**Anonymous Health Information Exchange (HIE) Transfer with Credibility Check against Fraud through Chaum Mixes and Crowds.**

*By: Aaron Silcott*

**Intro**

Current widely used technologies give us more computing power, faster data transfer, and more information than ever before. On a simple home network set up, it take under a second for a single packet of data to make a round trip to google.com’s servers. A simple google search for the phrase, “how long did this take?” generated about 715,000,000 results in 0.90 seconds. In roughly a second, any information one would ever need to know can be searched for and found; yet despite these great accomplishments, in 2004 only 18% of physicians utilized some form electronic medical record storage/tracking (HealthIT.gov). Without the use of technology to store and manage patient data, retrieving, transferring, updating, and storing must all be done by nurses and receptionist with physical hard copies of the records. This leads to longer wait time for records, select hours where records can be accessed, lack of scalability as more patients attend a hospital, synchronization issues across differing medical facilities and many more issues. All of these problems can become a major inconvenience when dealing with non-critical situation such as appointments with a doctor or obtaining lab results, but in scenarios where a person’s life is on the line such as an Emergency Room visit, every second counts and any mistakes or misdiagnosis could become fatal. It was reported by the Institute of Medicine that before 2004, an estimated 44,000 to 98,000 Americans die due to medical errors such as inappropriate treatments or misdiagnosis in emergency situations (Kohn, Corrigan, Donaldson, 2000). Malpractices and mistakes like these can be avoided by allowing for the transfer of Patient medical records because it will highlight certain flags in a patients record such as allergies, recent diseases, and/or injuries that could cause bad reactions to specific treatment options. President George W. Bush saw the potential that electronic medical records had and in 2004 set forth a 10 year plan to have all medical facilities switch to Electronic Health Record (EHR) system and allow for the development of standards regarding the transfer of patient data between valid locations, also known as Health Information Exchange, HIE (Kohn, Corrigan, Donaldson, 2000). Even with the support of the government, EHR is slow to catch on with only 57% of physicians utilizing it in 2011 (HealthIT.gov). The delay in adaptation is due to many reasons from economic issues such as a cost of switching to social issues like an increase in incorrect data due to copy and pasting of templates. Although all of the issues currently being looked into are important, I will focus specifically developing a solution to solve the problem of patient security when transferring and requesting data between different organizations that house patient data, and the issue of developing an auditable technique to hold doctors accountable for their requests/actions in regards to patients information. My goal of this research is to develop a solution that will accomplish the following 4 parameters. First, the patient that the information is being requested for must remain anonymous to all servers receiving the request excluding the final destinations (which will be the sender and the receiver). The second goal is to ensure that the sender of the request for information cannot be traced to the specified receiver. Third, the solution must allow for some form of doctor authentication to verify that the request has come from a genuine doctor who would have access and there must be an auditable mechanism in place to allow for the owner of the medical records to see who has accessed their information and why. Lastly, the request must be guaranteed to complete in a timely manner ensuring the doctor can access the needed information quick enough to make informed medical decisions. I will be looking at 3 specific network implementations in an attempt to see if they are viable candidates to support the requirements set forth, Chaum mixes, Crowds, and a hybrid of the two.

**History of Technology in the Medical Field**

Throughout the years EMR and EHR systems would continually be redefined and improved upon and eventually the two terms would represent different technologies. EMR came to represent an electronic file containing the standard medical and clinical data gathered in one provider’s office (HealthIT.gov). An example of these files would be the document a family doctor would keep about a patient over a period of time. EHR’s in today’s society are defined as, “[files] designed to contain and share information from all providers involved in a patient’s care. EHR data can be created, managed, and consulted by authorized providers and staff from across more than one health care organization (HealthIT.gov). These provide a much broader scope of the care which one receives and are instrumental to the transfer of synchronized medical data. With the development of a storage standards the next step in the 2004 edict by President Bush was to facilitate the development of Regional Heath Information Organizations (RHIOs). RHIOs are defined as a an organization, “that allow access to clinical data required to treat a patient across provider groups (Sutherland 2005).” Put differently, they act as storage center for patient data that hospitals, laboratories, and other medical facilities can “subscribe” to allowing for transfer and central storage of patient data. The protocols for the transfer itself is not governed by the RHIOs individually, it must adhere to the guidelines set forth by the HIE. Currently there are 3 types of HIE as defined by healthIt.gov,

*“Directed exchange gives health care providers the ability to electronically send and receive secure information – such as such as laboratory orders and results, patient referrals, or discharge summaries – to other health care providers involved in a patient’s care over the Internet via encrypted, secure, and reliable messaging. Query-based exchange gives health care providers the ability to find and/or request information on a patient from other providers and is often used for unplanned/emergency care. Consumer mediated exchange gives patients the ability to aggregate and manage their health information on the Internet. When in control of their own health information, patients can help transfer information between providers, correct inaccurate demographic, medical, or billing information, and track and monitor their own health.”*

