

# Securing Medical Records on Smart Phones

Ryan W. Gardner  
Johns Hopkins University  
Baltimore, Maryland, USA  
ryan@cs.jhu.edu

Sujata Garera  
Johns Hopkins University  
Baltimore, Maryland, USA  
sgarera@cs.jhu.edu

Matthew W. Pagano  
Johns Hopkins University  
Baltimore, Maryland, USA  
mpagano@cs.jhu.edu

Matthew Green  
Johns Hopkins University  
Baltimore, Maryland, USA  
mgreen@cs.jhu.edu

Aviel D. Rubin  
Johns Hopkins University  
Baltimore, Maryland, USA  
rubin@cs.jhu.edu

## ABSTRACT

There is an inherent conflict between the desire to maintain privacy of one's medical records and the need to make those records available during an emergency. To satisfy both objectives, we introduce a flexible architecture for the secure storage of medical records on smart phones. In our system, a person can view her records at any time, and emergency medical personnel can view the records as long as the person is present (even if she is unconscious). Our solution allows for efficient revocation of access rights and is robust against adversaries who can access the phone's storage offline.

## Categories and Subject Descriptors

E.3 [Data]: Data Encryption; K.6.5 [Management of Computing and Information Systems]: Security and Protection—*authentication, unauthorized access*

## General Terms

Security

## 1. INTRODUCTION

Physicians and emergency medical technicians (EMTs) routinely encounter unconscious patients who possess little or no information documenting their medical status. This poses a significant hazard when a patient has a serious condition such as epilepsy or diabetes. The additional time required to diagnose a patient can lead to severe injury or even loss of life.

Unfortunately, negative consequences can ensue if someone's personal medical information is revealed. For example, some employers fear that they incur cost or risk by hiring people with chronic conditions. In 1998, diabetic pharmacist Stephen Orr was fired by Wal-Mart because he required a 30-minute lunch break every day as an important part of regulating his glucose and insulin levels [42]. In 2004, Cirque

de Soleil officials acknowledged that they fired Matthew Cusick because he was HIV positive, resulting in a \$600,000 settlement [37].

Despite the possible negative consequences of disclosing health information, the availability of this information can be vital during times of medical need. Hypoglycemia (low blood sugar), for example, is a potentially dangerous condition that affects diabetics at a much higher rate than most people. Severe cases can result in reduced brain activity, unconsciousness, coma, and even death if not addressed. However, knowledge that a patient has diabetes may expedite diagnosis of hypoglycemia, which can lead to simple treatment. Furthermore, 100,000 people die every year due to adverse drug reactions [29]. With knowledge of a patient's medical information, emergency room physicians may be able to administer more effective medications with reduced likelihood of inducing dangerous drug interactions. As such, the availability of individual medical records can have an significant impact on a person's health.

In this paper, we focus on the problem of granting emergency access to patient medical information while preserving patients' privacy in all other circumstances. One approach to this problem is to deploy a central, trusted database that stores each person's personal health record (PHR). When a medical employee requests someone's PHR, the database verifies her credentials using standard access control techniques, logs the request, and sends her the record if she is authorized. Such a solution has several shortcomings. First, the solution relies on the constant availability of a network connection to the central database. Such a requirement may not always be reasonable for highly mobile EMTs. Furthermore, all EMTs and physicians receive quick access to potentially millions of records regardless of each record owner's situation or preferences. This is especially worrisome in the event that a medical employee loses her credentials. Lastly, patient identification may be infeasible in many cases.

To avoid these limitations, we propose an alternative approach whereby every person can carry all of her vital medical records on a small, personal electronic device, such as a cellular phone. We introduce new cryptographic protocols to enable record access under specific, acceptable circumstances only. In the most common case, the owner of a medical record can access her record at any time. This is desirable because she may wish to review aspects of it for correctness or check doctor recommendations while away from home. Through her own access, the person can also easily

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SPIMACS'09, November 13, 2009, Chicago, Illinois, USA.

Copyright 2009 ACM 978-1-60558-790-5/09/11 ...\$5.00.

disclose her records to medical personnel. Moreover, we designed our solution so that emergency personnel can access a patient’s medical record when she is present, even if unconscious, as long as some biometric information is available from her (such as her face or a finger). At the same time, someone who steals her phone, even a medical professional, cannot learn anything useful about her record without compromising her password or biometric. This is true even if the thief has full access to the phone’s hardware.

Our technique is primarily based on secret sharing and allows for simple revocation of compromised or retired credentials. Overall, the solution demands little of the users and requires few resources of their mobile electronic devices. We believe it is quite practical with the existence of one or more reliable biometrics. It is also flexible and could be implemented without biometrics.

## 2. RELATED WORK

The medical and technological communities have explored varied techniques for health record storage and controlled access in emergency situations. In this section, we briefly discuss previous work in this area.

AllOne Mobile Health [1] represents one such product. This solution provides consumers with access to their health records through a mobile phone. It allows users to control access to their PHR, receive health information updates, and access history during an emergency. Most details of the product are proprietary and unavailable to the public, and unlike the solution we present, the security of the medical records relies exclusively on a user-entered personal identification number (PIN). Similar to AllOne Mobile, icePHR Mobile [13] (In Case of Emergency PHR Mobile) also allows users to manage their health information on their cellular phones. In addition, icePHR Mobile allows the user to edit her personal medical record on the icePHR server through the mobile client application. The users’ medical records can be downloaded onto the phone after the user has authenticated herself to the server. However, once information is stored locally, it may be viewed by any person who has access to the phone.

