Verifiable and Redactable Medical Documents

Jordan Brown, Douglas M. Blough, PhD
School of Electrical & Computer Engr., Georgia Inst. of Technol., Atlanta, GA

Abstract

This paper considers how to verify provenance and integrity of data in medical documents that are exchanged in a distributed system of health IT services. Provenance refers to the sources of health information within the document and integrity means that the information was not modified after generation by the source. Our approach allows intermediate parties to redact the document by removing information that they do not wish to reveal. For example, patients can store verifiable health information and provide subsets of it to third parties, while redacting sensitive information that they do not wish employers, insurers, or others to receive. Our method uses a cryptographic primitive known as a redactable signature. We study practical issues and performance impacts of building, redacting, and verifying Continuity of Care Documents (CCDs) that are protected with redactable signatures. Results show that manipulating redactable CCDs provides superior security and privacy with little computational overhead.

1 Introduction

In the near future, electronic health information will be routinely exchanged among widely distributed entities and for a variety of purposes such as treatment, efficacy evaluation, medical research, and public health. To date, there are no proposals for building trust into health information exchange, aside from direct interactions between two trusted parties. While such direct interactions will be common, there are likely to be many indirect interactions as well. Prime examples are: 1) the use of personal health records (PHRs) by patients to store their health information and provide it to third parties on an as-needed basis and 2) research data repositories, which gather medical data from various sources and allow multiple research projects access to different data subsets. We address the problem of how to verify health information that is provided by an entity that was not the original source of the information. Establishing data provenance for trusted data analytics is especially problematic in distributed systems. The problem is also complicated by the need for intermediate entities to redact some of the information coming from the source, e.g. for privacy reasons.

To provide verifiability and redaction, we propose to use a cryptographic primitive known as a redactable signature [1, 2]. Using a redactable signature, the source of health information, e.g. a health care provider, can sign a medical document and provide it to another party, e.g. the patient who is the subject of the document. The second party will then be able to redact information in the document that they do not want to disclose and pass the document to other parties. An intermediate party can also merge data from different documents provided by different sources and create a new document, while maintaining the signatures of the data sources. The recipient of a signed document can verify all of the sources of data in the document and can verify that no other party has modified the data items since they were signed by their original sources.

As a simple example of the use of such a technology, consider a situation where parents must supply proof of vaccinations of their daughter for a summer camp or school that she will attend. In this situation, the parents can receive a complete electronic vaccination record signed by their health care provider. However, certain vaccinations, e.g. the HPV vaccine, might be considered sensitive by the parents and not required for attendance. The parents would be free to remove evidence that the child received the HPV vaccine and forward on the remainder of the vaccination record to the third party. The third party can cryptographically verify that the vaccinations listed were performed by the family's health care provider without seeing the redacted information. The second party, in this example the parents, is free to retain the complete record and can provide any subset of the health information contained in it on a per-use basis. Thus, the document can be provided in different forms as many times as needed.

In this paper, we report on a prototype implementation we have built for managing redactable and verifiable Continuity of Care Documents (CCDs). Our prototype handles the construction, redaction, and verification of these special CCDs. Using an example CCD that is in the public domain, we evaluate the

overhead of manipulating redactable CCDs, as compared to ordinary CCDs. We show that the costs associated with redactable CCDs are minimal, while the security and privacy advantages are great. While this paper has considered only basic redaction capability, future implementations could be augmented with additional capabilities that have been developed in the security community for redactable signatures, including the inclusion of simple disclosure policies coming from a document's source, which would limit the ways in which disseminating parties can redact it.

2 Background

2.1 Terminology

CCD - This refers to a Continuity of Care Document, which is a document containing a patient's medical history and recent treatment records. The CCD format has been standardized by HL7 as part of their Clinical Document Architecture [3]. The CCD provided by Dr. John Halamka[1] was used as an example for the implementation reported in this paper.