**Project Scope and Road Map**

The goal of my research is to define a possible implementation of a network solution for query based HIE in ER situations where the patient’s records reside in an RHIO other than the one the current hospital is subscribed to. Direct exchange is not viable network structure in the case of Emergency Room visits as they are, usual, unplanned and sudden. The primary care physician wouldn’t know to send the records in the first place. Consumer mediated exchange may be possible in some cases where the reason for the ER visit is not life threatening, but this is not the case for all scenarios and would require the patients to remember things like logins etc. which could be difficult for patients in shock or physically not able, as well as requiring that the users have access to a device where they can mediate the exchange. The fact that the consumer mediated exchange would only work in specific scenarios leaves it undesirable as a HIE standard for ERs.

In order to take a closer look at how the different solutions work I will use the following case study to show how the data will be handled by that specific instance. Bob is a Gainesville resident whose primary care physician is a subscriber to a local RHIO located somewhere in Florida. Bob goes out of town to Indiana for vacation and break his leg. He is then rushed to a near-by ER for treatment. Bob is in a lot of pain and needs treatment immediately, unfortunately he is allergic to a specific form of anesthetic but is unable to remember which one due to the injury. The doctor will need to request his medical records from Bob’s RHIO in Florida.

For the scope of this paper there are a few assumptions that are made, they are as follows:

1. Because the current and expected number of RHIOs is small, <1000, it is assumed that each RHIO know about and maintains keys for public key cryptography from each of them.
2. The patient knows what RHIO his information is located in. Although technically this could be rectified by sending a query to every single RHIO, which is possible due to the previous assumption, but in doing so break one of the 4 goals which is to maintain anonymity for the patient as now every RHIO knows that Bob’s records were requested. This would also be a terrible load on the network potentially causing slowdown in response time. This oversight could be solved by having the insurance company track where records are stored but as there is no current infrastructure for this, I left it as an assumption.

**Solution Take 1 Chaum Mixes**

The first proposed solution is to use the canonical form of mix networks set forth by David Chaum in his paper. In short summary, the sender will hide who he is sending to by rerouting his message through a third party called a mix. In order to protect the content of the message, the sender will encrypt the message using the destinations public key so that the mix doesn’t know what the message is. The already encrypted message is then encrypted again, but this time with the mixes public key, which ensures that if the message gets stolen nobody know who the final target is, they only know a mix is receiving a message from someone. Once the mix receives the message it decrypts is using it private key which will reveal the next recipient and passes the message along. Once the intended recipient get the message it decrypts it and can see the message content. This process can also be cascaded to allow for a chain of mixes to increase the security from different forms of attacks (Chaum 1981). The message which will be sent in our network would consist of the patient being requested and some form of doctor ID. The passing of doctor credentials will allow for the receiving RHIO to ensure that the request did in fact come from a valid doctor and provides a “trail” which can be audited to show that records were accessed by him at this time. To adapt the RHIO intercommunication into a mix network, each RHIO must act as a Mix when selected by a sender. The final piece of the network still requiring defining is how the requested server send information back to the sender. Chaum also devised a solution to this issue where the sender of the message will attach an encrypted return path so that the receiver only sees what mix to send it too, then once received by that mix it will decrypt and see the next mix in the path (Chaum 1981). This means that the sender of the request must choose both the path to the recipient and back. One decision that must be made is, who is the sender? One possibility is the hospital that needs the patient’s record, but this would mean that the hospital would need to maintain a list of all existing RHIOs. A single facility requiring the knowledge of all RHIOs does not seem like such a difficult task, but this means every hospital across the nation would also need to track this data and when the scenario of a new RHIO joining or and old one leaving, the lack of scalability shines through. A better solution is to have the RHIO to which the requesting hospital belongs to as the sender. In terms of the case study, this means the hospital Bob is at would send a message to its RHIO saying that it was this patient’s information, which is stored at that RHIO. The paths would then be generated and the communication will take place. Once the data is acquired from Bob’s Florida RHIO and sent back to the Indiana RHIO, it will then be forwarded to the Hospital Bob was at. This is a much better solution as it abstracts the communication between the hospitals and outside RHIOs allowing for a scalable solution. As RHIO enter and leave, they only need to update the lists inside of other RHIOs instead of every medical facility in the nation.