USB flash drives are another means of storing one’s PHR. Personal HealthKey [14], for example, allows a patient to store her health record encrypted on a USB device. Wright *et al.* [44] evaluate the security of these USB devices and demonstrate that the encryption is used incorrectly. In particular, the authors show that the encryption key on the analyzed devices [14, 28] is constant and not dependent on the password chosen by the user. In previous research Wright *et al.* [45] also show that these devices present a security threat to hospital computers. Furthermore, as noted by Tang *et al.* and Olla and Tan, such stand-alone devices that rely solely on user input often fail to maintain an up-to-date health record [41, 31].

Hinkamp [23] introduces a new RFID-based system that provides medical personnel with necessary data during an emergency. The device may be attached via a wrist bracelet and can supply critical information such as blood type, allergies, and emergency contact information.

While the system by Hinkamp [23] provides vital medical information to health care providers, it is not equipped for real-time processing of health status. WAITER [46] and AMON [26] are notable examples that do not have this limitation. WAITER, introduced by Wu *et al.*, continuously

monitors personal body status such as heart beat and temperature, and issues alerts in case a threatening or abnormal situation arises. Furthermore, WAITER uses a mobile phone to process the vital signals captured in real time and periodically sends health-status reports to a health care center. AMON, an invention of Lukowicz *et al.*, is targeted specifically towards cardiac and respiratory patients and monitors several parameters such as oxygen level, level of physical activity, and one-lead ECG. Similar to WAITER, it utilizes the cellular network to transmit information to the medical center. Preuveneers *et al.* [35] present a device similar to AMON and WAITER for diabetic patients. In addition to presenting a full prototype implementation, the authors conduct a usability study demonstrating the effectiveness of the device.

Dillema and Lupetti [18] present a patient record access control technique that provides access only if the patient and health care provider physically meet. Under their scheme, the health worker carries the complete encrypted database of patient records while the patient carries the necessary capabilities to allow a health worker access to her medical record. Although this access control method provides a useful alternative to centralized systems and to patients carrying their own records, it does not allow patients to view the contents of their records. Furthermore, it structures stored records such that large portions of them are stored redundantly requiring significantly more space than occupied by the records themselves.

The notion of “break-glass” emergency access systems has also been explored in previous work [34, 24, 19]. Povey introduces the concept of optimistic access control as a means of allowing users to exceed their normal privileges in an emergency [34]. He also discusses applications of optimistic security for medical emergencies, mission-critical system crashes, and sand-boxing semi-trusted applications. Moreover, Ferreira *et al.* [19] implement an emergency break-the-glass access policy for the virtual electronic medical record system of the University of Porto. Compared to the model introduced by Povey, their policies are more restrictive.

Patient permission is an important factor to consider when providing access to medical records. Nepal *et al.* [30] present a framework that allows a patient to express her consent with respect to access of her health record. In particular, the patient can decide which health care facilities and providers may access her record. While our solution currently falls short of providing such granular control, we believe it can be incorporated into our framework.

## 3. ADVERSARIAL MODEL

The focus of our paper is to store personal health records (PHRs) on users’ personal devices, such as smart phones. We do not address protection or storage of users’ data in central databases or repositories, and we assume that all transfers occur over a fully authenticated, confidential connection such as SSL.

We consider two different types of adversaries. The first type of adversary, which we refer to as an *online adversary*, interfaces with the phone through its normal operating system and applications. The typical prototype for such an adversary is a person who takes the phone belonging to someone nearby, attempts to view private data on it, and covertly returns it before its owner notices. She might be an EMT trying to quickly see a record that she should not view

or a misbehaving physician who wants to learn the medical history of an acquaintance or a celebrity.

The second type of adversary, which we refer to as an *offline adversary*, is the primary focus of this paper. This adversary has complete read access to all components of a user’s phone. It is someone who may have stolen a phone or has the ability to access it while the owner is away for a significant period of time. She can read the phone’s raw memory and storage regardless of its software or electronic access control.

We do not specifically consider adversaries who can write to a phone’s storage and then cause the user to use the phone although our solution could be adopted to protect against such adversaries with hardware support for checking memory integrity. Intuitively, an adversary who can alter a phone’s storage can, for example, add malicious software to simply send the user’s PHR to the adversary whenever someone accesses (decrypts) the PHR legitimately. Similarly, we assume adversaries do not compromise data or software on the phone remotely. Hopefully, someday, methods from trusted computing [38, 6] could be applied to phones to verify the integrity of their data and prevent such attacks.

## 4. OUR APPROACH

We now present our general approach to confidentially storing medical records on smart phones. We discuss the finer details and design decisions of our architecture in the section that follows.

### 4.1 Tamper Resistance vs. Cryptography

Recall that our goal is to make personal medical information available under specific circumstances, such as the presence of an authorized user while preventing access to it in all other instances.

One possible solution to this problem utilizes trusted hardware. At a high level, secure hardware on each phone guards the owner’s medical record, and anyone wanting to access it (transparently) provides the hardware with her credentials or proof that appropriate circumstances exist to enable access. The hardware then allows access if and only if the credentials or proof are valid.<sup>1</sup>

A benefit of the trusted hardware approach is the fact that it can prevent offline attacks to guess or obtain credentials. For example, trusted hardware can erase private keys needed to access the record after some number of bad authentication attempts. Hardware may also allow logic for which no cryptographic method is known. However, the primary downside is that new phones would require additional, likely expensive hardware, and the solution could not be deployed on existing phones. Furthermore, trusted hardware, such