MHT - This refers to a Merkle Hash Tree, which is a binary tree containing information about data items that may be redacted by a user. In this paper, we focus on MHTs that include a digital signature, which allows arbitrary subsets of the data items to be electronically verifiable. Hash values are calculated for the data items and stored in the leaf nodes of the tree. Depending on the implementation, the data items might be stored separately or they might be stored in the leaf nodes with their corresponding hash values. In our implementation, we include the data items in the tree. Each internal node of the tree forms its own hash value by concatenating the hash values of its left and right children and calculating the hash of the concatenated values. Once the root calculates its own hash, the hash value is signed by an authority so that the data can be verified later. Any mention of MHT or "tree structure" herein refers to only the Merkle Hash Tree (not the CCD containing the data items used to construct the MHT). An implementation could combine the CCD and the MHT into a single object for transport but, for ease of explanation, we consider them to be separate items in the remainder of the paper. If redactable documents become the standard, we expect that the the structural tree information and root signature will be allocated their own designated section of the CCD. The structure of an MHT is shown in Figure 1.

2.2 Use Cases

We provide three example use cases for redactable and verifiable medical documents. Before presenting the use cases, we note that some use cases involve patient care, while others do not. Whenever medical documents are used in patient care, redaction presents ethical and legal challenges. These challenges, and the associated risk management practices that institutions must employ to address them, are largely outside the scope of this paper. However, we note the following:

- unsigned CCDs can easily be redacted without any way of detecting the redaction and so the capability we provide does not introduce any ethical or legal issues that are not already present with ordinary CCD use (the standard practice of encrypting and signing email covers the email transmission but does not cover the document itself, i.e. the document can be undetectably redacted prior to email transmission), and
- our enhanced redactable signature scheme presented in [2] can help address these challenges; in this scheme, a provider can encode the data in such a way as to prevent certain redactions from taking place without invalidating the signature.

The first use case involves medical providers giving copies of CCDs to a patient, from which the patient can then selectively choose medical data to provide to third parties while maintaining the signatures of the providers. This could be done, for example, in the context of a personal health record (PHR). Signed CCDs from different providers would be stored in the PHR and the patient would authorize construction of a redacted CCD with only the relevant information to be sent to third parties on an as needed basis.

[1] http://services.bidmc.org/geekdoctor/johnhalamkaccddocument.xml

This includes the example described earlier where vaccination information that is needed for school or summer camp enrollment can be provided without releasing an entire medical record, while still allowing the school/camp to verify that the data came from a legitimate medical provider.

A second use case involves medical research using data from electronic health records. In research, there is a process of approval for obtaining the records and access is typically granted for only those sections of the records that are required for the study. It is also common to have a hierarchy of access, where a research supervisor might be allowed to see a larger subset of the data than the assistants working under her. This produces the need for multiple levels of redaction while maintaining the verifiable nature of the data. In this scenario, CCDs with complete information can be stored in a centralized repository. When particular studies are authorized to access subsets of data from specific CCDs, the system can generate a redacted set of CCDs and supply them to the research supervisor or administrator. The supervisor or administrator (or software acting under their control) can then generate further redactions for individual research users, if necessary.

A third use case maintains signed information for data provenance, a form of record keeping. In this use case, signature by a provider allows verification that the provider did provide certain services and generate the included data for a particular patient. Any additions to the document would require signature by the provider of the new data, allowing verification that the document was modified by that provider also. This approach can extend to tracking modifications by individuals within an organization if the individuals with authority to enter information into a document are issued public/private key pairs. As an extension to the basic scheme, individuals or providers can also attach a record that is covered by their signature whenever they supply a CCD to a third party. This could include information such as to whom they are providing the CCD and what the stated use is. In this way, if a document is used in an unauthorized way, the party that is responsible can be identified. This capability requires a minor extension to the signature scheme, which combines both redactable and unredactable data under a single signature.

2.3 Related Work

The medical field is rapidly moving from the world of paper to the electronic domain. Applications such as personal health record (PHR) repositories and standards such as NHIN Direct [4] are leading the way for this kind of transition by allowing patients and doctors access to electronic medical forms and results. Microsoft HealthVault [5] is the most well known PHR repository. It has a large infrastructure in place to allow for the storing of patient medical records. However, PHRs are still fairly new applications and there are many downsides to these services currently. One of their disadvantages is that the data cannot currently be verified for validity and integrity. This creates a problem when the information is to be distributed, because consumers of the information cannot trust it and therefore cannot use it for critical purposes.