**Considerations Mix Types and Attacks**

There are 5 mix firing strategies that a mix can use to disguise the message receiver from its sender as defined by Serjantov and Diaz. Threshold, after n messages are received fire, Timed, after n seconds fire, Timed pool, after n seconds fire some messages leaving p in a pool, threshold pool, after n messages fire n-p of them leaving p in the pool, and lastly timed dynamic pool, of n seconds if there are m messages fire m-p of them and leave p in the pool (Diaz, Serjantov, 2003). When considering pool mixes, because the messages that are held in the pool are selected at random, it allows for the possibility that a message would be held in the pool for an extended amount of time. Even if the probability of this happening is extremely low, because the chance that a message could be delayed for many rounds goes against one of the core goals of the solution which is to have some form of guarantee that the request will complete in a timely manner. Therefor any pool style mix will not be suitable as a firing strategy for a Chaum Mix style HIE. This leaves two possible styles to analyze, timed and threshold. Threshold mixes by nature only work when there is significant traffic flow through a mix, if there is not, the mix will wait until there is. This causes the same issue as pools in the fact that there is no guarantee of speedy delivery. If traffic conditions are low, by natural conditions or due to an attacker intentionally delaying traffic, a request will sit and wait, therefore Threshold mixes cannot be used either. The final type to consider is the timed mix which will fire all messages it hold at regular intervals. This solution is ideal in terms of speedy delivery but unfortunately is very vulnerable to trickle attacks, where an attacker will delay all messages excluding the target message effectively eliminating any chance of hiding the likability between sender and receiver (Serjantov, Dingledine, Syverson, 2013). One possible solution to combat this issue is to use a slightly modified version which will pad the batch being fire with dummy messages if a certain thresh-hold cannot be met. This will eliminate the vulnerability of both the trickle attack and the flooding attack (attacker sends n-1 messages at a mix where n is the threshold limit, in attempt to isolate the target message) (Serjantov, Dingledine, Syverson, 2013). Although this new style mix will thwart both of these attacks individually, it is unable to defend against if both are used in unison. The trickle attack would be used to delay all real traffic excluding target and the flood would then send n-1 messages at the mix to ensure that no dummy messages would be fired. In order to prevent this blended attack, a slight modification must be made to the canonical Mix communications.

**A Slight Change**

In order to prevent a flood attack from isolating a single target message in our modified dummy-padded timed mix, we need a way to verify that a message did in fact come from a valid mix. This can be handled by adding a verification code to the message being passed from one RHIO to another. This code or verifiable ID, which we will refer to as U, is a code that each mix will have and every other mix will know what it should be. This value U is then encrypted using the public key of the next mix in the chain and is then appended to the message being sent. Once the next mix receives this message it will decrypt and look for the U verification code. If it is found, then the message is admitted to the mix, if not, the message is discarded. This will stop a Flood attack from preventing the creation of dummy messages, and in doing so protect our message from all attacks excluding a compromised RHIO.

**Closing statements**

In the purely canonical sense, using Chaum mixes is not a viable strategy from this network. Without the probabilistic nature of pool mixes to protect against flooding attacks, general timed mixes are too vulnerable. But if the modifications stated above are made then this solution would work. The only attack not considered is the compromised node attack as this would effectively give an attacker a work around to the previous sender verification modification proposed. The reason this was not considered is because RHIO are not just simple servers on the web, they’re government and privately funded data centers where medical data is housed. The security set in places like these should be strong enough to prevent a center from becoming compromised. One major drawback of this technique is that when an RHIO wishes to join the list as a crowd it must query an existing RHIO to receive a list of the current ones. This process leaves the decision of determining if the new RHIO is not compromised to the RHIO to which the list was requested from. The final consideration to make is that this solution does not provide the architecture for doctor verification beyond the transfer of the verifiable ID. So a separate infrastructure would be needed.

**Solution Take 2 Crowds**

The canonical crowd network works as follows, a user will generate a message. It will then pass that message to a Jondo, who will then “flip a coin” to decide if it will forward the message to the target or to another Jondo (Reiter, Ruben, 1998). When a message is received on a Jondo, it is tracked in that Jondo by its message id and the connection it came in on, this information is stored in a virtual circuit and the coin flip determines that it will forward the message to another Jondo, the VC is updated with a new outgoing message id and the connection it is sent out on. When a Jondo wishes to enter or exit the crowd it must go through a Blender which would handle telling all the others that it is no longer a member or it would add it to the list of Jondo and alert all existing ones to its presence, as well as handle key distribution (Reiter, Ruben, 1998). In terms of implementing a crowd as a solution to our query based HIE every RHIO would act as a Jondo in the crowd. Some trusted national authority would then act as the blender. The contents of the messages would be very similar to that of the mix, the only difference is that the route is not selected by the requester, and the requester encrypts the patient and doctor ID with the targets public key, then the entire message is encrypted with the next Jondo’s public key. When that Jondo receives the message it decrypts it and can see the receiver.

**Consequences**

There are two main problems that arise with using crowds as a network solution for query based HIE.