<sup>1</sup>Specifically, the approach can be summarized as follows: Each user stores a private key on some tamper-resistant memory located within a piece of trusted hardware in her phone. She uses the corresponding public key to encrypt a symmetric key that is used to encrypt her health record, and both the encrypted key and encrypted record are stored on the phone’s primary storage. Whenever anyone wants to access the record, she can provide her credentials to the trusted hardware or otherwise give it proof that appropriate circumstances exist to allow record access. If the hardware verifies that this is indeed the case, it decrypts the symmetric key used to encrypt the record, which then allows the record to be decrypted itself and viewed. Importantly, the hardware never reveals the private key itself.

as a trusted platform module (TPM) is generally limited in functional complexity due to its stringent physical security requirements, and it is of course much harder to change than software. Thus, intricate logic or requirements for access control and decrypting records might not be possible or may become very expensive, and implemented solutions would be semi-permanent. Because a working method that can be implemented purely in software is applicable to current devices with appropriate hardware for reading biometrics, significantly more cost-effective, and easier to modify after deployment, we introduce a cryptography-based solution.

### 4.2 Biometrics

Password-based authentication is a popular technique for controlling access to resources. However, during a critical medical situation, a record owner may be unconscious and thus unable to provide her password. For this reason, a more persistent credential that can be associated with users’ identities is useful for regulating access to health records. Because physical attributes stay with people at all times, we consider the use of biometrics in our framework. We suggest that non-behavioral features such as fingerprints and face geometry can provide a practical authentication credential because they are readily available and require no action from the user. Both have also been explored for authentication using current cellular phone hardware [36, 2].

As we describe in the following sections, our framework utilizes biometric measurements to generate cryptographic keys for encryption. Previous research has introduced methods of biometric key generation from various modalities [47, 17, 16, 25]. We note, however, that most of the previous techniques use sensing hardware that may not be available on current phones. As the field of biometric key generation continues to advance, our solution will become more applicable.

### 4.3 Division of Capabilities

The primary idea behind our approach is to divide access capabilities among a variety of entities. To accomplish this, we leverage secret sharing. Secret sharing [39, 11] is a cryptographic technique for distributing trust among a number of parties. Specifically, a secret sharing scheme allows one to divide a secret value  $s$  into  $n$  pieces or shares such that any combination of  $t$  of those shares allows the secret to be reconstructed.  $n$  and  $t$  can be chosen as any values from  $\mathbb{N}$  such that  $t \leq n$ , and a group with fewer than  $t$  shares cannot gain any information about the secret  $s$ .

The objective of our architecture is to allow people to have quick access to their own PHR at all times, to enable medical personnel to access people’s PHRs in emergencies, and to disallow all other attempted accesses. Fundamentally, dividing access capabilities allows us to associate the credentials of different entities with varying weight in terms of the entities’ right to access a record. We distribute these partial weighted rights such that in most cases no one credential is sufficient to obtain access to a record, but appropriate combinations of credentials are.

The access rights for each medical record can be described with a tree as illustrated in Figure 1. Each node of the tree represents a value with the root  $K_r$  being a secret key that allows decryption of a corresponding health record. The non-leaf nodes also represent threshold gates implemented

using a secret-sharing scheme [39, 11], and each is labeled with the number of child values that must be known to obtain the node’s value. For example, the value at node  $Y$  is a secret shared among the values at nodes  $B$ ,  $C$ , and  $X$ . If any two of those values are known, the value at  $Y$  can be computed. Lastly, the leaves of the tree ( $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$ ) represent the available shares of  $K_r$ , and they are labeled above by their proposed source. A graph of the tree, along with a description of the threshold gate at every internal node, is stored on the phone, so certain combinations of the shares allow one to obtain  $K_r$  and decrypt the health record. The precise storage of the shares themselves is discussed in Section 5.1, but their availability can be summarized as follows:

- BTG (“Break-The-Glass”): a share that is used only as a last-resort access mechanism if all other shares are unavailable. It can only be obtained through a special authorization process.
- Password: a share accessible with the PHR owner’s password.
- EMT: a share that is available to EMTs.
- Face: a share that is accessible with the PHR owner’s face.
- Finger: a share that is accessible with the PHR owner’s fingerprint.

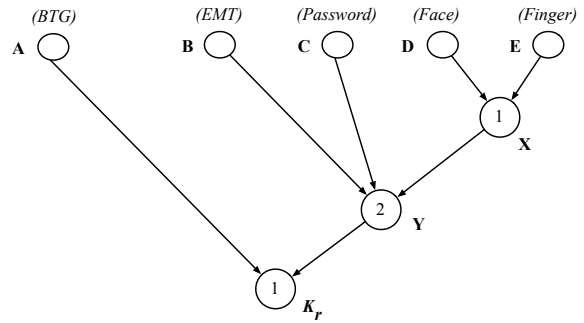
With these shares in mind, notice how the access tree in Figure 1 controls access in various situations. The owner of a PHR can enter her password and take a quick picture of her face, for example, to view her own record whenever she wishes. However, an adversary who obtains one of either her password or a clear image of her face, perhaps by finding her unconscious, remains unable to see the PHR. An EMT, on the other hand, who arrives to help an unconscious person can scan or take a picture of the person’s finger and use her EMT share to access the PHR. Finally, suppose an authorized medical professional finds someone in a health emergency but has no EMT share available or cannot obtain a suitable face or finger biometric. She can authorize herself to a BTG authority and obtain the BTG share, allowing her to decrypt and view the medical record.

Although we offer the tree in Figure 1 as one particular way of distributing access capabilities, our method is general. It can be adapted and modified according to the results of user studies and the development of new technologies such as improved feature extraction. For instance, if usability research found that a speech biometric was also desirable, namely in non-emergency situations, the tree could be modified to accommodate that.

