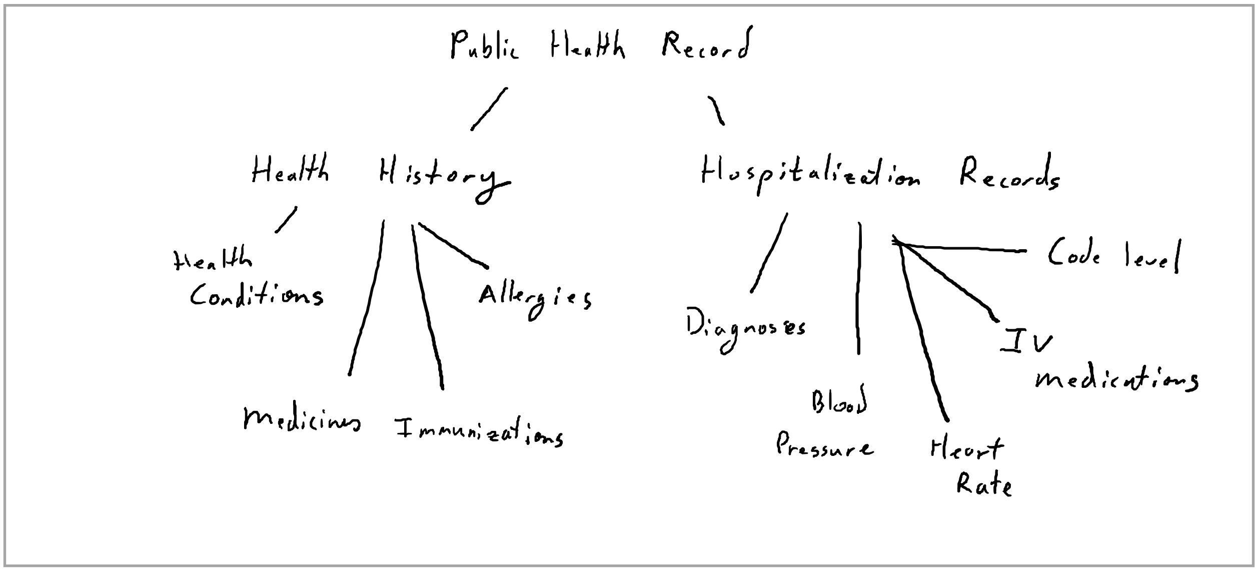
This PHR system uses MA-ABE with revocation to realize a secure online PHR. The framework divides access into two main domains using both KP-ABE and MA-ABE, as suggested by Li[7]. The public domain includes anyone in the medical professional roles such as doctors, nurses, and pharmacist while the personal domain includes friends and family that may also wish to access the PHR.

MA-ABE is used for the public domain to allow for multiple attribute authorities to certify users. For instance, the AMA may certify that a user has a MD while the ABMS may specify a specialty that the user has. By using this, roles can be specified, and doctors/medical professionals don’t need to directly contact the user to get attribute keys. Also, a PHR owner doesn’t need a list of professionals and attributes before they can encrypt their desired structure. They can specify what structure they wish in advance, and those professionals who’s keys fit the access structure will be allowed to decrypt the PHR.

KP-ABE is used for the personal domain since the PHR owner knows the family and friends he wishes to give access to personally. This allows for a greater level of trust for key distribution, and allows a PHR owner to grand access to data on a case by case basis, as data attributes are defined into each PHR file. This multidomain system allows for better security, as attributes are verified by the best party in each case.

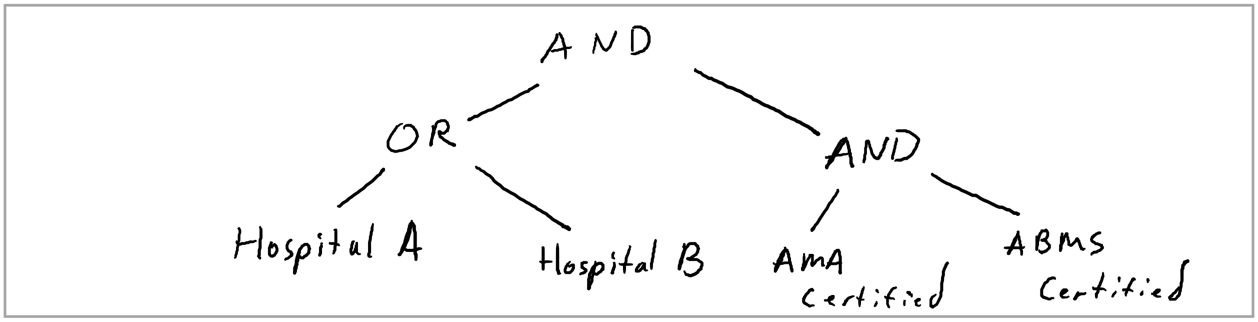
Framework:

During the beginning setup process, for the personal domain, the system will first define common attributes that the personal domain users all have access to. Then, the PHR owner’s client will generate public and secret keys, which will be distributed to the allowed users. A user will only be allowed access to the files according to a predetermined access structure (See figure 1). In the public domain, MA-ABE is used. A doctor or pharmacist will go to an attribute authority to obtain keys which bind their attributes to them, visiting multiple authorities to get all the attributes certified.



(This file access diagram shows how files can be organized to allow or block users from viewing certain files in the personal domain. For instance, a user with access to Medicines would not be able to view Hospital Records but someone with access to Health History would be able to view Immunizations as well as Medicines)

In the encryption stage, a PHR owner will use ABE to encrypt their PHR using the CP-ABE method. This allows the owner to encrypt files ahead of time, without needing to get a list of attributes from the AMA or ABMS.



(Possible Key Policy for Public Domain- In this case, if someone from Hospital C attempts to decrypt the PHR, he would not be able to due to not fitting the access structure)

For the personal domain, a PHR owner must first collect user attributes before encoding files with KP-Abe. A user in this domain will only have access to parts of the file that his key has access to, as defined by the PHR owner.

As stated before, revocation is a big issue for PHRs. To accomplish this, the previously stated MA-ABE with revocation should be used. With version numbers and rekeying, the PHR owner can revoke access easily, even with the MA-ABE in use.

Another big issue for PHRs is emergencies. When an emergency occurs, it may become important for medical records to become accessible, regardless of a user's access structure. For instance, what if a patient is unconscious and can't update policies to allow access? Thus, a PHR must also allow an emergency access function. To accomplish this, a special emergency key will be derived and sent to a special authority called an ED[7]. Upon an emergency, any hospital can contact the ED, who will then verify the caller, hospital and condition of the patient, before releasing the emergency key to the hospital. After the emergency, a user can revoke access by using the revocation method stated earlier, and reissuing a new emergency key to the ED.

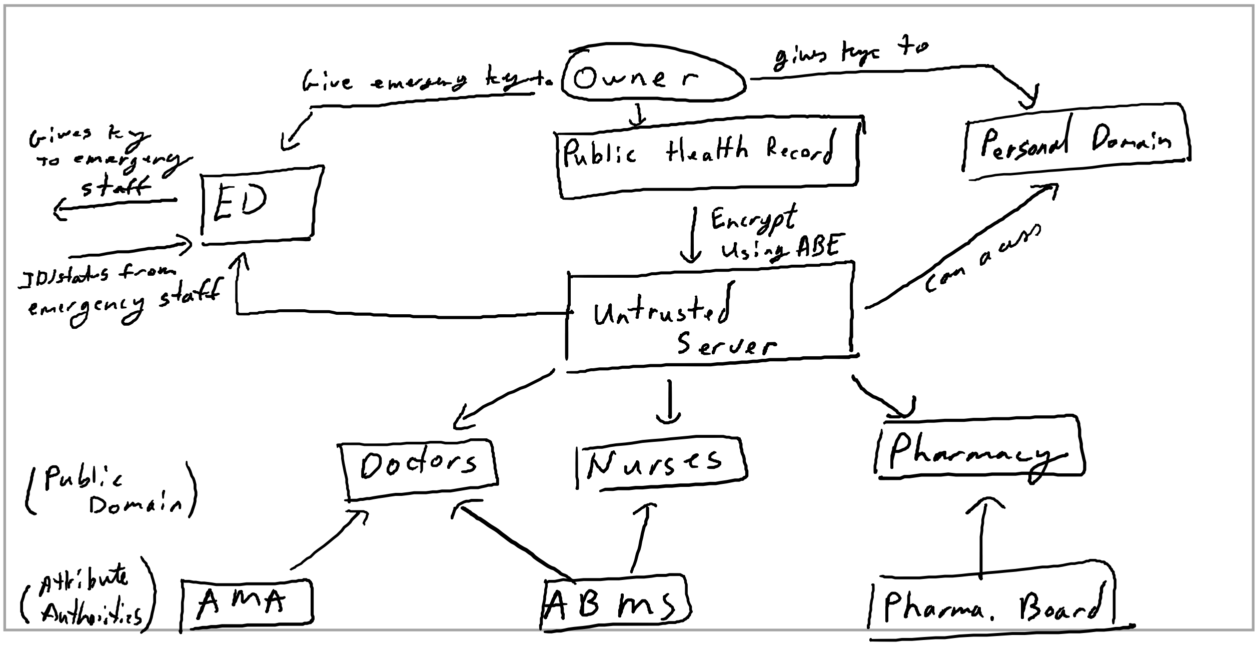


Diagram of PHR System

Example Scenario:

Imagine that Alice created a PHR following a structure that allows for doctors certified by the AMA and part of Hospital A to access her PHR. If someone from hospital C attempts to access Alice's PHR, he will not be able to as his attributes will not match the access structure. If a doctor from hospital A with certification from the AMA attempts to access her file, he will be able to.

On the other hand, imagine that Alice has a stroke while being on holiday to Hawaii. The emergency staff will quickly discover and contact Alice's ED and call them. The staff member gives detailed proof of their identity as well as Alice's condition, and will get the emergency key. The emergency staff can then access the files and save Alice's life. When Alice is released, she will be able to thank the emergency staff, revoke the emergency key and supply a new one to the ED.

Rollout Out Ideas and Issues:

Ideally, this ABE structure provides a backbone for implementations using smartphones and computers. The setup and uploading of PHRs can be done via a phone app. This app will communicate with the servers which will hold the PHR. By encrypting the files onboard the phone/computer, attacks on the communication protocols will not compromise the PHR. However, if Eve is able to block communications between the PHR holder and the server and upload a false PHR, she may be able to cause serious injury or even death. This protocol does not protect against this. Using a "staff" app, health workers could contact attribute authorities to get attribute keys, assuming that attribute authorities are trustworthy. An idea to allow old app users to verify their identity would be to use biometrics. If a user can verify his identity using his fingerprint, an attribute authority can better trust that the attributes assigned match the requesting person. New students graduating out of medical school can be assigned a secret number which the app can use authenticate their identity/credentials. The app then can act as an access point to a user's PHR. By having decryption done on the device, we can reduce the issues with insecure transmission of PHRs. However, a function must be built to prevent screenshots and to overwrite the temporary storage location of the PHR on the device after access. In addition, the ED's have a big burden on their hands. Because the ED has the most power and is the weakest link, most attacks on this system will focus on the ED. Since the ED can bypass the access structure, a method must be developed to secure the ED. Finally, what if a trojan or worm attacked a doctor's device? In this case, an adversary could steal keys and use it pretending to be the doctor. A possible solution would be to use biometric verification immediately before the PHR is accessed, however there may be better implementations.

Concluding Thoughts:

In closing, this MA-ABE system allows for PHRs to be more secure than before. Through utilizing multiple authorities, the key escrow problem in ABE is solved and using Li's revocation modification allows for users to revoke access to their PHR. However, there are still many problems to be solved before PHRs are secure. In addition to the roll out issues discusses previously, what happens when a user passes away? Will his PHR be deleted or saved? What happens if an error in the PHR system causes someone to die, while a paper system wouldn't have? These are all issues that should be explored in future research.

References:

[1] Adi Shamir. Identity-based cryptosystems and signature schemes. In Proceedings of CRYPTO 84 on Advances in cryptology, pages 47-53. Springer-Verlag New York, Inc.,1985

[2] Cocks, C.: An identity based encryption scheme based on quadratic residues. In: IMA International Conference on Cryptography and Coding. pp. 360, 363. Springer (2001).

[3] Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Annual International Cryptology Conference. pp. 213, 229. Springer (2001).

[4] Sakai, R., Kasahara, M.: ID based cryptosystems with pairing on elliptic curve. IACR Cryptology ePrint Archive 2003, 54 (2003)

[5] Sahai, A., Waters, B. 2004. Fuzzy Identity-Based Encryption. Cryptology ePrint Archive. 2004/086.

[6] Chase, M., Chow, S.S.M.: Improving privacy and security in multi- authority attribute-based encryption. In: Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09) pp. 121–130 (2009)

[7] Li, Jiguo Qian, Huiling, et al. "Privacy-Preserving Personal Health Record Using Multi-Authority Attribute-Based Encryption with Revocation." International Journal of Information Security, vol. 14, no. 6, 2014, pp.487-497