Redactable signatures were first introduced by Johnson, et al. [1]. The Johnson, et al., scheme and most subsequent redactable signature schemes are based on Merkle hash trees, which were introduced in [6, 7]. The implementation reported in this paper is an application and evaluation of the approach discussed in Bauer, Blough, and Cash [8]. Bauer, et al., extend the Johnson, et al., scheme to include trees containing data from multiple authorities in which each authority signs a sub-tree of its data. This functionality is the basis for our design and an overview is included herein; however, more information may be found in [8]. Since a single CCD might contain information from multiple health care providers, we need to include signatures from multiple authorities in the MHT structure, which is handled by the Bauer, et al., design.

A few papers have considered redactable signatures in the context of electronic health information [9, 10, 11]. In [9], a server-based approach allows a single redaction of a medical document at the section level. This approach does not permit documents to be passed to third parties for subsequent redaction, does not provide redaction to the granularity of individual entries in a document, and does not allow documents to contain health data from multiple authorities. In [10, 11], a redactable signature scheme that is tailored to structured documents, of which CCDs are an example, is presented and analyzed. This work proves the security of the scheme but does not provide an implementation nor a performance evaluation.

Several enhancements have been proposed for redactable signature schemes. For example, Bauer, Blough, and Mohan [2] show how to encode data dependencies within a MHT, which allows the data sources to enforce simple policies on how their data can be disclosed. The work by Miyazaki, et al. [12, 13], uses a commit vector and bilinear maps respectively to allow for data sanitizing, which is their term for redaction. Their work provides methods for hiding the number of items removed as well as controlling when items may no

longer be removed from the document. Haber, et al. [14], discuss the use of hash trees to allow for data redaction and data dependency trees in the context of data generalization and pseudonymization. Izu, et al. [15], describe a system, which they name PIATS, for tracking redactions in a way that does not prevent redactions from being made by an untrusted source but allows viewers of the data to see who redacted what data. We do not consider these enhancements in this work, because we are primarily concerned with the performance and practicality of basic redaction.

Other work has been done in the area of data integrity, which can be considered relevant to the work described here. The work done by Polivy and Tamassia [16] describes a process for using XML signatures in conjunction with Web Services on sets of data to provide easily distributed responses to queries on distributed authenticated dictionaries. Since the XML signatures and Web Services follow a set of common standards, the data is easily transferred from source to client and easily verified when obtained. This method could be used to send health information but does not account for the need to redact information from previously signed data. Bull, et al., take "content extraction signatures" from [17], which details multiple schemes for achieving redaction for a given source, and expand them to work with XML signature standards in both "single dimensional" [18] and "multidimensional" [19] scenarios. Here, single dimensional refers to treating each data item equally in redaction selection and multidimensional refers to grouping items that have dependencies on a key item such that those items may be present or removed only if the key item is disclosed and default to hidden if the key item is hidden.

We do not explicitly consider data confidentiality, typically achieved by encryption. However, redactable signature schemes can easily be integrated with encryption. Notable when discussing encryption for health information exchange is NHIN Direct [4], which is an emerging set of standards for secure health transactions. However, the standards deal only with a single verifiable source sending data to another entity. The standard provides for encryption together with a digital signature on the entire set of data items. This does not allow the recipient to further pass along only specific items from the set and maintain source verifiability. Our work addresses this by providing the patient with a dynamic set of health items, signed by medical authorities, which a patient may distribute selectively. This could be incorporated into NHIN Direct by expanding the standard or within other encryption schemes such as the work by Suenaga and Takada [20].

3 Design

The implementation is based on the design detailed in the paper by Bauer, Blough and Cash [8]. A basic description of the design is provided in this section, along with some minor differences in our implementation compared to [8]. The interested reader is referred to [8] for an in depth design description and analysis.

3.1 Tree Structure (MHT)

In the design section of [8], the nodes were differentiated by altering the hash value of the node to determine if the node was a leaf node or an intermediate node. However, our implementation uses enumeration values to differentiate the nodes. It is a very similar method but the design difference should be noted. A basic visual of the structure is provided in Figure 1.