The reader may find that our solution resembles attribute-based encryption [32, 10, 22, 21, 43]. Attribute-based encryption, however, is specifically designed to prevent collusion (or combining attributes from different entities) while our intention is to allow different parties to bring their credentials together at appropriate times to gain information access. Attribute-based encryption is also much more computationally demanding than the approach we propose.

## 5. DISCUSSION

We have described our general method for distributing access capabilities. We now discuss details essential to implementing our architecture.



**Figure 1: PHR access tree.** Each of the internal nodes is a threshold gate labeled with the number of child values that must be known to obtain the value represented by that node. The value at the root of the tree is a key  $K_r$  for decrypting a corresponding medical record. The leaves represent shares of the key available to different parties with certain features or knowledge.

## 5.1 Share Management

In order to securely manage the shares of key  $K_r$  as shown in Figure 1, each share is encrypted and stored on the user’s phone. We discuss methods for doing this below.

### 5.1.1 EMT Share

Recall that one of the shares is designated for use by EMTs. This share is created to provide EMTs with the capability to view the user’s PHR in an emergency. To have immediate access to the user’s PHR, EMTs must possess some information to allow them to decrypt the share. We explore the use of inexpensive trusted hardware, such as smart cards, to minimize the possible spread of credentials that have been lost or compromised. However, even with the use of smart cards, methods for revoking the EMT access rights need to be considered in the case of malicious EMTs or lost credentials.

We discuss three potential solutions for the management of the EMT share:

1. Each EMT is assigned her own public-private key pair by a trusted entity such as a regional health information organization (RHIO). This entity makes the public keys accessible over the Internet and programs each corresponding private key onto a smart card, which is assigned to each EMT. The entity also generates and publishes extra public keys that can be assigned in the future to new EMTs or EMTs who lose their smart cards. Client software on the user’s phone encrypts the EMT share with each of these public keys and saves the resulting ciphertexts. The client software can revoke a lost or compromised key by deleting the corresponding ciphertext.
2. Using a public-key broadcast encryption scheme such as that proposed by Boneh *et al.* [12] or by Park *et al.* [33], a trusted entity such as a RHIO generates a private key for each EMT and a broadcast public key. This organization programs the private keys onto a smart card for each EMT and maintains a list of non-revoked keys. The client software uses this list and the

broadcast public key to encrypt the EMT share such that only valid EMTs can perform decryption.

3. The same public-private key pair is used by all EMTs in a given region. This key pair is valid only for one day and is then renewed. Specifically, a trusted entity such as a RHIO generates a public-private key pair for each day and makes the public key available to all users many days in advance. Every day, the private key designated for that day is programmed onto a smart card that is carried by each EMT. The client software on a user's phone downloads all the available public keys periodically, encrypts the EMT share with each key, and stores the resulting ciphertexts. At the end of each day, the software deletes the ciphertext for that day. As one example of how often keys might be updated, suppose the keys are made available 150 days in advance. Then the client might update the ciphertexts of her EMT shares every 100 days or every time the number of stored, valid ciphertexts drops below 50.

These approaches have several tradeoffs with respect to key revocation, user requirements, bandwidth, and auditing. The first and second options are convenient in that they do not require EMTs to update their keys each day. Of these, the second is more storage efficient because it does not store a single ciphertext for every EMT. However, even option one, the least space-efficient of all three of the approaches, requires a fairly insignificant amount of space on most modern smart phones, which have on the order of tens of megabytes of storage or more.<sup>2</sup> The primary benefit of the first solution over the second is that key revocation is simpler and completely transparent to the user. With the first method, if a smart card is discovered missing, it can be reported, and a central trusted entity can notify all phones to delete the corresponding ciphertext, without any interaction (or even knowledge) from the user. (Recall that extra public keys are made available in advance, which can be re-assigned to EMTs.) In the second approach, if a key needs to be revoked, the shares must be re-encrypted using the new list of valid/revoked keys. Doing so, however, requires the EMT share plaintext, which, of course, cannot be stored directly on the phone. We discuss storage of all the shares for update purposes in more detail in Section 5.1.5, but from the user's perspective, accessing a plaintext share (for re-encryption, for example) requires entry of a password and a biometric. Because completely transparent key revocation is more convenient and immediate, we prefer option one to option two.

The third solution presented is more robust than the first two against compromised private keys in some ways. Using the third solution, keys can be revoked even when the user has no service since the phone does not require a specific notification to revoke a key. Additionally, because keys are only valid for a short period, the compromise of a current key is not effective on ciphertexts obtained after 24 hours, at most. The third solution also consumes less bandwidth. Each phone only needs to check for updated keys every 100

<sup>2</sup>If we use the Integrated Encryption Scheme (IES) [40, 15] with NIST's key-size recommendations through 2030 [9], an encrypted EMT share would be 60 bytes in size. Assuming the client software encrypts an EMT share for 1,000 EMTs within the phone's current region, the ciphertexts would only require approximately 60 KB of storage.

days in our example, whereas in the first two methods, every phone must check for revoked keys much more regularly.<sup>3</sup> Lastly, a tradeoff of the third solution is that it does not accommodate auditing quite as well as the first for online adversaries. That is, under option one, if an EMT accesses a user's record, the user's phone knows which EMT share ciphertext was decrypted and can log an access by the corresponding EMT. Since option three uses one universal EMT ciphertext for every day, that type of auditing is not possible. Auditing can be enabled, however, by implementing the client software to require smart cards to authenticate themselves uniquely before providing ciphertexts. With these considerations in mind, we suggest the third method for storing EMT shares due to its key revocation and bandwidth usage benefits. However, the first method could also be implemented if key updates for the EMTs are found to be excessively inconvenient.