First would be how to handle the joining or quitting of a Jondo. This process requires that all existing Jondo wipe their Virtual Circuits and redirect all traffic they have, so that the new Jondo has an equal probability of getting traffic routed through it. But when the VC wipes happen, what happens to the traffic currently going through the network? There are 2 options and neither are good, the worst of the two is to just drop all data currently traveling through the network. This could corrupt data being transferred, entire requests could be lost, and many other issues. The better option would be to let all currently travelling data finish their routes, then update while holding off all the rest of the traffic. Even though this solution is better than outright dropping the data, it still poses long delays which goes against the guaranteed speedy delivery goal.

Second is the probabilistic coin flip, this goes back to the same issue that pool Mixes caused in the sense that because there is a probability that the message will continually be bounced to new Jondos, one cannot guarantee that a patients data will be processed and returned quickly.

One good consequence that comes from the crowd structure is the existence of the Blender. This national third-party gives an easy point of entrance to and exit from the crowd.

**Closing Statements**

Crowds are not a viable network solution to the query based HIE. The probabilistic nature of the Crowds goes against one the of primary goals of this research and that is to provide a quick method of medical record transfers, and while it could be very fast, it could also be extremely slow. Also the entrance and exits of Jondos could prove to leave awkward transitions that would have detrimental consequences. But the existence of the Blender is a promising prospect in terms of management.

**Solution Take 3 Hybrid Model**

The goal of this solution is to see if there is any way to eliminate the down falls of the previous two solutions by pulling bits and pieces from one another. Mixes presented a strong case as a possible implementation so I based this Hybrid off of the fundamental structure provided by the canonical mix style and used concepts from crowds to cover the weaker points in the Mix architecture. The underlying message transfer used will be the one described in solution 1, it will use sender path building (with encryption at each step), and each mix will be the modified dummy-padded timed mix with verification ID added at each step between mixes. As stated in solution 1’s closing statements and consequences, this structure lacks a single point of entry for a RHIO which causes a lack of security in verifying new RHIO request as valid. The other issue that arose was the lack of infrastructure to support the authentication process of a doctor ID requiring a completely separate networking solutions. By taking the Blender Idea from Crowds and applying it to the hybrid model, it effectively eliminated bother concerns. The Blender would act as a verification point for new RHIO requesting to join as a mix. If the credentials are accepted, the Blender would then push out a notification to all existing RHIOs that a new one is joining and would distribute the public keys of the new one to all who already exist. If the Blender Stores a list of public keys from each Mix, then instead of the new Mix asking every member for its public key, the process can be skipped by the Blender simply giving the new Mix a copy of the list. One possible downside to the use of the Blender in mixes is that the interval by which a RHIO will join or leave will be relatively infrequent. RHIOs are not just servers that anyone can turn on at the drop of a hat. They are large business funded by multiple shareholders who would not invest without some form of stability in the organization. This means that for most of the year a Blender would sit idle, but I propose that the Blender can also be used at the verification server for doctor IDs. This would maximize the efficiency of the system by minimizing idle time and saving the cost of a separate architecture for doctor authentication.

**Conclusion**

The defined hybrid model not only a valid choice for a network solution, but rather it is the best choice as it eliminates the downsides of the previous two proposed structures. The Chaum Mix solution was did not have very many pit falls to begin with as the crowds did, but with the addition of the blender for key distribution and single point of entry, the hybrid model has alleviated one of the two main issues that Mixes had. But the downside of adding this blender was the amount that the server would be idle. Although this paper is not focused on economic issues, this amount of idle server time is enough to make a point about its inefficiency. With the proposition of also using the Blender as a doctor authentication sever, the idle time would be eliminated and a separate architecture would no longer be needed. The only consequence of this would be that it introduces a single point of failure for the system.

**References**

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, *24*(2), 84-90.

Diaz, C., & Serjantov, A. (2003, January). Generalising mixes. In *Privacy Enhancing Technologies* (pp. 18-31). Springer Berlin Heidelberg.

HealthIT.gov. (n.d.). Retrieved December 10, 2015, from <https://www.healthit.gov/patients-families/electronic-health-records-infographic>

Kohn, Linda T., Janet M. Corrigan, and Molla S. Donaldson, eds. *To err is human:: building a Safer Health System*. Vol. 6. National Academies Press, 2000.

Reiter, M. K., & Rubin, A. D. (1998). Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, *1*(1), 66-92.

Serjantov, A., Dingledine, R., & Syverson, P. (2003, January). From a trickle to a flood: Active attacks on several mix types. In *Information Hiding* (pp. 36-52). Springer Berlin Heidelberg.

Sutherland, J. (2005). Regional Health Information Organization (RHIO): Opportunities and Risks.