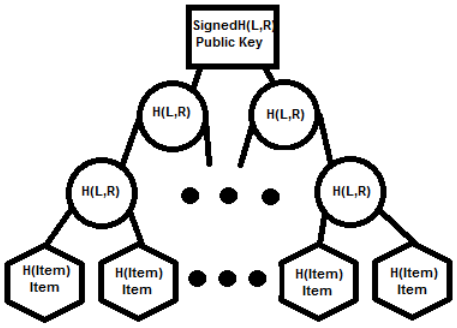

Figure 1: TreeStructure

Date / Time:	Oct 22, 2007	Oct 19, 2006
Height (in)		
Weight (lb)	172.4	174.8
BMI	22.83	23.15
BP Systolic (mm Hg)	116	128
BP Diastolic (mm Hg)	72	81
Temperature (deg F)		
Pulse rate (/min)	53	51
Pulse rhythm	regular	regular
Resp rate (/min)	14	13

Figure 2: Example Vital Sign Readings from Halamka's CCD

3.1.1 Leaf Nodes/Health Items

Each individual piece of health data that should be redactable is stored in a separate node and these data items are formatted so that they may be repopulated into a CCD when needed. Once collected, the items are randomized and stored. Randomization helps reduce the information that can be inferred once data items have been redacted. If the health items are collected and stored in order, someone might be able to infer details of the medical history based on the gaps in the provided data items. For example, individual encounters are typically stored in order by date in the "Encounters" section of the CCD. So, if there are two unredacted encounters surrounding a redacted encounter, someone viewing the redacted document can infer that there is a missing encounter within a certain time interval, which might be more than the owner of the CCD wishes to disclose. After the data items are randomized and stored, the items in each leaf are hashed and the hash of the item is also stored in the same leaf node for verification.

3.1.2 Intermediate Nodes

Each of the remaining nodes calculates a hash from its left and right children in the tree. The hash values of the two children are concatenated and the new value is then hashed to form the hash of the intermediate node. This function is propagated up to the root of the tree until all of the nodes have a hash value.

3.1.3 Root Nodes

The entity providing the health data items located in the tree signs the root hash and places the signature in the root node. If multiple sub-trees are used, then each one of these trees has a signature at the root. These sub-trees are treated as leaf items and their hashes are propagated up to the top level root. That root hash is then signed as mentioned earlier. In [8], the signature of the hash was stored in another location, an auxiliary branch to the root node, and the three branches were used to calculate the hash of that node. In our implementation only the signature of the calculated hash value is stored in each root. In the prototype implementation reported in this paper, we store the public key of the signing entity in the root node. To be used in practical scenarios, the public key would have to be replaced by a certificate containing the public key and signed by a trusted certification authority, or by the name of the source assuming that that the recipient can obtain the source's public key through a separate trusted service.

3.1.4 Verification

When verification of the provided health data is needed, the process is similar to tree construction. The hashes for the items that are present are calculated and those hashes are combined, calculated, and compared up to the root hash value. If a data item has been redacted, the hash of that leaf is provided in its place. Once all hashes up to the root have been reproduced, the signature is verified with the public key from the source. If there are multiple sub-trees, this process is done to verify each sub-tree as well as the encompassing tree. Due to the hash function's one-way nature and collision resistance property, if either a provided data item or the hash of a redacted item is altered in any way, the calcuated root hash value will be different from the signed value and the alteration will be detected.

3.1.5 Redactions

Redaction of data items is done by removing the provided health item from its leaf node. The item to be redacted must be located within the tree before it is removed. To reduce overhead, all nonessential nodes are removed upon redaction. If a node has two children, which both contain no items beneath them, then the child nodes may be removed. To calculate the hash for verification, a node is kept until the parent node also verifies that the other child node has no data items beneath it. Therefore, as the result of a redaction call is propagated up the tree, each node checks the status of its children and removes them if possible.

3.1.6 Additions

Additions of data items to a CCD require a reconstruction of the associated MHT structure and a new signature, because the root hash value will change. As discussed earlier, the CCD source will simply construct

the new CCD and MHT structure and supply them to the intermediate party, i.e. the patient in our primary use case. The patient will then replace the older versions with the newer ones.