Recall that even if an EMT share is compromised, an adversary must also have a user's biometric or password to view her medical record.

### 5.1.2 BTG Share

In the event that a medical professional needs to decrypt someone's PHR but cannot do so using other methods, she must decrypt the BTG (break-the-glass) share on that person's phone. Availability of such a last-resort mechanism is important because unexpected circumstances like malfunctioning cards, forgotten passwords, and others can arise. Because the BTG share alone is sufficient to decrypt someone's PHR (see Figure 1), we propose storage methods where access to one user's BTG share does not help someone gain access to that of another user. In the same light, we intentionally avoid giving any one type of medical professional access to the shares and rather limit the number of people with such capabilities.

To limit these capabilities, we utilize what we refer to as a BTG authority, a trusted third party such as a RHIO that centrally manages access to BTG shares. This could be the same organization that manages EMT keys, and it could also be distributed to reduce necessary trust. Its main purpose is to store and potentially release private information that can be used to decrypt the BTG shares stored on the phones within its region. That is, if a medical professional needs access to someone's PHR, she can call the BTG authority and authenticate herself possibly with a password and/or secret questions. If authentication succeeds, the BTG authority sends the necessary private information to the professional's phone, which can then be used to decrypt the PHR.<sup>4</sup> The BTG authority also logs the release of the private information to the medical professional, who becomes legally accountable for inappropriate breaches of privacy.

More specifically, users and the BTG authority interact as follows to store and manage the BTG shares: Each user registers her phone with the BTG authority under some unique phone identifier. This might be the same identifier that is used by the phone's service provider. The authority main-

<sup>3</sup>Some of the bandwidth needs could be reduced by implementing a method for broadcasting revoked keys to all phones at periodic intervals. However, designing such a mechanism introduces its own complexities.

<sup>4</sup>The medical professional's phone might engage in some protocol over a Bluetooth connection for fast transfer of keys or ciphertexts to the user's phone.

tains a public-private key pair for every phone and provides users with their associated public keys.<sup>5</sup> Each user stores her phone’s public BTG key and an encryption of the BTG share under it. In the case of an emergency, a medical professional needs to identify the user’s phone<sup>6</sup> in addition to authenticating herself to the BTG authority in order to obtain the private key associated with the phone. After its release, the BTG authority generates a new key pair for the user’s phone, which consequently updates its encryption of the BTG share using the new public key.

Using this approach, private information for one person’s BTG share is not useful for accessing another person’s BTG share or PHR. Furthermore, only a small number of people, i.e. those working at the BTG authority, have access to keys, and those who gain access are accountable for it. The downside is that the person accessing the BTG share must have a usable means of communicating with the BTG authority. Unfortunately, this seems unavoidable without allowing people to carry BTG access capabilities with them, which may be excessively permissive.

### 5.1.3 User Password Share

A user-selected password is used to encrypt one of the shares of the key  $K_r$  (the password share). One significant downside of using a key generated from a password is the fact that human-memorable passwords (or keys) often possess insufficient entropy, allowing for offline dictionary attacks. To address this potential weakness, we suggest the use of one of two potential protocols.

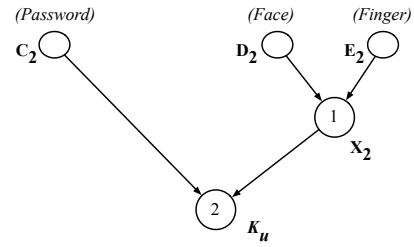
As one option, the time required to process passwords can be significantly increased to hinder dictionary attacks. Manber presents one approach to such password processing where each password is appended by a secret, randomly chosen salt that is never stored [27]. Every time the user enters her password, the receiving system finds the secret salt by linearly searching every possible salt value. The size of the salt is chosen so that the time required to process a single password is negligible, yet processing many passwords (i.e., during a dictionary attack) requires a significant amount of additional work.

As a second option, the user’s phone can (transparently) engage in a protocol with a semi-trusted server every time she enters her password to obtain a strong secret. The server never learns any information about her password or her strong secret as long as it does not obtain her encrypted PHR (in which case, it could launch a dictionary attack). The downside of this password-hardening step for the user is that she cannot use her password if her phone does not have network access. To allow for increased usability in situations with low cell-phone reception, we provide the user with an option to bypass this password-hardening step and derive the encryption key directly from the password itself. Even when this password-hardening step is disabled, our distribution of access rights (Figure 1) only allows adversaries with knowledge of a biometric or EMTs to launch a dictionary attack.

We suggest the password-strengthening technique intro-

<sup>5</sup>One way the BTG authority can keep private keys with minimal storage is to make each private key the output of some pseudo-random key derivation function.

<sup>6</sup>One could implement a Bluetooth protocol that automatically provides the user’s phone ID to the caller’s phone, which automatically sends the ID to the BTG authority.



**Figure 2: Access structure for obtaining  $K_u$ , the key used to encrypt copies of all shares of  $K_r$ .**

duced by Ford and Kaliski [20]. The idea is as follows: A password-strengthening server stores a high-entropy value for each of its registered users. When a user wishes to obtain a strong secret from her password, she sends a blinded version of her password to the server. The server uses the high-entropy value associated with that user to transform the user’s blinded password and returns the result. To allow the user to verify that the result was computed correctly, the server might also send a corresponding proof. The user is then able to unblind the result to obtain a strong secret. Because of the blinding component, the server cannot gain any knowledge of the user’s password or the strong secret. Furthermore, since only the server has knowledge of the high-entropy value associated with each user, an adversary cannot compute the secret. Finally, the user generates a key from her obtained secret using a key-derivation function, which is used to encrypt her password share.

