

Anonymous Medicine

jbrohan@tradersmicro.com
2010 07 08 rev 2101 01 07

This paper explores ways in which some medical data can be captured and maintained anonymously. This is done by using Hashing functions like MD5. Hashing transforms a key such as the user's phone number into a 32 character string. There is no way to reverse the process. The hash code cannot be transformed back into the original key. The MD5 function is available in most computer languages and its code is usually open source, anyone can see how it's done, and it's done in a way that you can't undo it.

It is not proposed to store all medical records like this, just some test results or similar. In the main hospital computer there is a record like... *name = "Jo Blo", phone = "+1 555 123 1234", dob = "10-05-1965", hospitalno = "123123123"* and so on. Mostly this will be encrypted using a secret key like "k45772". If you know the key, and you can get access to the database, you can read all the records. They must keep this key secret. They do this by controlling access to the computers and restricting what certain classes of people can see and change. It works quite well, but it is expensive and uncertain. This is not the problem I am tackling here.

The blood glucose levels for a patient are stored on an insecure database in records with the following example structure. *HashPhone = "162a8105b749d21cc4b4691954800702", date = "05-12-2009", BloodGlucose = "0.72"*. Obviously (if we have access to the database) we can easily find all the records for "162a8..." and draw a graph of his blood Glucose. However we do not know (and cannot find out) who he is. To query this database we need to start with a phone number, for example "+1 555 123 1234" which of course hashes to "162a8..." and we have got his data.

This approach gains security from the fact that even if you did discover a pair, a hashed and its corresponding phone number, this would not help you find another one.

The patient's name and date-of-birth etc never go outside the main hospital computer system. Expensive and careful safeguards are in place here. For the peripheral system of daily blood glucose readings, there is no link back to the main database.

There is a possible challenge. American phone numbers are 10 digits long. We could hash all possible phone numbers and do a lookup to find the phone number corresponding to a particular hash code. There are however several techniques, called 'salting' to defend against this. For example there are several ways to write phone numbers (555) 123-1234 or +15551231234 and so on. Also we can add a few secret letters "555 I like cats 1231234" for example before hashing. This approach is called obfuscation, if you know it you can un-do it. However you first have to find it out, (it's usually kept quite secret in rarely accessed computer code) and then you have to do the billion or so MD5 transformations and then you can reverse-

lookup the database, if you can get to it through normal commercial defences. This may be regarded as really, really a lot of bother, to the extent that it is well nigh impossible. This would work only for the blood-glucose data, not for the weight data etc. the hacker would have to start again for those. On my laptop to MD5 a simple file at a rate of 237 seconds per million and requires 32MB. This means 1 billion records would take 10,000 times as long and require 320GB.

There is the small but finite possibility that two different phone or numbers could hash to the same hash code. An approach to the collisions problem is to hash the phone number normally and then sort the characters of the phone number and hash the sorted string, keep both hashes on the database as the key and the check-key. In the search both keys are used. This reduces the probability of collisions to an infinitesimal level.

Of course the patient can change his phone number which is a nuisance. Lost passwords can present a problem, because we simply do not know who is the owner of the record. Passwords are not normally used when adding data to the files, just when extracting it.

Several conclusions follow from this.

1. The Blood Glucose database can serve many hospitals. It can be quite independent of them.
2. The hospital computer system does not know how to hash the phone number, so they cannot read all the records in the database, just the ones for "this" phone number, one at a time.
3. The HTTP request for data can be made through HTTPS, so that both the requesting phone number and the response set of readings is encrypted against eavesdropping.
4. Different test results can be maintained by different companies and accessed from the main hospital computer, using the phone number or some other patient identifier as a search argument. The actual hashing function and obfuscation used is not known at the hospital.
5. The resources to manage an anonymous database are much less than managing a regular in-hospital one, where a patient has to be registered to one clinic or another.

There is another aspect of this anonymous approach. It is the patient who is in charge. It's his blood-glucose reading after all, not the doctor's, and it's up to the patient to share it with the doctor or with other medical/ statistical/ insurance/ etc resources. There are approaches like <http://www.epatients.com/> or <http://www.patientslikeme.com/> which encourage the ownership of disease management. It is widely thought that if a patient participates in the management of a medical condition he cures quicker or generally 'does' better than if he is passive to the doctor.

Example Applications in working prototype stage.

Teleuroflow.com captures the sound of the urine stream splashing into the water in a toilet using a regular cell phone. The sound file is processed to produce a tracing like a regular office uroflow instrument. The patient or the doctor can access these tracings through a web site which requires a phone number (of the cell phone used to record the sounds) and a user generated password. The user can decide to use the software spontaneously, or a doctor can

ask him to do it. There is no registration process. Payment is for access to a view of multiple tracings, using a code number bought through PayPal or issued by a hospital clinic.

WoundFollowUp.com monitors wound healing by using photos sent in by a visiting nurse or the patient himself. The pictures are sent to an email address composed of the user's phone number and a domain like WFUemail.com, and the pictures are stored until the wound care nurse(WCN) downloads them. The WCN uses a Wound Image Program to seek in the database if there are any of her patients with pictures waiting, and to download them automatically. In this case the patient has to register with the WCN for the system to work. Payment is from the WCN clinic on a per use or fixed plan.

ALZ-Locate.com uses an Android phone to find demented people who have gotten confused and lost while out walking (or driving). Again, it's hashed phone numbers as the key, and watchers (usually younger family members) register to view that phone number's GPS location, when the wanderer's partner phones them to say that the wanderer is overdue. No names at all, and the phone numbers are hashed in the system. Payment is in advancing the Android app from a limited trial period to a 2 year agreement.