3.2 CCD Example

The following sections discuss the layout of the document and the work done to collect the medical items.

3.2.1 CCD Document Structure

For the results reported herein, an XML formatted CCD document was taken as the input. Specifically, the aforementioned CCD of Dr. John Halamka was used. An example of the visual format of the Vital Signs section of this document is provided in Figure 2. The elements in the document are represented in two ways in the XML document. First there is the table representation, shown in Figure 2, which provides formatting information for tools, such as browsers, to visually render the information. Each section of the document also contains a machine-readable representation of the table data, as well as other supporting data that are relevant. We focus on the machine-readable portions of the CCD in constructing the MHT structure.

To construct a MHT, we take advantage of the well defined structure of a CCD. This enables us to avoid treating each syntactical unit of the XML file as a data item and including it in the tree. Instead, we can focus on the specific health-related data that are present in the CCD and store only those items in the MHT. In this manner, the constructed MHT is much more compact and efficient to process as compared to one that was generated from an arbitrary XML file. Because a valid CCD must follow a specified structure, we can repopulate a skeleton CCD with the health data items after the data are verified and taken out of the MHT. Only the data from the machine readable representations are collected in this implementation and are stored as items that may be redacted at a later time. These health data items are augmented with the section from which they came. A specific set of data items can then be populated back into an empty CCD at a later time in order to provide them for a particular use. To reconstruct the visual portions of the document, relevant information regarding each entry is extracted from the machine readable portions and used to create the visual table. These tables are dynamically constructed, displaying only rows or columns which actually contain some data to display.

3.2.2 Document Handling

To understand the structure of the CCD and the types of data objects our scheme must handle, we analyzed all CCD sections and the format of their entries. For each section, common notable data locations were hardcoded into our code for populating the visual portion of the document. This allows a skeleton document with only the barest section headers to be dynamically constructed for viewing based on the entries provided by a patient. This may, from time to time, result in missing visual data due to oversight of a given field, but the machine readable entries always maintain the level of integrity and full specification as initially provided by the healthcare provider. For this reason, providers are strongly encouraged to avoid providing data that is contained in the visual portion of the CCD but not in the machine-readable portion.

In this implementation, data dependencies are handled similar to the work by Bauer, et al., [2]. Data dependencies may be seen, for example, in the results and vitals sections. In these sections, a single entry may have multiple independent portions of data.The data seen in Figure 2 would contain each of the columns in the vitals section table as a single entry representing the specific visit. In these cases, a patient may wish to redact some of the information found within the column but the surrounding entry header would still be required. To account for this possibility, a directed graph was implemented, which stores dependent health information in the following fashion. The highest level of the entry is stored in the root of the graph. This is usually the surrounding header for the health entry. Each one of the associated items, relevant to this entry, is stored one level down in the graph. Each of these second-level nodes has an associated hash value. The hash value of each of these nodes is calculated by hashing a concatenation of the hash of the data within the leaf and the hash output of the node above it (the surrounding header). Only the leaf nodes store the calculated hash values. This structure ensures that none of the information is verifiable unless at least an entire path of the graph is released. If a higher level node is released, there is no direct hash value to compare it to and if only a leaf is released the hash still depends on the upper level items so no verification is possible. To relate this data back to the normal tree structure, the hash that represents the entire graph is calculated,

by hashing a concatenation of all the leaf node hashes, and is stored in the MHT with the graph in the same node.

The approach just described ensures that each individual vital sign reading from a particular visit is redactable, while ensuring that no reading is supplied without the date of the reading also being present and jointly verified. Otherwise, it would be possible to attribute readings from one visit to a different visit.

We note that this scenario only covers two levels of depth and "OR"-type dependencies, but in future versions of our software, we plan to support more than two levels and "AND"-type dependencies [2] also.