### 5.1.4 Biometric Shares

Some of the shares of  $K_r$  are designated for access with the user’s biometrics. Our architecture uses these biometrics to generate cryptographic keys, which are then used to encrypt corresponding shares of  $K_r$ .

To ensure that the biometric shares can be reliably decrypted, the client software must provide an interface that allows the user or an EMT to easily enter the user’s biometrics. Previous research has developed and implemented methods of using face and hand biometrics [36] and finger biometrics [2] to authenticate users on modern cell phones. In combination with the feature identification introduced in these studies that use current phones, steadily improving cameras and fingerprint scanners<sup>7</sup> may provide hardware capable of obtaining sufficiently detailed and accurate biometric data to generate keys. We refer the reader to other and ongoing research for the generation of such keys [47, 17, 16, 25, 8, 7].

### 5.1.5 Share Storage for Updates

Recall that the plaintext shares of  $K_r$  (EMT, BTG, password, and biometrics) must be accessible to the user for efficient updates of EMT keys and share re-encryptions as discussed in Section 5.1.1. In addition to storing the shares of  $K_r$  as we have discussed thus far, all of the shares are also encrypted using credentials that are readily available to the user. Figure 2 illustrates one possible access structure that can accomplish this, using the notation that is shown in Figure 1. As depicted in Figure 2, the user can provide

<sup>7</sup>such as those on Lenovo ThinkPad notebooks (<http://www.thinkpad.com/>)

her password and a biometric to decrypt all of the shares of  $K_r$ .<sup>8</sup>

Let us look at an example. Suppose a user needs to update ciphertexts of her EMT share to incorporate new EMT public keys. To encrypt the share with the new keys, she requires access to its plaintext value. Thus, she enters her password and takes a picture of her finger, which ultimately allows her to obtain the key  $K_u$ . From this point, she uses  $K_u$  to decrypt all the shares (one for each of the leaves in Figure 1), and can then generate the new encryptions of the EMT share using the plaintext share and the new public keys.<sup>9</sup>

## 5.2 Protocol

In this section we detail the protocol steps executed by the entities of our system. We assume that the user of our system has an account with a personal health information centralization service such as Google Health [3] or Microsoft HealthVault [5] and all of her health records are stored online with that service.

**Initialization.** The initialization process is illustrated in Figure 3. The user begins by using client software on her phone to synchronize with the health information centralization service and downloads her PHR data. The software generates a random secret key  $K_r$ , divides it into shares as described in Section 4.3, and uses it to encrypt the PHR, deleting the plaintext version. The client software then downloads the EMT public keys ( $K_{ei}$ ) for the next 150 days, for example. The software uses the keys to encrypt its EMT share, once with each of the 150 keys. Each corresponding private key is programmed on a daily basis onto smart cards carried by every EMT. Next, the client software registers her phone with the BTG authority under a unique identifier. The BTG authority provides a public key ( $K_b$ ) associated with the phone that is used to encrypt the BTG share. After encrypting the BTG share with this public key, the client software asks the user to enter a password and input each of her biometrics. It generates different keys ( $K_p$ ,  $K_f$ , and  $K_c$ ) from each and encrypts the corresponding shares using them. If the user chooses, she may use a strengthening server to obtain a strong secret from her password that is not susceptible to dictionary attacks. Lastly, the software generates another key  $K_u$  that is used to encrypt all the shares as described in Section 5.1.5.  $K_u$  can be obtained with the user’s password and biometrics. As such, the shares can be accessed if they need to be re-encrypted with new keys. All plaintexts and keys are securely deleted from the phone at the end of initialization. In what follows, we detail the protocols followed in varied emergency situations.

**Conscious user.** If a conscious user wants to access his record, he simply provides his password and biometric information to his phone. Using the password and biometric,

<sup>8</sup>The password and biometric shares for the tree in Figure 2 are stored in the same manner as the other password and biometric shares, described in Sections 5.1.3 and 5.1.4.

<sup>9</sup>Some secret-sharing schemes would allow for values to be determined *up* an access tree, which could provide another method of obtaining plaintext shares. However, such an approach is not generalizable to all secret sharing schemes, and would not work for complex access trees.

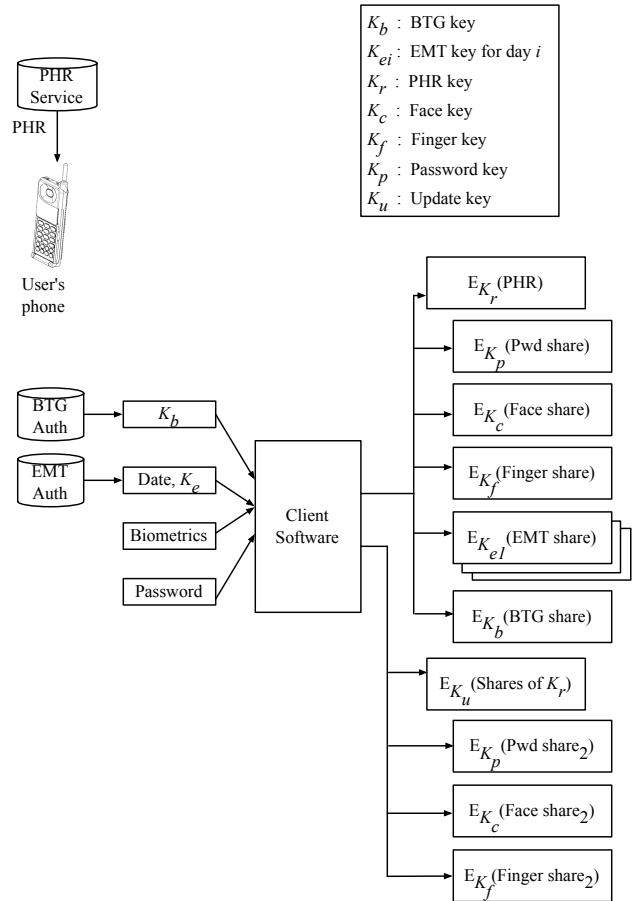


Figure 3: Initialization process.

the client software decrypts the respective shares stored on the phone and uses them to compute the key  $K_r$ . The software then decrypts and displays the record.