3.2.3 Validity of Redaction/Reconstruction

At the level of implementation described above, all entries are left intact. This fact, along with the section tagging that is done upon extraction, ensures that no false claims may be made by a patient. This fact may be demonstrated by examining the structure as well as the control the patient has over the data items themselves. Given the fact that a patient may at most remove an arbitrary number of items from the tree, and has no control over the content of the items themselves; let us examine the validity of our argument. The data items each contain header information collected upon tree construction which details the section of the document from which they originated. This ensures that the patient has no control over to which sections items are assigned. This prevents a possible scenario of a patient attempting to claim that an item located in the problems section actually resides in the family history section of the document, or performing other section alterations. Also, since the patient may not alter any given entry, the item will always maintain the same meaning as given to it by the healthcare provider upon tree construction whenever it is disclosed by the patient.

4 Performance Results

In this section, we examine the performance of the proposed scheme on the first use case, where providers supply signed CCDs to a patient who then generates new CCDs on an as-needed basis containing only the specific data items necessary for a specific transaction. We examine the time for a provider to create and sign its own individual MHT from a CCD, the time for a patient to generate the MHT for a merged and redacted CCD containing data from multiple sources, and the time for a third party to verify an MHT it receives from a patient along with a CCD.

The platform used for all results in this section is as follows. All code was written in Java and executed in Eclipse 3.6.2 with Java SE 1.6 running on a Windows 7 PC with a 2.4 GHz Intel Core i5 processor with 4 GB of RAM. All times reported are CPU times.

4.1 Tree Creation

We will first begin by examining the creation of an MHT. In this study, we assume that the data items from the CCD are available as input. In the likely scenario where a health care provider extracts data from their electronic health record (EHR) system to generate a CCD that is provided to the patient, the data items can be simply saved for use by our tool as they are extracted from the EHR. Once the data items are collected, they must be randomized for storage in the tree and then the tree structure along with all of the hashes must be computed. Figure 3 provides the time overhead for the creation of a single MHT with various numbers of health items, given the items are provided as input.

The process of tree creation can be divided into tree formation and root signature generation, and Figure 3 shows the time for each of these operations. As shown by Figure 3, even at an extreme case of 1000 unique data items, the tree formation and root signature take about 13 milliseconds. We also ran this experiment on a collection of 206 actual CCDs, which we received in a de-identified form from an Atlanta-area medical provider. The execution times agreed with the above results and the average number of data items in the actual CCDs was 190, with a maximum of 828. Thus, the range of data sizes reported in Figure 3 is representative of typical scenarios and only tens of milliseconds are added to the time necessary to generate a CCD for a third party for the largest data sizes.

Our software tool is also capable of extracting health data items from a CCD. This would be necessary if the provider's native storage format was the CCD. However, we envision this to be an uncommon scenario.

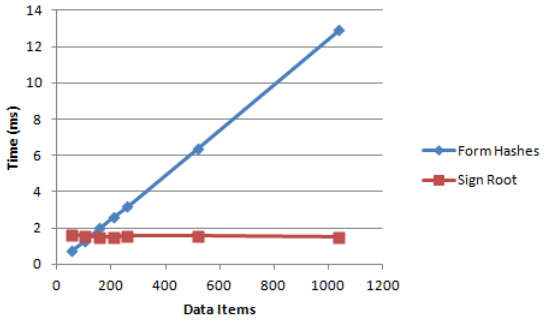

Figure 3: Form Tree Vs. Sign Tree

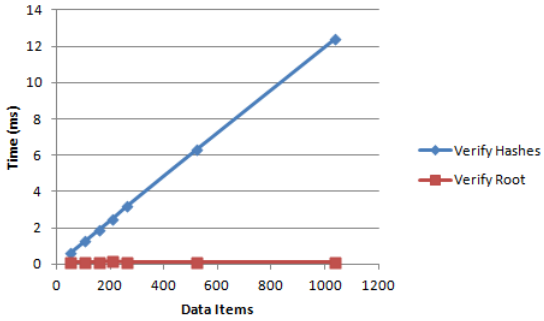

Figure 4: Verify Merkle Hash Tree

Our current data extraction tool uses a quick and dirty approach based on XPath queries, which is rather inefficient. In the worst case, this approach takes about 1 second to extract 1000 data items. If this scenario turns out to be more common than anticipated, a more efficient data extraction tool, which simply parses the document in a single pass to extract the items, can easily be implemented and should have performance on a par with standard XML parsers.