**Unconscious user.** In the event that a user is found unconscious, an EMT may require immediate access to the user’s health record. Although the EMT does not know the user’s password, she obtains his biometric information by taking a picture of his face, for instance. She may either take the picture with the user’s phone or transfer the user’s encrypted health record and encrypted shares to her own device (using Bluetooth, for example) and take a picture with it.<sup>10</sup> The client software converts the image of the face to a key  $K_c$ , which is then used to decrypt the share designated for that biometric. Additionally, the EMT uses her smart card to decrypt the EMT share on the user’s phone. The phone or the EMT’s device provides the smart card with the encrypted EMT share (possibly over Bluetooth), which the smart card decrypts and returns. With both shares, the client software computes  $K_r$ , decrypts the PHR, and displays it to the EMT.

<sup>10</sup>Phone-locking mechanisms can be modified to allow for transfer of encrypted data or biometric/smart-card reading while locked.

*Break-the-glass scenario.* Consider the case in which a user is unconscious and an EMT is unable to obtain a valid biometric reading. In such a situation, the break-the-glass protocol must be invoked to obtain immediate access to the user's health record. To begin, the EMT calls the BTG authority, identifies herself, and provides the unconscious user's phone ID. She authenticates, possibly by providing a password, answering some secret questions, or taking a picture of herself making a requested gesture. If she authenticates successfully, the BTG authority sends a private key  $K'_b$  corresponding to the user's phone, and makes a record that the key was released to that EMT. The key  $K'_b$  is transferred from the EMT's device to the user's phone. Once the phone receives it, the client software decrypts the BTG share, computes  $K_r$ , and decrypts the user's PHR. Following the incident, the BTG authority notifies the user that his PHR was accessed. It also generates a new key pair for his phone.

## 6. FUTURE WORK

One important property we would like to incorporate into our design is granular access control over the health records. In particular, we seek to provide the user with the flexibility of revealing only certain portions of her health record to different groups of people. For example, someone may wish to prevent her psychologist from viewing her gynecological records and likewise with her gynecologist and psychological information, but she may prefer for emergency physicians to have access to both. One natural cryptographic tool for implementing such control is attribute-based encryption (ABE) [32, 10, 22, 21, 43], which allows one to specify attributes that are required to decrypt different ciphertexts.<sup>11</sup> However, the exact details and challenges of using ABE to incorporate granular access control into our current architecture remain unexplored. Designing an efficient and scalable solution of doing so is a possible area of future work.

In addition, we seek to create a working prototype of our architecture on cellular phones, likely using Google's Android platform [4]. We would like to integrate the implementation with global PHR services such as Google Health [3] or Microsoft HealthVault [5].

## 7. CONCLUSION

We have presented an architecture for confidentially storing personal medical data on smart phones while keeping it available at desired times such as emergencies or instances of personal review. We suggest one possible delegation of access rights to allow for record access in situations where they seem most useful yet appropriate. However, our general methods are flexible and our solution could be adapted as practical usability issues are understood more thoroughly and as technology evolves. We also outlined possible methods for managing the information shares that allow for record access, discussing tradeoffs of various approaches. We believe our architecture is practical and hope to implement it on current cellular phones in the near future.

<sup>11</sup>This could actually be a formula of attributes that must be satisfied, and the formula may apply to either the ciphertext or rather to an entity's key, depending on the ABE scheme used.

## Acknowledgments

We thank Umesh Shankar for his helpful feedback. We also thank Mastooreh Salajegheh, Andres Molina, and Kevin Fu for their useful comments on a draft of this paper.

## 8. REFERENCES

- [1] AllOne Mobile: AllOne Health. <http://www.allonemobile.com/>.
- [2] Classifeye. <http://www.classifeye.com/>.
- [3] Google Health. <https://www.google.com/health/>.
- [4] Google Mobile: Android. <http://www.google.com/mobile/android/>.
- [5] Microsoft HealthVault. <http://www.healthvault.com/Personal/index.html>.
- [6] W. A. Arbaugh, D. J. Farber, and J. M. Smith. A secure and reliable bootstrap architecture. In *IEEE Symposium on Security and Privacy*, 1997.
- [7] L. Ballard, S. Kamara, F. Monrose, and M. K. Reiter. Towards practical biometric key generation with randomized biometric templates. In *CCS '08: ACM Conference on Computer and Communications Security*, 2008.
- [8] L. Ballard, S. Kamara, and M. K. Reiter. The practical subtleties of biometric key generation. In *USENIX Security Symposium*, 2008.
- [9] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for key management. NIST Special Publication 800-57, 2007.
- [10] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, 2007.
- [11] G. Blakley. Safeguarding cryptographic keys. In *AFIPS '79: American Federation of Information Processing Societies National Computer Conference*, 1979.
- [12] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO '05: Advances in Cryptology*, 2005.
- [13] CapMed. icePHR mobile. <https://www.icephr.com/iceMobileHelp.htm>.
- [14] CapMed. Personal healthkey. <http://www.capmed.com/solutions/applications.asp>.
- [15] Certicom Research. Standards for efficient cryptography, SEC 1: Elliptic curve cryptography, 2000.
- [16] Y.-J. Chang, W. Zhang, and T. Chen. Biometrics-based cryptographic key generation. In *ICME '04: IEEE International Conference on Multimedia and Expo*, 2004.
- [17] B. Chen and V. Chandran. Biometric based cryptographic key generation from faces. In *DICTA '07: Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, 2007.
- [18] F. Dillema and S. Lupetti. Rendezvous-based access control for medical records in the pre-hospital environment. In *HealthNet '07: ACM SIGMOBILE International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*, 2007.