4.2 Generation of Merged and Redacted CCDs

The primary reason for this approach, as opposed to a single electronic signature across an entire document, is the feature of redaction. Figure 5 provides data relating to the time overhead required to redact a various number of items from a combined tree structure. In this scenario, a larger tree with twenty different sub-trees was provided. Health elements were randomly redacted in even distributions across the tree. The trend seen in Figure 5 shows a linear upper-bounded approximation while redacting increasing numbers of elements. The bending of the curve below a linear upper bound is due to the tree size decreasing as the number of items redacted increases (see Figure 6).

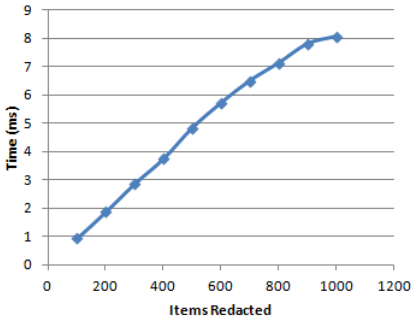

Figure 5: Time to Redact

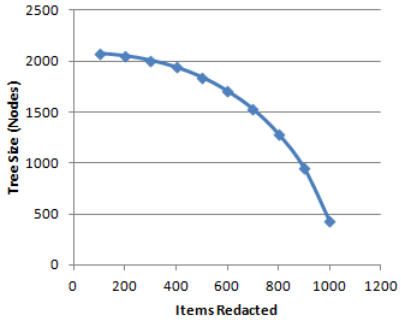

Figure 6: Number of Nodes

These results demonstrate the overhead to the patient in our primary use case. In general, the patient would take the time required on a given transaction to redact the unwanted items from the MHT structure. Additionally, the patient would be required to pay the cost of combining root nodes from multiple data sources and generating one additional signature for the new root. Due to the fact that the patient would only be combining a few structures, the cost of the combining, according to Figure 3, would be less than the cost of signing. The computational time required in this case is at most 12 milliseconds (at most 8 ms for redaction, at most 2 ms for combining structures, and 2 ms for signing the top level). Thus, the overhead for generating the information to make the merged and redacted CCD verifiable is quite low.

4.3 Verification

Another common interaction with such a data structure would be the act of verification. As mentioned earlier, verification is essentially re-constructing the tree structure with a public key verification instead of a signature. Therefore, one can notice that the results seen in Figure 4 mirror those seen in Figure 3. Signature verification time is 0.3 milliseconds as compared to the signature creation time of 2 milliseconds.

While these results demonstrate the verification for an MHT with a single signature, or source, the use case for multiple sources providing information can easily be analyzed as well. If it is necessary to verify data items across multiple sources, the signature verification time is simply multiplied by the number of sources. For example, with five sources, an offset of 1.5 milliseconds (5 sources times 0.3 milliseconds) would be added to the time needed to verify the tree, as depicted in Figure 3. Thus, in all foreseeable scenarios, the verification time will be in the tens of milliseconds.

The overhead seen in verification would be incurred by a third party who is verifying this information. For example, this third party may be another medical establishment or an insurance company verifying an individual's medical information for determining coverage.

5 Conclusion and Future Work

Overall, the results of the previous section demonstrate that redactable and verifiable CCDs can be implemented with very little overhead on any of the involved parties, medical providers that generate verifiable CCDs, patients or other intermediaries who merge and redact CCDs and pass them along, or third parties who receive and verify CCDs. The excellent performance of the approach derives from its reliance primarily on hash functions with only a small number of public key cryptography operations.

The strong performance of our approach is coupled with its ability to do privacy-preserving transformations, i.e. redactions, and cryptographically track data integrity and provenance. With this unique combinination of capabilities, the approach appears to be extremely well suited for the trusted indirect exchange of medical documents among a widely distributed set of entities, which is a health data exchange model that is of importance in a rapidly growing number of use cases.

The work done in this implementation was only a preliminary proof of concept for the use of Merkle Hash Trees for redactable signatures on medical documents. In a deployed system, a GUI that would allow users to visually browse CCD information and select specific data items to be redacted or to be kept would improve usability of our tools. Integration of additional data dependency constraints into the MHT structure would allow medical providers to assert more control over how the information they supply can be redacted. Finally, integration of the approach with relational databases that store individual data items extracted from medical documents such as CCDs would allow the approach to be deployed in realistic research scenarios. We are pursuing all of these directions in our current research efforts.