- [19] A. Ferreira, R. Cruz-Correia, L. Antunes, P. Farinha, E. Oliveira-Palhares, D. Chadwick, and A. Costa-Pereira. How to break access control in a controlled manner. In *CMBS '06: IEEE Symposium on Computer-Based Medical Systems*, 2006.
- [20] W. Ford and B. Kaliski. Server-assisted generation of a strong secret from a password. *WETICE '00: Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2000.
- [21] V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP '08: International Colloquium on Automata, Languages and Programming, Part II*, 2008.
- [22] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS '06: ACM Conference on Computer and Communications Security*, 2006.
- [23] T. Hinkamp. System providing medical personnel with immediate critical data for emergency treatments. Patent Application Publication 11/510,317, 2007.
- [24] Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC). Break-glass – an approach to granting access to healthcare systems. NEMA (National Electrical Manufacturers Association-USA), COCIR (European Coordination Committee of the Radiological and Electromedical Industry), JIRA (Japan Industries Association of Radiological System), 2004.
- [25] Y. J. Lee, K. R. Park, S. J. Lee, K. Bae, and J. Kim. A new method for generating an invariant iris private key based on the fuzzy vault system. *IEEE Transactions on Systems, Man, and Cybernetics*, 38(5), October 2008.
- [26] P. Lukowicz, U. Anliker, J. Ward, G. Troster, E. Hirt, and C. Neufelt. AMON: a wearable medical computer for high risk patients. In *ISWC '02: International Symposium on Wearable Computers*, 2002.
- [27] U. Manber. A simple scheme to make passwords based on one-way functions much harder to crack. *Computers and Security*, 15(2), 1996.
- [28] MedicAlert. E-healthkey. <http://www.healthcentral.com/migraine/reviews-202629-5.html>.
- [29] C. Myers. Which prescription drugs could create a deadly combination? *ABC13 Eyewitness News*, May 2007. Available at <http://abclocal.go.com/ktrk/story?section=news/health&id=5320077>.
- [30] S. Nepal, J. Zic, F. Jaccard, and G. Kraehenbuehl. A tag-based data model for privacy-preserving medical applications. In *EDBT '06: Current Trends in Database Technology*, 2006.
- [31] P. Olla and J. Tan. Personal health records systems go mobile: Defining evaluation components. *Mobile Health Solutions for Biomedical Applications*, 2009.
- [32] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *CCS '07: ACM Conference on Computer and Communications Security*, 2007.
- [33] J. H. Park, H. J. Kim, M. Sung, and D. H. Lee. Public key broadcast encryption schemes with shorter transmissions. *IEEE Transaction on Broadcasting*, 54(3), September 2008.
- [34] D. Povey. Optimistic security: A new access control paradigm. In *NSPW '99: Workshop on New Security Paradigms*, 1999.
- [35] D. Preuveneers and Y. Berbers. Mobile phones assisting with health self-care: A diabetes case study. In *MobileHCI '08: Conference on Human Computer Interaction with Mobile Devices and Services*, 2008.
- [36] J. Rokita, A. Krzyżak, and C. Y. Suen. Cell phones personal authentication systems using multimodal biometrics. In *ICIAR '08: International Conference on Image Analysis and Recognition*, 2008.
- [37] L. Romney. Cirque de Soleil settles with gymnast fired over HIV. *Los Angeles Times*, April 2004.
- [38] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn. Design and implementation of a TCG-based integrity measurement architecture. In *13th USENIX Security Symposium*, 2004.
- [39] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), 1979.
- [40] V. Shoup. A proposal for an ISO standard for public key encryption (version 2.1). Available at [http://www.shoup.net/papers/iso-2\\_1.pdf](http://www.shoup.net/papers/iso-2_1.pdf), 2001.
- [41] P. Tang, J. Ash, D. Bates, J. Overhage, and D. Sands. Personal health records: Definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association*, 13(2), 2006.
- [42] United States Court of Appeals. Stephen C. Orr vs. Wal-Mart Stores, Inc. *Appeal from the United States District Court for the District of Nebraska*, (01-2959), July 2002.
- [43] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290, 2008. Available at <http://eprint.iacr.org/2008/290/>.
- [44] A. Wright and D. Sittig. Encryption characteristics of two USB-based personal health record devices. *Journal of the American Medical Informatics Association*, 14(4), 2007.
- [45] A. Wright and D. Sittig. Security threat posed by USB-based personal health records. *Annals of internal medicine*, 146(4), 2007.
- [46] W. Wu, J. Cao, Y. Zheng, and Y. Zheng. WAITER: A wearable personal healthcare and emergency aid system. In *PerCom '08: IEEE International Conference on Pervasive Computing and Communications*, 2008.
- [47] X. Wu, N. Qi, K. Wang, and D. Zhang. A novel cryptosystem based on iris key generation. In *ICNC '08: International Conference on Natural Computation*, 2008.