Acknowledgment

This research was supported in part by the National Science Foundation under Grant CNS-CT-0716252 and by the PHS Grant UL1 RR025008 from the Clinical and Translational Science Award program, National Institutes of Health, National Center for Research Resources.

References

[1] Johnson R, Molnar D, Song D, Wagner D. Homomorphic signature schemes. *Lecture Notes in Computer Science*. 2002;2271:204–245.

[2] Bauer D, Blough DM, Mohan A. Redactable signatures on data with dependencies and their application to personal health records. In: *Proceedings of the 8th ACM workshop on Privacy in the Electronic Society*. ACM; 2009. p. 91–100.

[3] Health Level 7 International. http://www.hl7.org/implement/standards/cda.cfm.

[4] NHIN Direct. http://wiki.directproject.org/.

[5] Microsoft Health Vault;. http://www.microsoft.com/en-us/healthvault/.

[6] Merkle R. Protocols for public key cryptosystems. In: *IEEE Symposium on Security and Privacy*. vol. 122; 1980.

[7] Merkle R. A certified digital signature. In: *Advances in Cryptology*–CRYPTO'89 Proceedings. Springer; 1990. p. 218–238.

[8] Bauer D, Blough DM, Cash D. Minimal information disclosure with efficiently verifiable credentials. In: *Proceedings of the 4th ACM Workshop on Digital Identity Management*. ACM; 2008. p. 15–24.

[9] Wu ZY, Hsueh CW, Tsai CY, Lai F, Lee HC, Chung Y. Redactable signatures for signed CDA documents. *Journal of Medical Systems*. 2010;p. 1–14.

[10] Slamanig D, Stingl C. Disclosing verifiable partial information of signed CDA documents using generalized redactable signatures. In: e-Health Networking, Applications and Services, 2009. Healthcom 2009. 11th International Conference on. IEEE; 2009. p. 146–152.

[11] Slamanig D, Rass S. Generalizations and extensions of redactable signatures with applications to electronic healthcare. In: *Communications and Multimedia Security*. Springer; 2010. p. 201–213.

[12] Miyazaki K, Iwamura M, Matsumoto T, Sasaki R, Yoshiura H, Tezuka S, et al. Digitally signed document sanitizing scheme with disclosure condition control. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*. 2005;p. 239–246.

[13] Miyazaki K, Hanaoka G, Imai H. Digitally signed document sanitizing scheme based on bilinear maps. In: *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*. ACM; 2006. p. 343–354.

[14] Haber S, Hatano Y, Honda Y, Horne W, Miyazaki K, Sander T, et al. Efficient signature schemes supporting redaction, pseudonymization, and data deidentification. In: *Proceedings of the 2008 ACM symposium on Information, Computer and Communications Security*. ACM; 2008. p. 353–362.

[15] Izu T, Kanaya N, Takenaka M, Yoshioka T. PIATS: A partially sanitizable signature scheme. *Information and Communications Security*. 2005;p. 72–83.

[16] Polivy DJ, Tamassia R. Authenticating distributed data using Web services and XML signatures. In: *Proceedings of the 2002 ACM Workshop on XML security*. ACM; 2002. p. 80–89.

[17] Steinfeld R, Bull L, Zheng Y. Content extraction signatures. *Proceedings of Information Security and Cryptology*, ICISC 2001. 2002;p. 163–205.

[18] Bull L, Stanski P, Squire DM. Content extraction signatures using XML digital signatures and custom transforms on-demand. In: *Proceedings of the 12th International Conference on World Wide Web*. ACM; 2003. p. 170–177.

[19] Bull L, Squire DM, Zheng Y. A hierarchical extraction policy for content extraction signatures. *International Journal on Digital Libraries*. 2004;4(3):208–222.

[20] Suenaga T, Takada A. Layered secure medical information exchange platform. In: *Proceedings of Information Technology Applications in Biomedicine*, 2007. ITAB 2007. 6th International Special Topic Conference on. IEEE; 2007. p. 157–160.